Mid-term exam – March 20., 2026 (solutions)

Exercise 1 (4 points). *Short questions*

1. **(1pt)** Both are deterministic functions that take some input message and output a fixed-length digest. Let $\{0,1\}^n$ be the digest space. Then a hash function takes variable-length inputs (usually $\{0,1\}^{<N}$ with very large N, or even $\{0,1\}^*$). Compression functions take fixed-length inputs, and another parameter from the digest space. (The goal of the *digest* parameter is to be able to iterate the compression function, when building a compression-based hash function.)
2. i. **(1pt)** True: Assume it is not second preimage resistant. Take any m and compute a second preimage m'. This algorithm builds a collision pair (m, m').
   ii. **(1pt)** False: It is not randomized, therefore cannot be IND-CPA secure. (Another way to write the same argument: With one pair (m, c), the adversary can obtain the key.)
   iii. **(1pt)** False: A hash function must be deterministic to be able use digests as fingerprints to test equality between two strings. (For instance, for hash-based MACs the verification is based on the recomputation of a hash.)

Exercise 2 (10 points). *CFB mode of operation*

1. **(1pt)** In pseudo-code: for $i = 1$ to t, $m_i \leftarrow c_i \oplus E_k(c_{i-1})$. *Beware: the decryption is not based on the inverse E_k^{-1} of E_k but on E_k itself.*
2. i. **(1pt)** Since the block cipher is itself deterministic, the IV must be random to get a randomized encryption scheme (which is necessary for the encryption scheme to be secure).
   ii. **(1.5pt)** We use padding. To be able to *unpad*, one idea is to use 10^s as padding, adding *at least* one bit (that is $s \geq 0$). The encryption computes $\hat{m} = m \| 10^s$ where s is such that $|\hat{m}|$ is a multiple of n and encrypts $\hat{m}$. The decryption decrypts to get $\hat{m}$ and removes the suffix of the form 10^s (from the right: remove all trailing zeroes, plus the first trailing one).
3. i. **(1.5pt)** Since $c_{i+1} = m_{i+1} \oplus E_k(c_i)$ and $c_{j+1} = m_{j+1} \oplus E_k(c_j)$, $c_{i+1} \oplus c_{j+1} = E_k(c_i) \oplus E_k(c_j) \oplus m_{i+1} \oplus m_{j+1}$. If $c_i = c_j$, $E_k(c_i) = E_k(c_j)$ (since E_k is deterministic) and the results follows.
   ii. **(1.5pt)** The block c_0 is uniform by assumption. Since E_k is a random permutation, $E_k(c_i)$ is uniform in $\{0,1\}^n$. As we have shown in the context of the one-time pad, a uniform random value XOR any other value remains uniform. Thus c_{i+1} is uniform for $i \geq 0$.
   iii. **(1.5pt)** The c_i's are uniform elements from $\{0,1\}^n$. Thus, if $t = \lfloor 2^{(n+1)/2} \rfloor$, the probability of a collision is $\geq t(t-1)/2^{n+2}$ according to the birthday bound. In particular, if $t = \lceil 1 + 2^{n/2} \rceil$, the probability is $\geq \frac{1}{4}$. *Note that $\lfloor 2^{(n+1)/2} \rfloor > \lceil 1 + 2^{n/2} \rceil$ as soon as $n \geq 5$.*
4. **(2pts)** The adversary creates one pair (m_0, m_1), where m_0 is only made of zeroes and m_1 is made of pairwise distinct blocks. Both are made of $1 + 2^{n/2}$ blocks. If the ciphertext blocks are pairwise distinct, the attack fails: Say the adversary returns always 0. Otherwise, assuming $c_i = c_j$, the adversary computes $c_{i+1} \oplus c_{j+1}$. If this is 0^n, it returns $\hat{b} = 0$ otherwise $\hat{b} = 1$.
   If there is a collision, we know that $c_{i+1} \oplus c_{j+1} = \hat{m}_{i+1} \oplus \hat{m}_{j+1}$ (where $\hat{m} = m_b$). Given the choice of m_0 and m_1, the result is either 0^n if $b = 0$, or something nonzero if $b = 1$. When $b = 0$, the probability that $\hat{b} = 1$ is 0 since in case of failure, the adversary returns 0. Hence, $\Pr\left[\hat{b} = 1 \middle| b = 0\right] = 0$. Next, when $b = 1$, $\hat{b} = 0$ if and only if there was no collision. This happens with probability $\leq \frac{3}{4}$, hence with probability $\geq \frac{1}{4}$, the adversary answers correctly.

Exercise 3 (6 points). *Multi-key security for MACs*

1. **(0.5pt)** The definitions coincide if $s = 1$.
2. i. **(1pt)** The adversary $\mathcal{A}$ forwards the query m_i to $\mathcal{C}$, and forwards the answer of $\mathcal{C}$ to $\mathcal{A}_{\mathsf{mk}}$.

ii. **(1pt)** The challenger $\mathcal{C}$ of $\mathcal{A}$ plays the role of $\mathcal{C}_j$, that is $\mathcal{C}$ uses the key k_j. Therefore, if $\mathcal{A}_{\mathsf{mk}}$ outputs a valid pair for k_j, this is a pair valid for $\mathcal{A}$'s game and $\mathcal{A}$ is successful.

iii. **(0.5pt)** Since there are s possible choices and the choice is uniform, the probability is $1/s$.

iv. **(1pt)** The adversary $\mathcal{A}$ is successful is the guess is correct (with prob. $\frac{1}{s}$) and if $\mathcal{A}_{\mathsf{mk}}$ is successful (the prob. is the advantage). Whence $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{Mac}}(\mathcal{A}) = \frac{1}{s}\mathsf{Adv}^{\mathsf{mk\text{-}EUF\text{-}CMA}}_{\mathsf{Mac}}(\mathcal{A}_{\mathsf{mk}})$.

3. **(2pts)** Assume that $\mathcal{A}_{\mathsf{mk}}$ is the best possible adversary: $\mathsf{Adv}^{\mathsf{mk\text{-}EUF\text{-}CMA}}_{\mathsf{Mac}}(q,s,t) = \mathsf{Adv}^{\mathsf{mk\text{-}EUF\text{-}CMA}}_{\mathsf{Mac}}(\mathcal{A}_{\mathsf{mk}})$. Then we built an adversary $\mathcal{A}$ that has advantage $\frac{1}{s}\mathsf{Adv}^{\mathsf{mk\text{-}EUF\text{-}CMA}}_{\mathsf{Mac}}(q,s,t)$. Therefore, the best possible adversary for the game EUF-CMA has at least the same advantage, that is $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{Mac}}(q,t) \geq \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{Mac}}(\mathcal{A}) = \frac{1}{s}\mathsf{Adv}^{\mathsf{mk\text{-}EUF\text{-}CMA}}_{\mathsf{Mac}}(q,s,t)$. (And this is for all s.)