Midterm Solution

Grading: 35 points for Q1, 40 points for Q2, 25 points for Q3 (out of a total of up to 100).

1 Generic Composition

(a) $\boxed{\text{True}}$: If (E, D) and (E', D') are both IND-CCA2, then $(\hat{E}, \hat{D})$ is guaranteed to be IND-CCA2.

Proof. We use the left-or-right formalization of IND-CCA2 security (since this is equivalent to the real-or-random formalization, we can use whichever is more convenient). Let $\hat{A}$ be a left-or-right style IND-CCA2-adversary that breaks $(\hat{E}, \hat{D})$. We show that we can use $\hat{A}$ to break (E', D'). In particular, define the adversary B as follows:

$B^{e,d}$:

1. Pick $k \xleftarrow{\$} \mathcal{K}$.
2. Run $\hat{A}$, answering $\hat{A}$'s oracle queries as follows:
   On query (ℓ, r) to $\hat{A}$'s encryption oracle, answer with $e(E_k(\ell), E_k(r))$.
   On query c to $\hat{A}$'s decryption oracle, compute $c' = d(c)$, and answer with $\perp$ if $c' = \perp$, else answer with $D_k(c')$.
3. Return whatever $\hat{A}$ does.

By the choice of B, we have

$$\Pr[k' \xleftarrow{\$} \mathcal{K}'; B^{E'_{k'} \circ \S_b, D'_{k'}} = 1] = \Pr[(k, k') \xleftarrow{\$} \mathcal{K} \times \mathcal{K}'; A^{\hat{E}_{k,k'} \circ \S_b, \hat{D}_{k,k'}} = 1].$$

If $\hat{A}$ never queries its decryption oracle with the response from any prior query to its encryption oracle, then the same is true of B. Consequently, if $\hat{A}$ is a valid left-or-right style IND-CCA2-adversary that breaks $(\hat{E}, \hat{D})$ with advantage ϵ, then B is a valid left-or-right style IND-CCA2-adversary that breaks (E', D') with the same advantage, the same number of queries, and only slightly more running time. □

Common mistakes. Most people who tried to prove the theorm using the real-or-random formalization succumbed to one of two fallacies:

- Failed reduction #1: Given an attack A against $(\hat{E}, \hat{D})$, build an attack B against (E, D) in the obvious way. The problem with this approach is that B might fail to be a valid IND-CCA2 attack: it might query its decryption oracle D_k on a ciphertext that was previously returned by its encryption oracle E_k. In particular, B can be invalid even if A is valid. The reason for this is that there can be many E'-ciphertexts corresponding to one plaintext, so even if c isn't a repeat of any previous output from $\hat{E}_{k,k'}$, $D'_{k'}(c)$ might be a repeat of a previous output from E_k.
- Failed reduction #1: Given an attack A against $(\hat{E}, \hat{D})$, build an attack B against (E', D') in the obvious way. The problem you run into is that the outputs from E_k aren't necessarily random-looking; it's not true that $\$(E_k(m)) \sim E_k(\$(m))$. (For instance, consider (E, D)

given by $E_k(m) = (0, \mathcal{E}_k(m))$, $D_k(0, c) = \mathcal{D}_k(c)$, $D_k(1, c) = \perp$, where $(\mathcal{E}, \mathcal{D})$ is any IND-CCA2 secure cryptosystem. Then (E, D) is IND-CCA2, but the ciphertexts from E aren't random looking.)

Grading: 15 pts for a correct proof. 10 pts for one of the above failed reductions, if there were no other oversights or errors. 5 pts for correctly answering True without valid proof.

(b) $\boxed{\text{False}}$: If (E, D) is IND-CCA2 and (E', D') is IND-CPA, then $(\hat{E}, \hat{D})$ is guaranteed to be IND-CCA2.

Proof. Here is a counterexample. Suppose (E, D) is IND-CCA2. Let $E'_{k'}(m) = (0, E_{k'}(m))$ and $D'_{k'}(c) = D_{k'}(c_1)$, where $c = (c_0, c_1)$. Then (E', D') is IND-CPA. However, $(\hat{E}, \hat{D})$ is not IND-CCA2, as the following attack shows:

$A^{f,g}$:
1. Choose any 100-bit message m. Set $(c_0, c_1) \leftarrow f(m)$ and $c' \leftarrow (1, c_1)$.
2. If $g(c') = m$ then output 1 else output 0.

Note that A is a legitimate adversary, because it never queries its second oracle on the output of any query to its first oracle (since we always have $c_0 = 0$ for the definition of $\hat{E}$ given above). Also, we have $\Pr[A^{\hat{E}_{k,k'}, \hat{D}_{k,k'}} = 1] = 1$ and $\Pr[A^{\hat{E}_{k,k'} \circ \$, \hat{D}_{k,k'}} = 1] = 1/2^{100}$, so A successfully breaks $(\hat{E}, \hat{D})$ with advantage $1 - 1/2^{100}$. $\square$

Grading: 10 pts for a valid counterexample. 5 pts for correctly answering False without valid proof.

(c) $\boxed{\text{True}}$: If (E, D) is IND-CCA2 and (E', D') is INT-CTXT, then $(\hat{E}, \hat{D})$ is guaranteed to be IND-CCA2.

Proof. If (E, D) is IND-CCA2, then it is also trivially IND-CPA (since any IND-CPA-adversary can also be considered as an IND-CCA2-adversary that does not use its decryption oracle). We proved in class that if (E, D) is IND-CPA and (E', D') is INT-CTXT, then their generic composition is IND-CCA2. $\square$

Grading: 10 pts for a valid proof. 5 pts for correctly answering True without valid proof.

2 Bit Commitment in the Random Oracle Model

(a) What's wrong with the following scheme? $\mathcal{M} = \{0, 1\}$; $\text{Commit}^{\mathcal{H}}(m, r) = \mathcal{H}(m)$; $\text{Verify}^{\mathcal{H}}(c, m, r) = 1$ if $\mathcal{H}(m) = c$, and 0 otherwise. Is it binding? concealing?

Answer: This scheme is binding but not concealing, and consequently is not a secure bit commitment scheme. It is not concealing, as the following adversary shows:

$A^{\mathcal{H}}(c)$:
1. If $c = \mathcal{H}(0)$, then output 0, else output 1.

Note that $A^{\mathcal{H}}(\text{Commit}(m, r))$ always outputs m (unless $\mathcal{H}(0) = \mathcal{H}(1)$, but that has exponentially small probability), so A has advantage close to 1 at breaking the concealing property. This scheme is computationally $(t, t^2/2^{n+1})$-binding: the probability that an adversary can find a pair m, m' such that $\mathcal{H}(m) = \mathcal{H}(m')$ but $m \neq m'$ using at most t $\mathcal{H}$-queries is (by the birthday paradox) at most $t^2/2^{n+1}$, assuming that $\mathcal{H}$ produces n-bit outputs. This example illustrates that the commit algorithm must be randomized, for much the same reason that encryption must be randomized if we want it to be IND-CPA-secure.

Grading: 5 pts for correctly justifying that it is binding; 5 pts for correctly justifying that it is not concealing.

(b) Specify a pair of algorithms (Commit, Verify) that form a bit commitment scheme that is (t, ϵ)-secure in the random oracle model, for some reasonable values of (t, ϵ) (say, $t = 2^{80}$, $\epsilon = 2^{-64}$). Specify your choice of the sets $\mathcal{M}$ and $\mathcal{R}$ and any assumptions you are making about the length of the output of $\mathcal{H}$.

Answer: One possible construction is $\text{Commit}(m, r) = \mathcal{H}(r||m)$, where $||$ denotes concatenation of bitstrings, $\mathcal{M} = \{0, 1\}^*$, $\mathcal{R} = \{0, 1\}^{161}$, and the hash $\mathcal{H}$ is assumed to produce 256-bit outputs. Also, define

$$\text{Verify}(c, m, r) = \begin{cases} 1 & \text{if } \mathcal{H}(r||m) = c \\ 0 & \text{otherwise.} \end{cases}$$

Grading: 5 pts for a secure commitment scheme.

(c) State explicitly the values of (t, ϵ) for which you are claiming your scheme from part (b) is secure.

Answer: The scheme in part (b) is $(t, t/2^{160})$-concealing and $(t, t^2/2^{257})$-binding. For instance, for $t = 2^{80}$, the scheme is $(2^{80}, 1/2^{80})$-secure.

(d) Prove that your scheme from part (b) is secure with the parameters listed in part (c).

Theorem 1. *The scheme (Commit, Verify) defined in part (b) is $(t, t^2/2^{257})$-binding.*

Proof. Assume without loss of generality that if A produces (c, m, r, m', r') as output, then it has previously queried both $r||m$ and $r'||m'$ to its oracle. The adversary A can make at most t queries to its oracle, since each query takes one unit of time. Assume without loss of generality that these $\mathcal{H}$-queries are all distinct. The responses are uniform random 256-bit strings, all independent of each other. By the birthday paradox, the chances that some pair of responses collides is at most $t^2/2^{257}$. Moreover, if no pair of responses collides, then A has surely failed to break the binding property. Consequently,

$$\Pr[(c, m, r, m', r') \leftarrow A^{\mathcal{H}}(); \text{Verify}^{\mathcal{H}}(c, m, r) = 1 \text{ and } \text{Verify}^{\mathcal{H}}(c, m', r') = 1 \text{ and } m \neq m'] \leq t^2/2^{257}.$$

□

Theorem 2. *The scheme (Commit, Verify) defined in part (b) is $(t, t/2^{160})$-concealing.*

Proof. Fix m, m'. Without loss of generality, I will assume that A is deterministic. If A is randomized, we can determinize it by fixing a sequence of coins that maximizes A's conditional success probability; it is easy to see that this cannot reduce A's advantage. Also without loss of generality, we may assume that if A makes a $\mathcal{H}$-query of the form $x||m$, it also makes the query $x||m'$, and vice versa. If the attacker does not satisfy this assumption, it can be transformed into one that does by introducing extra queries and ignoring the responses from these extra queries. This transformation may double the number of $\mathcal{H}$-queries, so I assume that A makes at most $2t$ $\mathcal{H}$-queries.

Let Bad denote the event that, when choosing $r \stackrel{\$}{\leftarrow} \mathcal{R}$ and then executing $A^{\mathcal{H}}(\mathcal{H}(r||m))$, at some point A queries its oracle with the value $r||m$. Also let Bad' denote the event that, when choosing $r \stackrel{\$}{\leftarrow} \mathcal{R}$ and then executing $A^{\mathcal{H}}(\mathcal{H}(r||m'))$, at some point A queries its oracle with the value $r||m'$. Note that under the above assumption, the events Bad and Bad' coincide.

I claim that $\Pr[\mathsf{Bad}] \le 2t/|\mathcal{R}| = t/2^{160}$. This follows because, so long as Bad remains false, the responses to A's $\mathcal{H}$-queries are uniform random strings chosen independently of r. Consequently, if Bad is not yet true and A makes one more query, it has no information about r, so the chances that A makes Bad become true on this query is at most $1/|\mathcal{R}|$. The claim follows by a union bound and using the fact that A makes at most $2t$ $\mathcal{H}$-queries.

Next, I claim that

$$\Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathcal{H}(r||m)) = 1 \text{ and } \neg\mathsf{Bad}] = \Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathcal{H}(r||m')) = 1 \text{ and } \neg\mathsf{Bad}].$$

This can be justified by examining the mapping $\varphi_r : h \mapsto \overline{h}$, where

$$\overline{h}(x) = \begin{cases} h(r||m) & \text{if } x = r||m' \\ h(r||m') & \text{if } x = r||m \\ h(x) & \text{otherwise.} \end{cases}$$

In particular, φ_r is a bijection that sends hash functions h where $A^h(h(r||m))$ outputs 1 and $\neg\mathsf{Bad}$ holds to hash functions $\overline{h}$ where $A^{\overline{h}}(\overline{h}(r||m'))$ outputs 1 and $\neg\mathsf{Bad}$ holds, and vice versa. Consequently, the equality shown above holds. It follows that

$$\Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathcal{H}(r||m)) = 1|\neg\mathsf{Bad}] = \Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathcal{H}(r||m')) = 1|\neg\mathsf{Bad}].$$

Finally, we can compute

$$\left|\Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathrm{Commit}^{\mathcal{H}}(m, r)) = 1] - \Pr[r' \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathrm{Commit}^{\mathcal{H}}(m', r')) = 1]\right| \le t/2^{160}$$

by a standard bad-event analysis. □

Common mistakes: Without the assumption that A queries $x||m$ iff it queries $x||m'$, the proof above falls apart. For instance, consider this attack: $A(c)$ tries many values v, computes $\mathcal{H}(v||m)$ using one $\mathcal{H}$-query, checks whether $\mathcal{H}(v||m) \stackrel{?}{=} c$, and outputs 1 if this equality ever holds, or 0 otherwise. Note that $\Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathcal{H}(r||m)) = 1] \approx t/2^{161}$, while $\Pr[r \stackrel{\$}{\leftarrow} \mathcal{R}; A^{\mathcal{H}}(\mathcal{H}(r||m')) = 1] \approx t/2^{256}$. Even if we condition on the event that A never queries $r||m$ to its $\mathcal{H}$-oracle, A still gains some non-zero advantage at distinguishing $\mathcal{H}(r||m)$ from $\mathcal{H}(r||m')$ (indeed, conditioning on this event makes it slightly more likely that the input was of the latter form than of the former form).

Grading: 10 pts for a valid proof that your scheme is binding; 15 pts for concealing.

3 Block Ciphers on Smaller Domains

Let R denote the set of 128-bit strings whose last bit is 0. We define a block cipher $E : \mathcal{K} \times R \to R$ by

$$E_k(x) = \begin{cases} \mathrm{AES}_k(x) & \text{if } \mathrm{AES}_k(x) \in R \\ E_k(\mathrm{AES}_k(x)) & \text{otherwise.} \end{cases}$$

In other words, the value of $E_k(x)$ is defined to be the first value from the sequence $\mathrm{AES}_k(x)$, $\mathrm{AES}_k(\mathrm{AES}_k(x))$, $\ldots$ that falls in the set R. Note that we can view E as a block cipher with signature $E : \mathcal{K} \times \{0,1\}^{127} \to \{0,1\}^{127}$ by using the canonical isomorphism $\{0,1\}^{127} \cong R$. Also the expected number of AES-invocations needed to compute $E_k(x)$ is exactly two, so E can be computed efficiently.

We will prove this secure by proving a more general lemma about permutations. Let R, S denote two sets with $R \subseteq S$. If π is a permutation on S, i.e., a bijective map $\pi : S \to S$, then we define the permutation $\overline{\pi}$ on R by

$$\overline{\pi}(x) = \begin{cases} \pi(x) & \text{if } \pi(x) \in R \\ \overline{\pi}(\pi(x)) & \text{otherwise.} \end{cases}$$

In other words, we have a mapping $\pi \mapsto \overline{\pi}$ that sends permutations on S to permutations on R. Let $\mathsf{Perm}(S)$ denote the set of all permutations on S.

Lemma 1. *If $R \subseteq S$ and $|R| = |S| - 1$ and the random variable π is uniformly distributed on $\mathsf{Perm}(S)$, then the random variable $\overline{\pi}$ has the uniform distribution on $\mathsf{Perm}(R)$.*

Proof. Let $n = |S|$. Define the function $f : \mathsf{Perm}(S) \to \mathsf{Perm}(R) \times \{1, \ldots, n\}$ by $f(\pi) = (\overline{\pi}, \pi^{-1}(n))$. We prove that f is in fact a bijection. In particular, the unique pre-image of (σ, i) under f is the permutation γ given by

$$\gamma(x) = \begin{cases} n & \text{if } x = i \\ \sigma(i) & \text{if } x = n \text{ and } i \neq n \\ \sigma(x) & \text{otherwise.} \end{cases}$$

It is easily seen that, for this choice of γ, $\overline{\gamma} = (\sigma, i)$, so it follows that f is a bijection. It follows that the map $\pi \mapsto \overline{\pi}$ is a n-to-1 mapping (every permutation $\sigma \in \mathsf{Perm}(R)$ has exactly n choices of π such that $\overline{\pi} = \sigma$, namely, the choices $f^{-1}(\sigma, 1), \ldots, f^{-1}(\sigma, n)$). Consequently, if π is uniform on $\mathsf{Perm}(S)$, then $\overline{\pi}$ is uniform on $\mathsf{Perm}(R)$. $\square$

Lemma 2. *If $R \subseteq S$ and the random variable π is uniformly distributed on $\mathsf{Perm}(S)$, then the random variable $\overline{\pi}$ has the uniform distribution on $\mathsf{Perm}(R)$.*

Proof. Choose any sequence of sets $R = S_0 \subset S_1 \subset \cdots \subset S_n = S$ such that $|S_{i+1}| = |S_i| + 1$ for each i. Define a sequence of permutations $\sigma_0, \ldots, \sigma_n = \pi$, where

$$\sigma_i(x) = \begin{cases} \sigma_{i+1}(x) & \text{if } \sigma_{i+1}(x) \in S_i \\ \sigma_i(\sigma_{i+1}(x)) & \text{otherwise.} \end{cases}$$

Invoking Lemma 1 n times, we see that each σ_i is uniformly distributed on $\mathsf{Perm}(S_i)$. Moreover, $\overline{\pi} = \sigma_0$. The result follows. □

Theorem 3. *If AES is a secure PRP, then E (as defined above) is a secure PRP.*

Proof. I will prove that if AES is a (t, q, ϵ)-secure PRP, then E is a (t, q', ϵ')-secure PRP, for some q', ϵ' with (very roughly speaking) $q' \approx 2q$ and $\epsilon' \approx \epsilon$. I will not attempt to quantify q', ϵ' in any exact or rigorous way.

Let $S = \{0,1\}^{128}$, R be the set of 128-bit strings whose last bit is 0, and $\Pi \xleftarrow{\$} \mathsf{Perm}(S)$. By assumption, AES_k is indistinguishable from Π:

$$\left|\Pr[k \xleftarrow{\$} \mathcal{K}; A^{\mathrm{AES}_k(\cdot)} = 1] - \Pr[\Pi \xleftarrow{\$} \mathsf{Perm}(S); A^{\Pi(\cdot)} = 1]\right| \le \epsilon$$

for all adversaries A running in time at most t and making at most q queries. Note that $E_k = \overline{\mathrm{AES}_k}$. Also, by Lemma 2, $\overline{\Pi}$ is uniformly distributed on $\mathsf{Perm}(R)$. Moreover, we can easily and efficiently compute $\overline{\varphi}(x)$ given oracle access to φ, even if we don't know what the function φ is. Consider any adversary B that tries to distinguish E_k from $\Pi' \xleftarrow{\$} \mathsf{Perm}(R)$. Define the adversary A by $A^{\varphi} = B^{\overline{\varphi}}$; in other words, A^{φ} runs B, answering B's oracle query x by computing the sequence $\varphi(x), \varphi(\varphi(x)), \ldots$ and responding with the first value that falls in R. We see that

$$\begin{aligned} &\left|\Pr[k \xleftarrow{\$} \mathcal{K}; B^{E_k(\cdot)} = 1] - \Pr[\Pi' \xleftarrow{\$} \mathsf{Perm}(R); B^{\Pi'(\cdot)} = 1]\right| \\ &\quad = \left|\Pr[k \xleftarrow{\$} \mathcal{K}; A^{\mathrm{AES}_k(\cdot)} = 1] - \Pr[\Pi \xleftarrow{\$} \mathsf{Perm}(R); A^{\Pi(\cdot)} = 1]\right| \le \epsilon. \end{aligned}$$

The number of queries issued by A is likely to be not much more than twice the number of queries issued by B (this can be quantified more rigorously using a PRF-to-PRF lemma and a union bound, but I won't bother). □

Common mistakes: Many people suggested taking $E_k(x)$ to be the low 127 bits of $\mathrm{AES}_k(0||x)$ (i.e., prepending a single 0 bit, AES-encrypting, and dropping the most significant bit of the ciphertext) and proved correctly that this E is a pseudorandom function. The problem is that while this E is indeed a PRF, it is not a PRP—indeed, it is not even a permutation (the block cipher E cannot be decrypted). A few folks tried to argue that the fact that E isn't a permutation doesn't matter, because the chances that the adversary notices this fact is bounded; however, this is problematic, because there is still no way to decrypt. If we want to use E in a mode of operation that requires the inverse operation E^{-1}, then we won't be able to plug in this choice of E. For instance, we cannot use this E in CBC mode, because there will be no way to decrypt. This was almost certainly the hardest problem on the exam.

Grading: 25 pts for a PRP along a valid proof. 10 pts for a scheme that isn't a permutation, if you provided a valid proof that it is a PRF.