

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) $H(\text{firstname}||\text{lastname})$

- ☒ 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

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) $MAC(k, \text{firstname}||\text{lastname})$

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

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.



(Question 1 continued...)

Q1.3 (3 points) $\text{MAC}(k, \text{firstname} \parallel \text{"-"} \parallel \text{lastname})$, 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

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

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

- ☐ 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

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

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

- ☐ 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

Solution: Because the last name and the first name have separate MACs, Mallory could swap the first name and the 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 $\text{MAC}(k, \text{nick}) \parallel \text{MAC}(k, \text{weaver})$ to $\text{MAC}(k, \text{weaver}) \parallel \text{MAC}(k, \text{nick})$.

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

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

Q2 Confidentiality and Integrity

(4 points)

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

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

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.

Alice sends to Bob

1. $c = \text{Hash}(\text{Enc}(K, m))$
2. $c = c_1, c_2$: where $c_1 = \text{Enc}(K, m)$ and $c_2 = \text{Hash}(c_1)$
3. $c = c_1, c_2$: where $c_1 = \text{Enc}(K, m)$ and $c_2 = \text{MAC}(K, m)$
4. $c = c_1, c_2$: where $c_1 = \text{Enc}(K, m)$ and $c_2 = \text{MAC}(K, c_1)$

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

☐ 1 ☒ 2 ☒ 3 ☒ 4

Solution: Bob cannot decrypt Scheme 1 because he cannot invert Hash.

Q2.2 (1 point) Consider an eavesdropper Eve, who can see the communication between Alice and Bob.

Out of all of the schemes decryptable in 2.1, which schemes also provide *confidentiality* against Eve?

☐ 1 ☒ 2 ☐ 3 ☒ 4

Solution: Scheme 3 does not provide confidentiality because the MAC is performed on plaintext. For the same message, the MAC is the same, so Eve can win the IND-CPA game.

Q2.3 (1 point) Consider a man-in-the-middle Mallory, who can eavesdrop and modify the communication between Alice and Bob.

Out of all of the schemes decryptable in 2.1, which schemes also 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 $(c', \text{Hash}(c'))$.

(Question 2 continued...)

Q2.4 (1 point) 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*.

For each scheme that has **both** confidentiality against Eve (2.2) *and* integrity against Mallory (2.3), how can the scheme be modified to prevent a replay attack?

Solution: Only scheme 4 meets both requirements. To modify the scheme such that a replay attack is not possible, change m to be equal to $(m \parallel \text{nonce})$ or $(m \parallel \text{timestamp})$, where nonce is an incrementing counter (such that no messages have the same nonce), or timestamp is the timestamp the message was sent at. Have Bob validate that the Nonce or timestamp had not been received before, and discard the message if it has.

Q3 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 $g^b \bmod p$.

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

a	b	g	p
<input type="radio"/> Public	<input type="radio"/> Public	<input checked="" type="radio"/> Public	<input checked="" type="radio"/> Public
<input checked="" type="radio"/> Private	<input checked="" type="radio"/> Private	<input type="radio"/> Private	<input type="radio"/> Private

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

Q3.3 (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*.