CS 258: Quantum Cryptography (Fall 2025)
Homework 1 (100 points)

1 Problem 1 (20 points)

Recall that CPA security is defined using negligible functions. Informally, a function is negligible if it is smaller than any inverse polynomial. There are a few ways to define negligible functions. Here is the most common one:

Definition 1. *A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if, for any polynomial p, there exists an $N_p \in \mathbb{N}$ such that $f(\lambda) < 1/p(\lambda)$ for all $\lambda \geq N_p$*

Part (a). 12 points. Let p be an arbitrary polynomial function, and f, g be two negligible functions. Prove the following (3 points each):

(i) $f + g$ is negligible

(ii) pf is negligible

(iii) $1/p - f$ is *not* negligible

(iv) If $h(\lambda)$ is a function such that there exists an $N \in \mathbb{N}$ such that $h(\lambda) \leq f(\lambda)$ for all $\lambda \geq N$, then $h(\lambda)$ is also negligible. In other words, any function smaller than a negligible function is also negligible.

Part (b). 6 points. For each of the following functions, decide if they are negligible or not, with a brief (informal is acceptable) explanation for why. (1 point each)

(i) $f_i(\lambda) = 100\lambda \times 2^{-\lambda}$

(ii) $f_{ii}(\lambda) = 7^{-\ln(\lambda)}$

(iii) $f_{iii}(\lambda) = \begin{cases} 1/\lambda & \text{if } \lambda \text{ is even} \\ 1/2^\lambda & \text{if } \lambda \text{ is odd} \end{cases}$

(iv) $f_{iv}(\lambda) = \lambda^{-\log_2(\log_2(\log_2(\lambda)))}$

(v) $f_v(\lambda) = 2^{-\sqrt{\log_2(\lambda)}}$

(vi) $f_{vi}(\lambda) = \lambda^{-1} - 100 \times 2^{-\lambda}$

Part (c). 2 points Give an example of a negligible function f and a *non*-negligible function g such that f/g is non-negligible.

OPTIONAL BONUS: Part (d). 3 points. Let $f_1, f_2, \cdots$ be an infinite sequence of negligible functions. Define $F(\lambda) = \sum_{i=1}^{\lambda} f_i(\lambda)$. One might expect that, since adding negligible functions produces negligible functions, $F(\lambda)$ is negligible. However, this is in general not the case. Give a sequence of functions $f_1, f_2, \cdots$, each of which are negligible, but for which $F(\lambda)$ is non-negligible.

Remark 2. *Note that by iteratively applying Problem 1.a.i, the sum of any constant number of negligible functions is negligible. However, once the number of functions becomes non-negligible, the sum is technically not guaranteed to be negligible.*

2 Problem 2 (30 points)

Consider the following variant of CPA security for public key encryption.

Definition 3 (Public key encryption, CPA security)**.** *For functions $q_1(\lambda), q_2(\lambda)$, a public key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is (q_1, q_2)-CPA-secure if, for all probabilistic polynomial time interactive algorithms $\mathcal{A}$, there exists a "negligible" function $\epsilon(\lambda) \leq \lambda^{-\omega(1)}$ such that $|\Pr[W_0(\lambda) = 1] - \Pr[W_1(\lambda) = 1]| \leq \epsilon(\lambda)$, where W_b is the event that $\mathcal{A}$ outputs 1 in the following experiment:*

- *Run* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, and give* pk *to $\mathcal{A}$.*
- *$\mathcal{A}$ now can make a polynomial number of queries. The queries come in one of two types:*
  - **Encryption Query.** *Here, $\mathcal{A}$ sends an arbitrary message $m \in \{0,1\}^*$, and in response receives $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$.*
  - **Challenge Query.** *Here, $\mathcal{A}$ sends two messages $m_0^*, m_1^* \in \{0,1\}^*$* of the same length*. In response, $\mathcal{A}$ receives $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b^*)$.*

  The queries can happen in any order, except that the total number of encryption queries is at most $q_1(\lambda)$, and the total number of challenge queries is at most $q_2(\lambda)$.
- *$\mathcal{A}$ outputs a bit b'.*

We call $|\Pr[W_0(\lambda) = 1] - \Pr[W_1(\lambda) = 1]|$ the advantage *of $\mathcal{A}$. We say that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* (poly, q_2)*-CPA-secure if it is (q_1, q_2)-CPA secure* for any *polynomial q_1. We likewise define (q_1, poly)-CPA-secure and* $(\mathsf{poly}, \mathsf{poly})$*-CPA-secure.*

Observe that the definition we saw in class is the same as $(0, 1)$-CPA-secure.

Part (a). 5 points. Show that *any* public key encryption scheme is $(\mathsf{poly}, 0)$-CPA secure.

Part (b). 5 points. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a $(\mathsf{poly}, \mathsf{poly})$-CPA-secure public key encryption scheme, but which is only capable of encrypting a single bit (as opposed to arbitrary bit strings). A simple way to extend it to arbitrary bit-strings is to encrypt bit-by-bit: $\mathsf{Enc}'(\mathsf{pk}, m) = (\mathsf{Enc}(\mathsf{pk}, m_0), \mathsf{Enc}(\mathsf{pk}, m_1), \cdots, \mathsf{Enc}(\mathsf{pk}, m_\ell))$ where $m = (m_1, \cdots, m_\ell) \in \{0,1\}^\ell$. Decryption Dec' is then performed piece-by-piece as well. Show that $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$ is also $(\mathsf{poly}, \mathsf{poly})$-CPA-secure. To do so, start with any supposed adversary $\mathcal{A}$ for the $(\mathsf{poly}, \mathsf{poly})$-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$, and build from $\mathcal{A}$ an adversary $\mathcal{B}$ for the $(\mathsf{poly}, \mathsf{poly})$-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The advantage of $\mathcal{B}$ should be equal to the advantage of $\mathcal{A}$. By the $(\mathsf{poly}, \mathsf{poly})$-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, the advantage of $\mathcal{B}$ is negligible, and thus so is the advantage of $\mathcal{A}$.

Part (c). 5 points. Show that if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is (q_1, q_2)-CPA-secure for any polynomials q_1, q_2, then it is also (q_1', q_2')-CPA-secure, for any $q_1'(\lambda) \leq q_1(\lambda)$ and $q_2'(\lambda) \leq q_2(\lambda)$. To do so, start with any supposed adversary $\mathcal{A}$ for the (q_1', q_2')-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, and build from $\mathcal{A}$ an adversary $\mathcal{B}$ for the (q_1, q_2)-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The advantage of $\mathcal{B}$ should be equal to the advantage of $\mathcal{A}$. By the (q_1, q_2)-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, the advantage of $\mathcal{B}$ is negligible, and thus so is the advantage of $\mathcal{A}$. Your proof should work for any possible polynomials q_1, q_2, q_1', q_2' as long as $q_1'(\lambda) \leq q_1(\lambda)$ and $q_2'(\lambda) \leq q_2(\lambda)$.

Part (d). 5 points. Show that if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is $(0, q_2)$-CPA-secure, then it is (q_1, q_2)-CPA-secure for any polynomial q_1 (and in particular, that it is (poly, q_2)-CPA-secure). To do so, start with any polynomial q_1 and any supposed adversary $\mathcal{A}$ for the (q_1, q_2)-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, and build from $\mathcal{A}$ an adversary $\mathcal{B}$ for the (q_1, q_2)-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Part (e). 10 points. Show that if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is $(q_1, 1)$-CPA-secure, then it is $(q_1 - q_2 + 1, q_2)$-CPA-secure for any polynomial $q_1 \geq q_2$ (and in particular, that it is $(0, \mathsf{poly})$-CPA-secure). To do so, you will use something called a *hybrid argument.* Let q_2 be any polynomial and $\mathcal{A}$ any supposed adversary for the $(q_1 - q_2 + 1, q_2)$-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Then, for $i = 0, \cdots, q_2(\lambda)$, define "hybrid experiments" H_i which are identical to the experiment in Definition 3, except where challenge queries are answered as follows:

- **Challenge Query.** Here, $\mathcal{A}$ sends two messages $m_0^*, m_1^* \in \{0,1\}^*$ *of the same length.* Suppose the current query is the jth challenge query. In response, $\mathcal{A}$ receives

$$c \leftarrow \begin{cases} \mathsf{Enc}(\mathsf{pk}, m_1^*) & \text{if } j < i \\ \mathsf{Enc}(\mathsf{pk}, m_b^*) & \text{if } j = i \\ \mathsf{Enc}(\mathsf{pk}, m_0^*) & \text{if } j > i \end{cases}$$

Define $W_i'(\lambda)$ as the probability that $\mathcal{A}$ outputs 1 in Hybrid H_i. First, explain that $W_0'(\lambda) = W_0$, $W_{q_2(\lambda)}'(\lambda) = W_1$.

Next, show that for each $i = 1, \cdots, q_2(\lambda)$, $|W_i'(\lambda) - W_{i-1}'(\lambda)|$ must be negligible: to do so, devise an algorithm $\mathcal{B}_i$ for the $(q_1, 1)$-CPA-security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, and show that the advantage of $\mathcal{B}_i$ is exactly $|W_i'(\lambda) - W_{i-1}'(\lambda)|$.

Finally, use the triangle inequality to conclude that $|W_0(\lambda) - W_1(\lambda)|$ is negligible. For this, you may assume that the sum of polynomially-many negligible functions is still negligible.

Remark 4. *Technically, the above proof does not work (see the bonus question Problem 1d), since summing a non-constant number of negligible functions is not necessarily negligible. However, the proof can be made to work by actually defining a single adversary $\mathcal{B}$ which chooses a random i and runs $\mathcal{B}_i$. There will be a single negligible advantage function for this $\mathcal{B}$, which can be shown to be equal to the advantage of $\mathcal{A}$ divided by $q_2(\lambda)$. By Problem 1.a.ii, we can then conclude the advantage of $\mathcal{A}$ is actually negligible. This subtle issue is almost never a real problem, and is largely ignored even in serious cryptography publications since it is simpler to just consider $q_2(\lambda)$ adversaries $\mathcal{B}_i$. We will ignore it as well in this course, and simply pretend that the sum of polynomially-many negligible functions is still negligible.*

Remark 5. *Typically,* $(\mathsf{poly}, \mathsf{poly})$-*CPA-secure encryption is what is actually desired in practice. Problem 2 shows that, for public key encryption, it suffices to consider the definition of CPA-security seen in class* ($(0,1)$-*CPA-secure), as it is equivalent to the desired notion. In this course, we will therefore largely focus on* $(0,1)$-*CPA-security. It also shows that it suffices to consider encryption for single-bit messages, which can then be generically extended to many bits.*

3 Problem 3 (20 points)

It is straightforward to generalize public key encryption as defined in class as well as Definition 3 to the case of symmetric key cryptosystems. Here, there is no Gen, and distinct public and secret key. Instead, $\mathsf{Enc}, \mathsf{Dec}$ take as input a key $k \in \{0,1\}^\lambda$, and security is defined by choosing k uniformly random in $\{0,1\}^\lambda$.

Part (a). 5 points. Recall the one-time pad discussed in class, where $\mathsf{Enc}(k, m) = k \oplus m$ and $\mathsf{Dec}(k, c) = k \oplus c$. Here, keys, messages, and ciphertexts are all n bits for some integer n. Show that the one-time pad is $(0,1)$-CPA-secure as a symmetric key encryption scheme.. *Hint: what is the distribution of* $\mathsf{Enc}(k, m)$ *for a random choice of* k*?*

Part (b). 5 points. Show that the one-time pad is not even $(0,2)$-CPA-secure.

Part (c). 5 points. In Problem 2, you showed that for public key encryption, $(0,1)$-CPA-security is equivalent to $(\mathsf{poly},\mathsf{poly})$-CPA-security, and in particular implies $(0,2)$-CPA-security. But for the one-time pad, and therefore symmetric key encryption in general, this is not true. Explain why.

Part (d). 5 points. Generalize part (2) to show that any $(\mathsf{Enc},\mathsf{Dec})$ where Enc is *deterministic*[1] cannot be even $(0,2)$-CPA-secure. (Note that this also applies to public key encryption schemes)

4 Problem 4 (30 points)

For each of the following, determine the output distribution resulting from measuring the state. That is, for each possible measurement outcome, determine the probability measuring the state produces that outcome. Express all probabilities as rational numbers. (5 points each)

(a) $|\psi_a\rangle = \frac{3}{5}|0\rangle + \frac{4}{5}|1\rangle$.

(b) $|\psi_b\rangle = \frac{3}{5}|0\rangle + \frac{4}{5}e^{-i\pi/3}|1\rangle$.

(c) $|\psi_c\rangle = \frac{1}{2}\left(e^{i\pi/3} - e^{-i\pi/3}\right)|0\rangle + \alpha|2\rangle$. Here, α is the unique positive real number such that $|\psi_c\rangle$ is a valid quantum state.

(d) $|\psi_d\rangle = \mathbf{H}|\psi_a\rangle$.

(e) $|\psi_e\rangle = \mathbf{H}|\psi_b\rangle$.

(f) $|\psi_f\rangle = \mathbf{H}|\psi_c\rangle$.

[1] that is, on any key/message pair, repeated runs of Enc will produce the same output