CS161 Spring 2026 — Introduction to Computer Security — Exam Prep 6

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.
- $\text{Enc}(B, M) = (C_1, C_2)$, where $C_1 = g^r \bmod p$ and $C_2 = M \times B^r b \bmod p$
- $\text{Dec}(b, C_1, C_2) = C_1^{-b} \times C_2 \bmod p$

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.

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 $E_{\text{Ksym}}\left(g^r \bmod p\right)$ and C_2 is computed as $E_{\text{Ksym}}\left(M \times B^r \bmod p\right)$. 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

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.

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

Q2 *Cryptography: EvanBot Signature Scheme* **(12 points)**

EvanBot decides to make a signature scheme!

To intialize the system, a Diffie-Hellman generator g and prime p are generated and shared to all parties. The private key is some $x \bmod p$ chosen randomly, and the public key is $y = g^x \bmod p$.

To sign a message m such that $2 \leq m \leq p-2$:

1. Choose a random integer k between 2 and $p-2$.
2. Set $r = g^k \bmod p$.
3. Set $s = (\mathsf{H}(m) - xr)k^{-1} \bmod (p-1)$. If $s = 0$, restart from Step 1.
4. Output (r, s) as the signature.

Clarification after exam: k is chosen to be coprime to $p-1$

To verify, check that $g^{\mathsf{H}(m)} \equiv$ _____ $\bmod p$. We will fill in this blank in the next few subparts.

Q2.1 (3 points) Select the correct expression for $\mathsf{H}(m)$ in terms of x, r, k, s, and $p-1$.
HINT: Use Step 3 of the signature algorithm.

- ◯ $k(xr)^{-1} + s \bmod (p-1)$
- ◯ $k^{-1} + xr \bmod (p-1)$
- ◯ $ks - xr \bmod (p-1)$
- ◯ $ks + xr \bmod (p-1)$

Q2.2 (4 points) Using the previous result, select the correct value for the blank in the verification step.
HINT: Replace the $\mathsf{H}(m)$ in $g^{\mathsf{H}(m)}$ with your results from the previous subpart.

- ◯ $y^s r^2 \bmod p$
- ◯ $r^y r^s \bmod p$
- ◯ $y^r r^s \bmod p$
- ◯ $r g^{yr} \bmod p$

Q2.3 (5 points) Show how to recover the private key x if a signature is generated such that $s = 0$ (i.e. the check on Step 3 is ignored).

Q3 *Hash Functions: YAAS (Yet Another Authentication Scheme)* **(8 points)**

EvanBot decides to design a new authentication scheme.

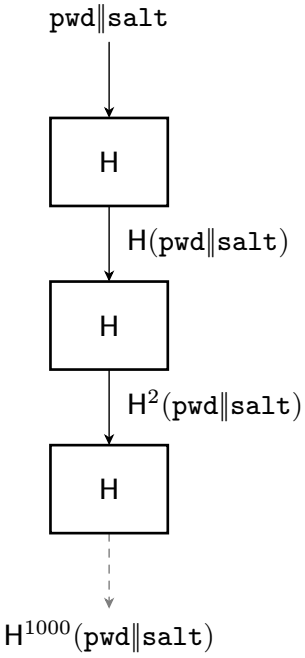
Define `pwd` to be a secure password that only EvanBot knows.

Also, define H^k to be the result of repeatedly applying H, a cryptographically secure hash function, k times. Note $\mathsf{H}^0(x) = x$.

$$\mathsf{H}^{k(x)} = \underbrace{\mathsf{H}(\mathsf{H}(...\mathsf{H}}_{k \text{ times}}(x)))$$

To sign up:
1. EvanBot securely generates a 128-bit salt `salt`.
2. EvanBot sends $\mathsf{H}^{1000}(\texttt{pwd}\|\texttt{salt})$ to the server.
3. The server maps EvanBot's username to a variable called `stored`. The server sets `stored` to be the value received in Step 2.

To log in for the n-th time (n starts at 1):
1. EvanBot sends $\mathsf{H}^{1000-n}(\texttt{pwd}\|\texttt{salt})$ to the server.
2. The server checks whether [ANSWER TO Q3.1].
3. If Step 2 succeeds, the server updates `stored` to [ANSWER TO Q3.2].



Q3.1 (1 point) Let `Step1` be the value received in Step 1 of the login process. Select the correct option for the blank in Step 2.

- ◯ $\mathsf{H}(\texttt{Step1}) = \texttt{stored}$
- ◯ $\mathsf{H}(\texttt{stored}\|\texttt{salt}) = \texttt{Step1}$
- ◯ $\mathsf{H}(\texttt{stored}) = \texttt{Step1}$
- ◯ $\mathsf{H}(\texttt{Step1}\|\texttt{salt}) = \texttt{stored}$

Q3.2 (1 point) Select the correct option for the blank in Step 3.

- ◯ `stored` (no update needed)
- ◯ `Step1`
- ◯ $\mathsf{H}(\texttt{stored}\|\texttt{salt})$
- ◯ $\mathsf{H}^2(\texttt{Step1})$

Q3.3 (1 point) Does the server need to know `salt` in order to complete the login process?

- ◯ Yes
- ◯ No

Q3.4 (1 point) Eventually, EvanBot logs in by sending $\mathsf{H}^{700}(\texttt{pwd}\|\texttt{salt})$. Given only `pwd`, `salt`, and $H^{700}(\texttt{pwd}\|\texttt{salt})$, how many calls to H does EvanBot need to make on the next login request?

- ◯ 0
- ◯ 1
- ◯ 699
- ◯ 700

Q3.5 (2 points) Eve is an on-path attacker.

Which of these sets of values, if seen by Eve, would allow Eve to learn the password? Each answer choice is independent.

- ☐ The first login attempt only
- ☐ Any two login attempts in a row
- ☐ The 999th login attempt only
- ☐ The 1000th login attempt only
- ☐ All of the first 999 login attempts
- ☐ All of the first 1000 login attempts
- ○ None of the above

Q3.6 (2 points) Assume an attacker has compromised the server and can modify `stored`. Can the attacker log in as EvanBot?

- ○ Yes, without knowing n, `pwd`, or `salt`.
- ○ Yes, but only if they know n and `salt`.
- ○ No, even if they know n and `salt`.
- ○ Yes, but only if they know `salt`.
- ○ Yes, but only if they know n.