Q1 *The Red Hood* (15 points)

Jason Todd decides to launch a communications channel in order to securely communicate with the Red Hood Gang over an insecure channel. Jason wants to test different schemes in his attempt to attain confidentiality and integrity.

Notation:

- M is the message Jason sends to the recipient.
- K_1, K_2, and K_3 are secret keys known to only Jason and the recipient.
- ECB, CBC, and CTR represent block cipher encryption modes for a secure block cipher.
- Assume that CBC and CTR mode are called with randomly generated IVs.
- H is SHA2, a collision-resistant, one-way hash function.
- HMAC is the HMAC construction from lecture.

Decide whether each scheme below provides confidentiality, integrity, both, or neither. For all question parts, the ciphertext is the value of C; t **is a temporary value that is not sent as part of the ciphertext.**

Q1.1 (3 points)

$t = \mathsf{CBC}(K_1, M)$ $C_1 = \mathsf{ECB}(K_2, t)$ $C_2 = \mathsf{HMAC}(K_3, t)$ $C = (C_1, C_2)$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q1.2 (3 points)

$t = \mathsf{ECB}(K_1, M)$ $C_1 = \mathsf{CBC}(K_2, t)$ $C_2 = \mathsf{HMAC}(K_3, t)$ $C = (C_1, C_2)$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q1.3 (3 points)

$$C_1 = \mathsf{ECB}(K_1, M) \qquad C_2 = H(K_2 \| C_1) \qquad C = (C_1, C_2)$$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q1.4 (3 points) For this subpart only, assume that i a monotonically, increasing counter incremented per message.

$$C_1 = \mathsf{CTR}(K_1, M) \qquad C_2 = \mathsf{HMAC}(i, H(C_1)) \qquad C = (C_1, C_2)$$

Clarification issued during exam: Assume that the counter, i, starts at 0.

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q1.5 (3 points) For this subpart only, assume that the block size of block cipher is n, the lengths of K_1 and K_2 are n, the length of M must be $2n$, and the length of the hash produced by H is $2n$.

$$C_1 = \mathsf{CBC}(K_1, K_2) \qquad C_2 = M \oplus C_1 \oplus H(C_1) \qquad C = (C_1, C_2)$$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q2 *PRNGs and Diffie-Hellman Key Exchange* **(15 points)**

Eve is an eavesdropper listening to an insecure channel between Alice and Bob.

1. Alice and Bob each seed a PRNG with different random inputs.
2. Alice and Bob each use their PRNG to generate some pseudorandom output.
3. Eve learns both Alice's and Bob's pseudorandom outputs from step 2.
4. Alice, without reseeding, uses her PRNG from the previous steps to generate a, and Bob, without reseeding, uses his PRNG from the previous steps to generate b.
5. Alice and Bob perform a Diffie-Hellman key exchange using their generated secrets (a and b). Recall that, in Diffie-Hellman, neither a nor b are directly sent over the channel.

For each choice of PRNG constructions, select the minimum number of PRNGs Eve needs to compromise (learn the internal state of) in order to learn the Diffie-Hellman shared secret $g^{ab} \bmod p$. Assume that Eve always learns the internal state of a PRNG between steps 3 and 4.

Q2.1 (3 points) Alice and Bob both use a PRNG that outputs the same number each time.

◯ (A) Neither PRNG ◯ (C) Both PRNGs ◯ (E) ——

◯ (B) One PRNG ◯ (D) Eve can't learn the secret ◯ (F) ——

Q2.2 (3 points) Alice uses a secure, rollback-resistant PRNG. Bob uses a PRNG that outputs the same number each time.

◯ (G) Neither PRNG ◯ (I) Both PRNGs ◯ (K) ——

◯ (H) One PRNG ◯ (J) Eve can't learn the secret ◯ (L) ——

Q2.3 (3 points) Alice and Bob both use a secure, rollback-resistant PRNG.

◯ (A) Neither PRNG ◯ (C) Both PRNGs ◯ (E) ——

◯ (B) One PRNG ◯ (D) Eve can't learn the secret ◯ (F) ——

For the rest of the question, consider a different sequence of steps:

1. Alice and Bob each seed a PRNG with different random inputs.
2. Alice uses her PRNG from the previous step to generate a, and Bob uses his PRNG from the previous step to generate b.
3. Alice and Bob perform a Diffie-Hellman key exchange using their generated secrets (a and b).
4. Alice and Bob, without reseeding, each use their PRNG to generate some pseudorandom output.
5. Eve learns both Alice's and Bob's pseudorandom outputs from step 2.

As before, assume that Eve always learns the internal state of a PRNG between steps 3 and 4.

Q2.4 (3 points) Alice and Bob both use a secure, but not rollback-resistant PRNG.

◯ (G) Neither PRNG ◯ (I) Both PRNGs ◯ (K) ——

◯ (H) One PRNG ◯ (J) Eve can't learn the secret ◯ (L) ——

Q2.5 (3 points) Alice and Bob both use a secure, rollback-resistant PRNG.

◯ (A) Neither PRNG ◯ (C) Both PRNGs ◯ (E) ——

◯ (B) One PRNG ◯ (D) Eve can't learn the secret ◯ (F) ——

Q3 *Bonsai* **(10 points)**

EvanBot wants to store a file in an *untrusted* database that the adversary can read and modify.

Before storing the file, EvanBot computes a hash over the contents of the file and stores the hash separately. When retrieving the file, EvanBot re-computes a hash over the file contents, and, if the computed hash doesn't match the stored hash, then EvanBot concludes that the file has been tampered with.

Clarification during exam: Assume that EvanBot does not know if hashes or files have been modified in the untrusted datastore.

Q3.1 (4 points) What assumptions are needed for this scheme to guarantee integrity on the file? Select all that apply.

☐ (A) An attacker cannot tamper with EvanBot's stored hash

☐ (B) EvanBot has a secret key that nobody else knows

☐ (C) The file is at most 128 bits long

☐ (D) EvanBot uses a secure cryptographic hash

☐ (E) None of the above

☐ (F) ——

For the rest of this question, we refer to two databases: a *trusted database* that an adversary cannot read or modify, and an *untrusted database* that an adversary can read and modify.

Assume that H is a secure cryptographic hash function and $\|$ denotes concatenation.

EvanBot creates and stores four files, F_1, F_2, F_3, and F_4, in the untrusted database. EvanBot also computes and stores a hash on each file's contents in the untrusted database:

$$h_1 = \mathsf{H}(F_1) \qquad h_2 = \mathsf{H}(F_2) \qquad h_3 = \mathsf{H}(F_3) \qquad h_4 = \mathsf{H}(F_4)$$

Then, EvanBot stores $h_{\mathsf{root}} = \mathsf{H}(h_1 \| h_2 \| h_3 \| h_4)$ in the *trusted* database.

Q3.2 (3 points) If an attacker modifies F_2 stored on the server, will EvanBot be able to detect the tampering?

◯ (G) Yes, because EvanBot can compute h_{root} and see it doesn't match the stored h_{root}

◯ (H) Yes, because EvanBot can compute h_2 and see it doesn't match the stored h_2

◯ (I) No, because the hash doesn't use a secret key

◯ (J) No, because the attacker can re-compute h_2 to be the hash of the modified file

◯ (K) ——

◯ (L) ——

Q3.3 (3 points) What is the minimum number of hashes EvanBot needs to compute to verify the integrity of all four files?

◯ (A) 1 ◯ (C) 3 ◯ (E) 5

◯ (B) 2 ◯ (D) 4 ◯ (F) More than 5