Q1 *The Lorenzo Von Matterhorn*   **(0 points)**

Barney needs to make sure that no attackers can access his highly sensitive, top secret playbook tricks!

For each password scheme, select all true statements. Assume that:

- Each user has a unique username, but not necessarily a unique password.

- All information is stored in a read-only database that both the server and the attacker can access.

- The server has a symmetric key K not known to anyone else. The server also has a secret key SK not known to anyone else, and a corresponding public key PK that everyone knows.

- An *operation* is defined as one of the following actions: hash, encryption, decryption, and HMAC.

- The attacker does not have access to a client UI; therefore, online attacks are not possible.

Q1.1 For each user, the database contains username and H(password), where H is a cryptographic hash function.

☐ If a user inputs a username and password, the server can verify whether the password is correct

☐ Given the information in the database, the attacker can verify that a given username and password pairing is correct.

☐ The server can list all plaintext passwords by computing at most one operation per user

☐ An attacker can list all passwords by computing at most one operation per possible password

☐ None of the above

Q1.2 For each user, the database contains username and HMAC(K, password).

☐ If a user inputs a username and password, the server can verify whether the password is correct

☐ Given the information in the database, the attacker can verify that a given username and password pairing is correct.

☐ The server can list all plaintext passwords by computing at most one operation per user

☐ An attacker can list all passwords by computing at most one operation per possible password

☐ None of the above

Q1.3 For this subpart, Enc denotes an IND-CPA secure symmetric encryption function.

For each user, the database contains username and $\mathsf{Enc}(K, \mathsf{password})$.

☐ If a user inputs a username and password, the server can verify whether the password is correct

☐ Given the information in the database, the attacker can verify that a given username and password pairing is correct.

☐ The server can list all plaintext passwords by computing at most one operation per user

☐ An attacker can list all passwords by computing at most one operation per possible password

☐ None of the above

Q1.4 For this subpart, RSA denotes RSA encryption without OAEP padding.

For each user, the database contains username and $\mathsf{RSA}(\mathsf{PK}, \mathsf{password})$.

☐ If a user inputs a username and password, the server can verify whether the password is correct

☐ Given the information in the database, the attacker can verify that a given username and password pairing is correct.

☐ The server can list all plaintext passwords by computing at most one operation per user

☐ An attacker can list all passwords by computing at most one operation per possible password

☐ None of the above

Q1.5 Consider a modification to the scheme in the first subpart: Instead of storing $\mathsf{H}(\mathsf{password})$ per user, we now store $\mathsf{H}(\mathsf{password}||\mathsf{salt})$ per user.

Assume that concatenation does not count as an operation. Compared to the original scheme, which of the following algorithms for generating salts would force the attacker to compute more operations to list all passwords? Select all that apply.

☐ A 128-bit value, randomly generated per user

☐ A 128-bit counter, starting at 0 and incremented per user

☐ A 128-bit counter, starting at a random number and incremented per user

☐ None of the above

Q1.6 Which of these hash algorithms makes the scheme in the first subpart most secure against offline brute-force attacks? Briefly explain (10 words or fewer).

○ MD5　　　　　　　　○ SHA2-256　　　　　　　○ Argon2Key (PBKDF2)

Q2 *So, you want a secure key?* (16 points)

Alice wants to create a secure channel of communication with a server. From CS 161, Alice remembers that the best way of communicating is to somehow end up with a shared, symmetric key, but has no idea how this process works.

Assume that there exists a certificate authority (CA) actively sending certificates to many clients. Assume that Mallory, a MITM attacker, and Eve, an eavesdropper, can both exist in all communication channels for all subparts unless otherwise specified.

Q2.1 (2 points) Alice first wants to authenticate the CA before authenticating the server. She remembers that certificates provide authenticity, so she exchanges the following messages with the CA:

1. Alice queries the CA for the CA's public key and receives PK_{CA}.

2. Alice queries the CA for the server's public key and receives
   {"The server's public key is PK_S"}$_{SK_{\text{CA}}^{-1}}$.

Can Mallory trick Alice into accepting a different public key PK_S' **of Mallory's choosing** as the server's public key without being detected?

○ Yes ○ No

For the rest of this question, assume that instead of querying the CA for their public key, Alice has the CA's correct public key, PK_{CA}, hardcoded into her computer.

Q2.2 (5 points) When Alice queries the CA for the server's public key, the CA sends {"The server's public key is PK_S"}$_{SK_{\text{CA}}^{-1}}$.

Can Mallory trick Alice into accepting a different public key PK_S', **not necessarily of Mallory's choosing**, as the server's public key without being detected?

If you mark "Yes", provide an attack that would accomplish this goal. If you mark "No", explain why not in 2 sentences or fewer.

○ Yes ○ No

Q2.3 (5 points) When Alice queries the CA for the server's public key, the CA selects a random number x between 1 and 20 and sends $\{\text{"The server's public key is } PK_S\text{"}\|x\}_{SK_{\text{CA}}^{-1}}$.

Can Mallory trick Alice into accepting a different public key PK_S', **not necessarily of Mallory's choosing**, as the server's public key without being detected?

If you mark "Yes", provide an attack that would accomplish this goal. If you mark "No", explain why not in 2 sentences or fewer.

○ Yes   ○ No

Q2.4 (4 points) Alice has received some public key PK_S from the CA, but she doesn't trust that PK_S belongs to the server. Which of the following messages can the server send to convince Alice that she is talking to the legitimate server? Select all that apply.

☐ The server sends $H(PK_S)$

☐ The server sends $\mathsf{Sign}(SK_S, H(PK_S))$

☐ The server sends $H(SK_S\|PK_S)$

☐ The server randomly generates a symmetric key K and sends $(\mathsf{Sign}(SK_S, K), \mathsf{HMAC}(K, PK_S))$

☐ None of the above

Q3 *Chegg Certificates* (7 points)

Chegg uses a certificate chain in order to verify tutors. When tutors post responses, they attach a digital signature of their response along with their certificate. Students can verify the authenticity of a response by verifying the certificate and using the public key in the certificate to verify the signature.

The certificate chain is below. Assume that the Chegg Root Certificate Authority (CA) is hardcoded into students' browsers.

1. Identity: Director of Chegg Recruiting (Verified by Chegg Root CA)

2. Identity: Campus Chegg Recruiter (Verified by Director of Chegg Recruiting)

3. Identity: Authorized Tutor (Verified by Campus Chegg Recruiter)

Q3.4 (4 points) EvanBot is not a valid tutor, but wants to create a fake tutor response with a valid signature. Which of these attacks would allow Bot to accomplish this?

☐ Steal the public key of the Campus Chegg Recruiter

☐ Steal the private key of the Director of Chegg Recruiting

☐ Steal the private key of the Chegg Root CA

☐ Steal the certificate of an authorized tutor

☐ None of the above

Q3.5 (3 points) EvanBot gains access to the private key of Dave, who is an authorized tutor. Which of the following can EvanBot do?

☐ Post a valid response as Nick, an existing tutor

☐ Post a valid response as Dave

☐ Create and sign a certificate for Raluca, a new tutor

☐ None of the above