UC Berkeley — CS 276 : Cryptography
Prof. Sanjam Garg

Midterm II Solutions

Midterm II
March 19, 2026

Name:

SID:

- **Time: You have *1 hour and 20 minutes* (80 minutes) to complete the exam. The exam runs from 11:10 AM to 12:30 PM. No extra time will be given after 12:30 PM.**
- **After the exam starts, write your name and SID on every odd-numbered page. We reserve the right to deduct points if you do not, and you will not be allowed to do so after time is called.**
- **For short questions, your answers must be written clearly inside the box region. Any answer outside the box will not be graded. For longer questions, if you run out of space, you must clearly mention in the space provided for the question if part of your answer is elsewhere.**
- **You may consult at most *1 double-sided sheet of handwritten notes*. Apart from that, you may not look at books, notes, etc. Calculators, phones, computers, and other electronic devices are NOT permitted for looking up content.**
- **There are 6 pages on the exam (counting this one). Notify a proctor immediately if a page is missing.**
- **We will not be answering questions during the exam. If you feel that something is unclear please write a note in your answer.**

1. **[25 points]** Let $\mathcal{H} = \{h_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{s\in\{0,1\}^k}$ be a UOWHF, where k is polynomial in n. Define the composed family

$$\mathcal{F} = \{f_{s_1,s_2,s_3} : \{0,1\}^{4n} \to \{0,1\}^n\}_{s_1,s_2,s_3\in\{0,1\}^k}$$

defined by

$$f_{s_1,s_2,s_3}(x_L\|x_R) = h_{s_3}(h_{s_1}(x_L)\|h_{s_2}(x_R)),$$

where $x_L, x_R \in \{0,1\}^{2n}$.

Is $\mathcal{F}$ necessarily a UOWHF? ☐
Either prove it or give a counterexample family.

Yes. Using independent seeds preserves universal one-wayness.
Suppose there is a PPT adversary $\mathcal{A}$ that breaks $\mathcal{F}$ with non-negligible probability. Let the target chosen by $\mathcal{A}$ be $X = x_L\|x_R$. After seeing (s_1, s_2, s_3), suppose $\mathcal{A}$ outputs $X' = x'_L\|x'_R \neq X$ such that

$$f_{s_1,s_2,s_3}(X) = f_{s_1,s_2,s_3}(X').$$

Write

$$u = h_{s_1}(x_L), \quad v = h_{s_2}(x_R), \quad u' = h_{s_1}(x'_L), \quad v' = h_{s_2}(x'_R).$$

Then one of the following must happen:

(a) $x'_L \neq x_L$ and $u' = u$, which is a collision for $\mathcal{H}$ under seed s_1.

(b) $x'_L = x_L$, so necessarily $x'_R \neq x_R$, and then $v' = v$, which is a collision for $\mathcal{H}$ under seed s_2.

(c) $(u', v') \neq (u, v)$, but

$$h_{s_3}(u\|v) = h_{s_3}(u'\|v'),$$

which is a collision for $\mathcal{H}$ under seed s_3 with target $u\|v$.

Hence whenever $\mathcal{A}$ succeeds, it yields a successful attack on at least one of the three cases. By averaging, one of these three events happens with non-negligible probability.

Reductions:

- For case 1, a reduction against $\mathcal{H}$ relays the challenge seed as s_1, samples fresh s_2, s_3, and if case 1 occurs outputs (x_L, x'_L).
- For case 2, a reduction against $\mathcal{H}$ samples s_1, s_3, relays the challenge seed as s_2, and if case 2 occurs outputs (x_R, x'_R).
- For case 3, a reduction against $\mathcal{H}$ first samples s_1, s_2, runs $\mathcal{A}$ until it commits to $X = x_L\|x_R$, then sets the target to be $u\|v = h_{s_1}(x_L)\|h_{s_2}(x_R)$. After receiving the challenge seed s_3, it gives (s_1, s_2, s_3) to $\mathcal{A}$ and, if case 3 occurs, outputs $(u\|v, u'\|v')$.

Each reduction is PPT and succeeds with non-negligible probability in its corresponding case, contradicting the UOWHF security of $\mathcal{H}$. Therefore $\mathcal{F}$ is a UOWHF.

2. **[25 points] Existential Unforgeability under Known Message Attack (EUF-KMA).** Let $\Pi = (\text{Gen}, \text{Sign}, \text{Verify})$ be a signature scheme. The EUF-KMA game $\mathsf{EUF\text{-}KMA}^{\mathcal{A}}_{\Pi}(n)$ proceeds as follows:

Game $\mathsf{EUF\text{-}KMA}^{\mathcal{A}}_{\Pi}(n)$

(a) Challenger samples $(vk, sk) \leftarrow \text{Gen}(1^n)$ and sends vk to $\mathcal{A}$.

(b) **Message Queries:** $\mathcal{A}$ sends messages $m^{(1)}, \ldots, m^{(q)}$ to the challenger. The challenger computes $\sigma^{(i)} \leftarrow \text{Sign}(sk, m^{(i)})$ for all $i \in \{1, \ldots, q\}$ and returns the signatures $(\sigma^{(1)}, \ldots, \sigma^{(q)})$ to $\mathcal{A}$.

(c) **Forgery:** $\mathcal{A}$ outputs (m^*, σ^*).

(d) $\mathcal{A}$ outputs 1 if $\text{Verify}(vk, m^*, \sigma^*) = 1 \wedge m^* \notin \{m^{(1)}, \ldots, m^{(q)}\}$ and 0 otherwise.

Definition (EUF-KMA). Π is *Existentially Unforgeable under Known Message Attack (EUF-KMA)* if for every PPT adversary $\mathcal{A}$

$$\Pr[\mathsf{EUF\text{-}KMA}^{\mathcal{A}}_{\Pi}(n) = 1]$$

is a negligible function in n.

Given a signature scheme $\Pi = (\text{Gen}, \text{Sign}, \text{Verify})$ that is EUF-KMA secure, and a hash function H (modeled as a random oracle), construct a signature scheme $\Pi' = (\text{Gen}', \text{Sign}', \text{Verify}')$ that is EUF-CMA secure in the Random Oracle model.

Construction of Π': Let $H : \{0,1\}^* \to \mathcal{M}$ be a random oracle, where $\mathcal{M}$ is the message space of Π.

- $\text{Gen}'(1^n)$: Run $(vk, sk) \leftarrow \text{Gen}(1^n)$. Output (vk, sk).
- $\text{Sign}'(sk, m)$: Compute $h \leftarrow H(m)$, and output $\sigma \leftarrow \text{Sign}(sk, h)$.
- $\text{Verify}'(vk, m, \sigma)$: Compute $h \leftarrow H(m)$, and output $\text{Verify}(vk, h, \sigma)$.

Proof of Security: Let $\mathcal{A}$ be a PPT EUF-CMA adversary making at most Q_H RO queries and q signing queries ($q \leq Q_H$). We construct an EUF-KMA adversary $\mathcal{B}$ that makes q queries:

(a) $\mathcal{B}$ samples q random values $r_1, \ldots, r_q \xleftarrow{\$} \mathcal{M}$. It queries them to its EUF-KMA challenger and receives signatures $\sigma_1, \ldots, \sigma_q$.

(b) $\mathcal{B}$ gives vk to $\mathcal{A}$. It picks a random index $i^* \xleftarrow{\$} \{1, \ldots, Q_H\}$ and initializes an empty table T to store RO responses.

(c) **RO Query on m:** If $T[m]$ is already defined, return $T[m]$. If this is the i^*-th unique RO query, select a fresh random hash $h^* \xleftarrow{\$} \mathcal{M}$, set $T[m] = h^*$, and return h^*. Otherwise, pick an untaken token r_j (or a fresh random value if no tokens are left), set $T[m] = r_j$, and return r_j.

(d) **Sign Query on m:** If m has not been queried to the RO, first simulate an RO query on m. Let $h = T[m]$. If $h = h^*$, $\mathcal{B}$ aborts. Otherwise, $h = r_j$ for some j. $\mathcal{B}$ returns the pre-obtained signature σ_j.

(e) **Forgery:** $\mathcal{A}$ outputs (m^*, σ^*). If m^* has not been queried to the RO, simulate an RO query. If $T[m^*] \neq h^*$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ outputs (h^*, σ^*) as its forgery in the EUF-KMA game.

Analysis: The RO simulation is perfect since all $T[m]$ values are uniform in $\mathcal{M}$. The signing simulation is perfect unless $\mathcal{B}$ aborts. Since m^* was never queried to the signing oracle, the event that m^*

corresponds to exactly the i^*-th RO query is independent of $\mathcal{A}$'s view. Thus, $\mathcal{B}$ does not abort with probability at least $1/Q_H$. Conditioned on not aborting, h^* was never queried to the EUF-KMA challenger, making (h^*, σ^*) a valid forgery. Thus $\mathbf{Adv}_{\Pi}^{\text{EUF-KMA}}(\mathcal{B}) \geq \frac{1}{Q_H}\mathbf{Adv}_{\Pi'}^{\text{EUF-CMA}}(\mathcal{A})$.

3. **[50 points] Combining Public Key Encryption Schemes.**
Let $\Pi_A = (\text{Gen}_A, \text{Enc}_A, \text{Dec}_A)$ and $\Pi_B = (\text{Gen}_B, \text{Enc}_B, \text{Dec}_B)$ be two public-key encryption (PKE) schemes. Assume that *at least one* of these schemes is IND-CPA secure (but not knowing which one).

   (a) **[15 points]** Construct an IND-CPA secure PKE scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$.

   Construction:

   - $\text{Gen}(1^n)$: Run $(pk_A, sk_A) \leftarrow \text{Gen}_A(1^n)$ and $(pk_B, sk_B) \leftarrow \text{Gen}_B(1^n)$. Output $pk = (pk_A, pk_B)$ and $sk = (sk_A, sk_B)$.
   - $\text{Enc}(pk, m)$: Sample a random string $r \xleftarrow{\$} \mathcal{M}$ (of the same length as m). Compute $c_A \leftarrow \text{Enc}_A(pk_A, r)$ and $c_B \leftarrow \text{Enc}_B(pk_B, m \oplus r)$. Output $c = (c_A, c_B)$.
   - $\text{Dec}(sk, c)$: Parse $c = (c_A, c_B)$. Compute $r' = \text{Dec}_A(sk_A, c_A)$ and $m' = \text{Dec}_B(sk_B, c_B)$. Output $r' \oplus m'$.

(b) **[20 points]** Prove that your construction from part (a) is IND-CPA secure.

Proof of Security: Without loss of generality, assume Π_A is IND-CPA secure. Let $\mathcal{A}$ be a PPT adversary against the IND-CPA security of Π. We construct a PPT adversary $\mathcal{B}$ against the IND-CPA security of Π_A as follows:

i. $\mathcal{B}$ receives pk_A from its challenger. It runs $(pk_B, sk_B) \leftarrow \text{Gen}_B(1^n)$ and gives $pk = (pk_A, pk_B)$ to $\mathcal{A}$.

ii. $\mathcal{A}$ outputs two messages m_0, m_1.

iii. $\mathcal{B}$ samples random strings $r, u \xleftarrow{\$} \mathcal{M}$. It outputs (r, u) to its IND-CPA challenger, receiving a ciphertext $c_A^* = \text{Enc}_A(pk_A, z_b)$, where z_b is either r or u.

iv. $\mathcal{B}$ flips a random coin $d \xleftarrow{\$} \{0,1\}$. It computes $c_B^* \leftarrow \text{Enc}_B(pk_B, m_d \oplus r)$. It gives $c^* = (c_A^*, c_B^*)$ to $\mathcal{A}$.

v. $\mathcal{A}$ outputs a guess d'. If $d' = d$, $\mathcal{B}$ outputs 0 (guessing $z_b = r$). Otherwise, $\mathcal{B}$ outputs 1 (guessing $z_b = u$).

Analysis: If the Π_A challenger chose $b = 0$, then $c_A^* = \text{Enc}(pk_A, r)$. The simulated ciphertext is $c^* = (\text{Enc}_A(pk_A, r), \text{Enc}_B(pk_B, m_d \oplus r))$, exactly as in the real game with bit d. Thus, $\mathcal{A}$ wins with probability $\frac{1}{2} + \mathbf{Adv}_{\Pi}^{\text{IND-CPA}}(\mathcal{A})$, and $\mathcal{B}$ outputs 0 with this probability.

If the Π_A challenger chose $b = 1$, then $c_A^* = \text{Enc}_A(pk_A, u)$. The simulated ciphertext is $c^* = (\text{Enc}_A(pk_A, u), \text{Enc}_B(pk_B, m_d \oplus r))$. Since u is chosen independently of r, r acts as a one-time pad perfectly hiding m_d. The distribution of c_B^* is identical regardless of whether $d = 0$ or $d = 1$. Thus, $\mathcal{A}$'s advantage is 0, meaning $\mathcal{A}$ outputs $d' = d$ with exactly probability $1/2$, so $\mathcal{B}$ outputs 1 with probability $1/2$.

Therefore, $\mathcal{B}$'s advantage in breaking Π_A is
$|\Pr[\mathcal{B} \to 1 \mid b = 1] - \Pr[\mathcal{B} \to 1 \mid b = 0]| = \left|\frac{1}{2} - \left(1 - (\frac{1}{2} + \mathbf{Adv}_{\Pi}^{\text{IND-CPA}}(\mathcal{A}))\right)\right| = \mathbf{Adv}_{\Pi}^{\text{IND-CPA}}(\mathcal{A})$.
Since Π_A is secure, this must be negligible. A symmetric argument applies if Π_B is secure (you can run both adversaries and one will break security of the secure scheme).
Another scheme that works is a double encryption $Enc_A(pk_A, Enc_B(pk_B, m))$ and can be proved by considering the two cases and describing a reduction in each case (depending on which scheme is the secure one).

(c) **[15 points]** Assuming both Π_A and Π_B are *IND-CCA2* secure, is your construction from part (a) IND-CCA2 secure? Justify your answer, and if not, demonstrate an attack.

No, it is not IND-CCA2 secure.

Attack: Let the challenge ciphertext be $c^* = (c_A^*, c_B^*) = (\text{Enc}_A(pk_A, r^*), \text{Enc}_B(pk_B, m_b \oplus r^*))$. The adversary can use the decryption oracle to recover m_b by performing a "mix-and-match" attack.

i. The adversary creates an encryption of 0 under Π_B: $c'_B \leftarrow \text{Enc}_B(pk_B, 0)$.

ii. They query the decryption oracle with $c_1 = (c_A^*, c'_B)$. This is allowed because $c_1 \neq c^*$. The oracle decrypts c_A^* to r^* and c'_B to 0, returning $r^* \oplus 0 = r^*$. The adversary now knows the random share r^*.

iii. The adversary creates an encryption of 0 under Π_A: $c'_A \leftarrow \text{Enc}_A(pk_A, 0)$.

iv. They query the decryption oracle with $c_2 = (c'_A, c_B^*)$. The oracle decrypts c'_A to 0 and c_B^* to $m_b \oplus r^*$, returning $0 \oplus (m_b \oplus r^*) = m_b \oplus r^*$.

v. The adversary computes $r^* \oplus (m_b \oplus r^*) = m_b$, fully recovering the encrypted message and winning the IND-CCA2 game with probability 1.

Note that this answer may be different depending on the construction; it would be secure if you used a double encryption, but most constructions that use two components in the final output are insecure due to the above mix-and-match style attack.