Q1 *Bob's Birthday* (11 points)

It's Bob's birthday! Alice wants to send an encrypted birthday message to Bob using ElGamal.

Recall the definition of ElGamal encryption:

- b is the private key, and $B = g^b \bmod p$ is the public key.
- $\mathsf{Enc}(B, M) = (C_1, C_2)$, where $C_1 = g^r \bmod p$ and $C_2 = M \times B^r \bmod p$
- $\mathsf{Dec}(b, C_1, C_2) = C_1^{-b} \times C_2 \bmod p$

Q1.1 (2 points) Mallory wants to tamper with Alice's message to Bob. In response, Alice decides to sign her message with an RSA digital signature. Bob receives the signed message and verifies the signature successfully. Can he be sure the message is from Alice?

- ● Yes, because RSA digital signatures are unforgeable.
- ○ Yes, because RSA encryption is IND-CPA secure.
- ○ No, because Mallory could have blocked Alice's message and replaced it with a different one.
- ○ No, because Mallory could find a different message with the same hash as Alice's original message.

> **Solution:** RSA digital signatures, when paired with a secure hash function, are believed to be unforgeable. See the textbook for a game-based definition of what exactly we mean by unforgeable.

As we discussed in class, ElGamal is malleable, meaning that a man-in-the-middle can change a message in a *predictable* manner, such as producing the ciphertext of the message $2 \times M$ given the ciphertext of M.

Q1.2 (3 points) Consider the following modification to ElGamal: Encrypt as normal, but further encrypt portions of the ciphertext with a block cipher E, which has a block size equal to the number of bits in p. In this scheme, Alice and Bob share a symmetric key K_{sym} known to no one else.

Under this modified scheme, C_1 is computed as $E_{K_{\mathsf{sym}}}(g^r \bmod p)$ and C_2 is computed as $E_{K_{\mathsf{sym}}}(M \times B^r \bmod p)$. Is this scheme still malleable?

- ○ Yes, because block ciphers are not IND-CPA secure encryption schemes
- ○ Yes, because the adversary can still forge $k \times C_2$ to produce $k \times M$
- ● No, because block ciphers are a pseudorandom permutation
- ○ No, because the adversary isn't able to learn anything about the message M

Solution: While block ciphers aren't IND-CPA secure, they are secure when encrypting "random-looking" values because of their properties as pseudorandom permutations. As long as the values you encrypt are unique, the output of the block cipher will always be secure. ElGamal's C_1 and C_2 both appaer random.

Additionally, because block ciphers are a PRP, the scheme is no longer malleable, because modifying the ciphertext in any way causes an unpredictable change to the result of decrypting the block cipher with $D_{K_{\mathsf{sym}}}$.

The remaining parts are independent of the previous part.

For Bob's birthday, Mallory hacks into Bob's computer, which stores Bob's private key b. She isn't able to read b or overwrite b with an arbitrary value, but she can multiply the stored value of b by a random value z known to Mallory.

Mallory wants to send a message to Bob that appears to decrypt as normal, but **using the modified key** $b \cdot z$. Give a new encryption formula for C_1 and C_2 that Mallory should use. Make sure you only use values known to Mallory!

Clarification during exam: For subparts 3 and 4, assume that the value of B is unchanged.

Q1.3 (3 points) Give a formula to produce C_1, encrypting M.

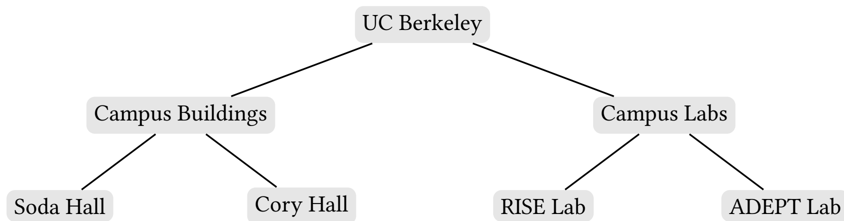
Solution: Mallory should send g^r with some randomly chosen r, as usual.

Q1.4 (3 points) Give a formula to produce C_2, encrypting M.

Solution: Mallory should send $C_2 = m \times B^{rz} \mod p$.

Q2 ***RISELab Shenanigans*** **(0 points)**

Certificate authorities of UC Berkeley are organized in a hierarchy as follows:



Alice is a student in RISELab at UC Berkeley and wants to obtain a certificate for her public key. Assume that only RISELab is allowed to issue certificates to Alice.

Q2.1 Which of the following values are included in the certificate issued to Alice? Select all that apply.

■ Alice's public key

□ Alice's private key

■ A signature on Alice's *public* key, signed by RISELab's private key

□ A signature on Alice's *private* key, signed by RISELab's private key

□ None of the above

> **Solution:** This follows from the definition of certificates: they include a user's public key, and a signature on the enclosed public key, signed by the issuer (which we state in the prologue is RISELab).

Q2.2 Assume that the only public key you trust is UC Berkeley's public key. Which certificates do you need to verify in order to be sure that you have Alice's public key? Select all that apply.

■ Certificate for Alice

□ Certificate for Soda Hall

■ Certificate for RISELab

■ Certificate for Campus Labs

□ None of the above

> **Solution:** To validate Alice's public key, we can follow our way up to our root of trust (which is UC Berkeley's public key). As such, we need certificates for Alice, RISELab, and Campus Labs.

Q2.3 RISELab issues a certificate to Alice that expires in 1 hour. Which of the following statements are true about using such a short expiration date? Select all that apply.

- ■ It mitigates attacks where Alice's private key is stolen
- ☐ It mitigates attacks where RISELab's private key is stolen
- ☐ It mitigates attacks where Campus Labs' private key is stolen
- ■ It forces Alice to renew the certificate more often
- ☐ None of the above

Solution: Short expiration times only mitigate the situation where Alice's private key is stolen. If RISELab's private key is compromised, the attacker can issue certificates with any expiration date, and it is up to the parent CA to revoke RISELab's certificate, not RISELab itself. The same argument applies to Campus Labs' private key.

Q3 *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.

Q3.1 For each user, the database contains $\mathsf{username}$ and $\mathsf{H}(\mathsf{password})$, 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

Solution:

A: The server can hash the password to check that it matches the hash in the database.

B: The attacker can hash the password (hashes aren't keyed) and check that it matches the hash in the database.

C: The server cannot compute one operation to reverse a hash.

D: The attacker can conduct an offline brute-force attack, hashing every possible password and comparing to the hashes in the database.

Q3.2 For each user, the database contains $\mathsf{username}$ and $\mathsf{HMAC}(K, \mathsf{password})$.

- ■ 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

Solution:

A: The server can HMAC the password to check that it matches the HMAC in the database.

B: An attacker cannot compute the output of HMAC without knowing the key input K.

C: The server cannot compute one operation to reverse HMAC.

D: The attacker cannot compute HMACs without knowing the key input K.

Q3.3 For this subpart, Enc denotes an IND-CPA secure symmetric encryption function.

For each user, the database contains $\mathsf{username}$ 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

Solution:

A: The server can decrypt the password in the database to check that it matches the password given by the user.

B: An attacker cannot decrypt passwords without knowing the key input K.

C: The server can decrypt the password in the database.

D: An attacker cannot decrypt passwords without knowing the key input K.

Q3.4 For this subpart, RSA denotes RSA encryption without OAEP padding.

For each user, the database contains username and RSA(PK, 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

Solution:

A: The server can encrypt the password to check that it matches the password in the database.

B: The attacker can also encrypt passwords, because everybody knows the public key.

C: The server can decrypt the password in the database.

D: An attacker can encrypt every possible password and compare the encryptions to the encryptions in the database.

Q3.5 Consider a modification to the scheme in the first subpart: Instead of storing H(password) per user, we now store H(password||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

Solution: Salts need to be unique per user. If salts are unique, then the attacker needs to hash the dictionary of passwords once per user, instead of once for all users.

Q3.6 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)

Solution: Hashes need to be slow to increase the amount of time a dictionary attack takes. MD5 and SHA2-256 are faster hashes, and Argon2Key (PBKDF2) is a slower hash.