Question 1 ***MAC Madness*** **(18 min)**

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 that usernames must be unique.

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: Now, the concatenation includes a separator between first name and last name, so the attack from the previous part is no longer possible. Note that names are alphabetical, so they would never include a dash in them.

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: Hashes have fixed-length output, so the attack from the previous part (shifting letters between the first and last name) is not possible here either. It will always be unambiguous where the first hash ends and the second hash begins.

Also, since both hashes are used as input to a single MAC, there is no way for an attacker without the key to generate a valid MAC for any different name.

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 ***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 $g^b \bmod p$.

Q2.1 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.

Q2.2 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 protocols which can be used by Alice (A) and Bob (B) to agree upon a shared key, K.

ElGamal-Based Key Exchange

Message 1	$A \rightarrow B$:	$\{K\}_{PK_B}$
	Key exchanged	
Message 2	$A \leftarrow B$:	$\{secret1\}_K$
Message 3	$A \rightarrow B$:	$\{secret2\}_K$

Diffie-Hellman Key Exchange

Message 1	$A \rightarrow B$:	$g^a \mod p$
Message 2	$A \leftarrow B$:	$g^b \mod p$
	Key exchanged	
	$K = g^{ab} \mod p$	
Message 3	$A \leftarrow B$:	$\{secret1\}_K$
Message 4	$A \rightarrow B$:	$\{secret2\}_K$

Some additional details:

- PK_B is Bob's long-lived public key.
- 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.

Q2.1 Is the confidentiality of Alice and Bob's prior ElGamal-based communication in jeopardy?

> **Solution:** Yes. The compromise of Bob's computer gives Eve access to Bob's private key, allowing Eve to decrypt the traffic she previously recorded that was encrypted using Bob's public key. Once decrypted, she obtains K, and can then apply it to decrypt the traffic encrypted using symmetric key encryption.

Q2.2 What about Alice and Bob's Diffie-Hellman-based communication?

> **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*.

Question 3 ***Public Key Encryption*** ()

The El Gamal encryption scheme is reproduced below:

- **Key Generation**: public key $= (g, h, p)$, where $h = g^k \pmod{p}$, private key $= k$
- **Encryption**: $c = (c_1, c_2) = (g^r \bmod p, m \times h^r \bmod p)$, where r is randomly sampled from $\{1, \ldots, p-1\}$.
- **Decryption**: $m = c_1^{-k} \times c_2 \pmod{p}$

Look at each scenario below and select the appropriate options.

Q3.1 With El Gamal, it is not a problem if the adversary can learn the value of g somehow.

● (A) True

○ (B) False

○ (C) —

○ (D) —

○ (E) —

○ (F) —

Solution: g is part of the public key, so it is fine for it to be known to the public (including the adversary).

Q3.2 With El Gamal, it is not a problem if the value r used during encryption is accidentally revealed after the encryption is complete.

○ (G) True

● (H) False

○ (I) —

○ (J) —

○ (K) —

○ (L) —

Solution: If the adversary learns r, they can compute $c_2 h^{-r} \bmod p$, and that will reveal the message m.

Now imagine that Alice (A) and Bob (B) want to communicate over an insecure network and they know each other's public key. Consider the following message exchange:

A: Hey Bob, it's Alice. How many dollars do I owe you?
B: 10000

The message is encrypted with Alice's public key using ElGamal encryption.
Alice decrypted this successfully, but suddenly remembered that she only owed Bob $100.

Q3.1 Assume Bob would not lie. How did an attacker tamper with the message?

Solution: The attacker multiplied c_2 by 100, or multiplied $c_1 \cdot c_1', c_2 \cdot c_2'$ where c' is a valid encryption of 100, or they encrypted an entirely new message.

Q3.2 What could Bob have additionally sent that would've stopped this attack?

Solution: Bob could attach a signature to his original message.