TD 3: Semantic security, PRFs, CPA security

Exercise 1. [*Introduction to Computational Hardness Assumptions - review*]

A group $\mathbb{G}$ is called *cyclic* if there exists an element g in $\mathbb{G}$ such that $\mathbb{G} = \langle g \rangle = \{g^n | n \text{ is an integer }\}$. Such an element g is called a *generator* of $\mathbb{G}$.

Definition 1 (Decisional Diffie-Hellman distribution)**.** *Let $\mathbb{G}$ be a cyclic group of prime order q, and let g be a public generator of $\mathbb{G}$. The* decisional Diffie-Hellman *distribution (DDH) is, $D_{\text{DDH}} = (g^a, g^b, g^{ab}) \in \mathbb{G}^3$ with a, b sampled independently and uniformly at random in $\mathbb{Z}_q$.*

Definition 2 (Decisional Diffie-Hellman assumption)**.** *The decisional Diffie-Hellman assumption states that there exists no probabilistic polynomial-time distinguisher between D_{DDH} and (g^a, g^b, g^c) with a, b, c sampled independently and uniformly at random in $\mathbb{Z}_q$.*

1. Does the DDH assumption hold in $\mathbb{G} = (\mathbb{Z}_p, +)$ for $p = \mathcal{O}(2^\lambda)$ prime?
2. Consider cyclic group $\mathbb{Z}_p$. We want to see whether DDH assumption hold in $\mathbb{G} = (\mathbb{Z}_p^\star, \times)$ for some p prime. The *square root* of $x \in \mathbb{Z}_p$ is a number $y \in \mathbb{Z}_p$ s.t. $y^2 = x \mod p$. An element $x \in \mathbb{Z}_p^*$ is called a *quadratic residue* (QR) if it has a square root in $\mathbb{Z}_p$. We introduce Legendre symbol:

$$\text{for } x \in \mathbb{Z}_p, \quad \left(\frac{x}{p}\right) := \begin{cases} 1, & \text{if } x \text{ is a QR in } \mathbb{Z}_p \\ -1, & \text{if } x \text{ is not a QR in } \mathbb{Z}_p \\ 0, & \text{if } x \equiv 0 \mod p \end{cases}$$

   (a) Let g be a generator in $\mathbb{Z}_p^*$. Prove that $g^{p-1} = 1$.
   (b) Prove that $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}}$ in $\mathbb{Z}_p^*$.
   (c) Let $x = g^r$ for some integer r. Prove that x is a QR in $\mathbb{Z}_p^*$ if and only if r is even. What can you say about the distribution of $\left(\frac{g^r}{p}\right)$ if r is uniformly sampled over $\{0, \cdots, p-1\}$?
   (d) Does the DDH assumption hold in $\mathbb{G} = (\mathbb{Z}_p^\star, \times)$ of order $p-1$?

3. Now we take $\mathbb{Z}_p$ such that $p = 2q + 1$ with q prime (also called a *safe-prime*). Let us work in a subgroup $\mathbb{G}$ of order q in $(\mathbb{Z}_p^\star, \times)$.

   (a) Given a generator g of $\mathbb{G}$, propose a construction for a function $\hat{G} : \mathbb{Z}_q \to \mathbb{G} \times \mathbb{G}$ (which may depend on public parameters) such that $\hat{G}(U(\mathbb{Z}_q))$ is computationally indistinguishable from $U(\mathbb{G} \times \mathbb{G})$ based on the DDH assumption on $\mathbb{G}$ (where, in $G(\hat{U(\mathbb{Z}_q)})$, the probability is also taken over the public parameters of $\hat{G}$).
   (b) What is the size of the output of $\hat{G}$ given the size of its input?
   (c) Why is it not a pseudo-random generator from $\{0,1\}^\ell$ to $\{0,1\}^{2\ell}$ for $\ell = \lceil \lg q \rceil$?

Exercise 2. [*Learning with errors*]

Definition 3 (Learning with Errors)**.** *Let $\ell < k \in \mathbb{N}$, $n < m \in \mathbb{N}$, $q = 2^k$, $B = 2^\ell$, $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$. The Learning with Errors (*LWE*) distribution is defined as follows: $D_{LWE,\mathbf{A}} = (\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \bmod q)$ for $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{e} \hookleftarrow U\left(\left[-\frac{B}{2}, \frac{B}{2}\right]^m \cap \mathbb{Z}^m\right)$.*

The *LWE assumption* states that, given suitable parameters k, ℓ, m, n, it is computationally hard to distinguish $D_{\mathrm{LWE},\mathbf{A}}$ from the distribution $(\mathbf{A}, U(\mathbb{Z}_q^m))$.

Let us consider the private-key encryption scheme below, which works under the following public parameters: $k, \ell, m, n, \mathbf{A}$, for which the $\mathrm{LWE}_{\mathbf{A}}$ holds.

Note. Here, "mod q"'s range is $[-\frac{q}{2}, \frac{q}{2} - 1] \cap \mathbb{Z}$ and not the usual $[0, q-1] \cap \mathbb{Z}$ to ease the description of the scheme.

Keygen(1^λ)**:** from 1^λ, this algorithm outputs a random vector $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$ as a secret key.

Enc$_{\mathbf{s}}(\mathfrak{m})$**:** from the secret key $\mathbf{s}$ and a message $\mathfrak{m} \in \{0,1\}^m$, the algorithm Enc samples a random vector $\mathbf{e} \hookleftarrow U\left(\left[-\frac{B}{2}, \frac{B}{2}\right]^m \cap \mathbb{Z}^m\right)$ and outputs $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + \frac{q}{2}\mathfrak{m} \bmod q$ as a ciphertext.

Dec$_{\mathbf{s}}(\mathbf{c})$**:** from the secret key $\mathbf{s}$ and a ciphertext $\mathbf{c}$, the decryption algorithm computes $\mathbf{v} = \mathbf{c} - \mathbf{A} \cdot \mathbf{s}$. Then Dec constructs the message $\mathfrak{m}'$ from $\mathbf{v}$: for each component of $\mathbf{v}$, sets the corresponding component of $\mathfrak{m}'$ as follows: 0 if $\frac{-q}{4} \le v_i \le \frac{q}{4}$, and 1 otherwise.

1. Prove the correctness of this cipher.
2. Show that this cipher is computationally secure.

If you take a look at this cipher, you can view it as a one-time pad on $\frac{q}{2}\mathfrak{m}$, which means that the message is hidden in the most significant bit of $\mathbf{e} + \frac{q}{2}\mathfrak{m}$. Now, if one wants to hide the message in the least significant bit of the OTP, one solution is to encrypt a message as: $\mathbf{c} = 2 \cdot (\mathbf{A} \cdot \mathbf{s} + \mathbf{e}) + \mathfrak{m} \bmod q$.

3. Construct a "decryption" algorithm that does not use the secret key to compute $\mathfrak{m}$.
4. Why is it also a bad idea to encrypt as $\mathbf{c} = \mathbf{A} \cdot \mathbf{s} + 2\mathbf{e} + \mathfrak{m}$?

Exercise 3. [*A weak-PRP is PRF*]

Definition 4. *Weak PRP. A function $F : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$ is said to be a Pseudo-Random Permutation (PRP) if*

- *For any $k \in \{0,1\}^s$, the function $F_k : x \mapsto F(k, x)$ is a permutation (i.e., a bijection from $\{0,1\}^n$ to $\{0,1\}^n$).*
- *All PPT algorithms $\mathcal{A}$ have a negligible advantage in the following game*

$\mathcal{C}$	$\mathcal{A}$
$b \hookleftarrow U(\{0,1\})$	
$k \hookleftarrow U(\{0,1\}^s)$	
if $b = 0$, then $F = F(k, \cdot)$	
else F is a uniformly chosen permutation of $\{0,1\}^n$	
	sends x to $\mathcal{C}$ (polynomially many queries)
sends $F(x)$ to $\mathcal{A}$	
	outputs a bit $b' \in \{0,1\}$

where $Adv_A^{\text{weak-PRP}}(F) = |\Pr\{b' = 1 | b = 1\} - \Pr\{b' = 1 | b = 0\}|$.

Remark. A PRP is very similar to a PRF, except that it is a bijection, and it should be indistinguishable from a uniform bijection (while a PRF should be indistinguishable from a uniform function).

The objective of this exercise is to show that a PRP is also a PRF. We will first show that a PPT algorithm cannot distinguish between a random function and a random permutation with non negligible advantage. Let $\mathcal{A}$ be a PPT algorithm with running time at most t. We want to show that $\mathcal{A}$ has negligible advantage in the following game.

$\mathcal{C}$	$\mathcal{A}$
$b \hookleftarrow U(\{0,1\})$	
if $b=0$, then F is a random permutation of $\{0,1\}^n$	
else F is a random function from $\{0,1\}^n$ to $\{0,1\}^n$	
	sends x to $\mathcal{C}$ (polynomially many queries)
sends $F(x)$ to $\mathcal{A}$	
	outputs a bit $b' \in \{0,1\}$

1. Give a pseudo-code algorithm for implementing $\mathcal{C}$ in the case where F is a random function and in the case where F is a random permutation.

2. Show that the advantage of $\mathcal{A}$ in distinguishing whether F is a random permutation or a random function is at most the probability that $\mathcal{A}$ finds a collision when F is a random function. In other words, show that

$$|\Pr\{\mathcal{A} \text{ outputs } 1 \,|F \text{ is a random function }\} - \Pr\{\mathcal{A} \text{ outputs } 1 \,|F \text{ is a random permutation }\}| \leq \delta$$

   where δ is the probability to find a collision when sampling t independent uniform elements in $\{0,1\}^n$ (that is, $\delta = \Pr_{y_1,\cdots,y_t \leftarrow \mathcal{U}(\{0,1\}^n)}\{\exists\, i \neq j \text{ s.t. } y_i = y_j\}$).

3. Show that $\delta \leq \frac{t^2}{2^n}$

4. Show that if $n \geq \lambda$ (the security parameter), then any pseudo-random permutation is also a pseudo-random function.

Exercise 4. [*Increasing the advantage of an attacker - review*]
Let G be a pseudo-random generator from $\{0,1\}^s$ to $\{0,1\}^n$ for some integers s and n. Let $i \in \{1,\dots,n\}$ and let $\mathcal{A}$ be a PPT algorithm such that, for all $k \in \{0,1\}^s$, we have:

$$Pr[\mathcal{A}(G(k)_{1\cdot\cdot i-1}) = G(k)_i] \geq \frac{1}{2} + \epsilon$$

where the probability runs over the randomness of $\mathcal{A}$. Note that unlike the definition of the advantage seen in class, here we consider only the probability over the randomness of $\mathcal{A}$ and not over the random choice of k (we will see why later). Our objective is to construct a new attacker $\mathcal{A}'$ with an advantage arbitrarily close to 1 (for instance $Pr[\mathcal{A}(G(k)_{1\cdot\cdot i-1}) = G(k)_i] \geq 0.999$ for all $k \in \{0,1\}^s$).

1. Propose a method to improve the success probability of $\mathcal{A}$

Let m be some integer to be determined. Let $\mathcal{A}'$ be an algorithm that evaluates $\mathcal{A}$ on $G(k)_{1\cdot\cdot i-1}$ $2m+1$ times, to obtain $2m+1$ bits $b_1,\dots,b_{2m+1}$ and then outputs the bit that appeared the most (i.e. at least $m+1$ times).

2. Give a lower bound on $Pr[\mathcal{A}'(G(k)_{1\cdot\cdot i-1}) = G(k)_i]$, for all $k \in \{0,1\}^s$. It may be useful to recall Hoeffding's inequality for Bernoulli variables: let $X_1,\dots,X_{2m+1}$ be independent Bernoulli random variables, with $Pr[X_i = 1] = 1 - Pr[X_i = 0] = p$ for all i, and let $S = X_1 + \cdots + X_{2m+1}$. Then, for all $x > 0$, we have
$$Pr[|S - \mathbb{E}(S)| \geq x\sqrt{2m+1}] \leq 2e^{-2x^2}$$

3. What should be the value of m (depending on ϵ) if we want that $Pr[\mathcal{A}'(G(k)_{1\cdot\cdot i-1}) = G(k)_i] \geq 0.999$ for all k? It may be useful to know that $e^{-8} \leq 0.0005$.

4. Do we have $PRED\mathrm{Adv}_{(\mathcal{A}')} \geq 0.999$ if $Pr[\mathcal{A}'(G(k)_{1\cdot\cdot i-1}) = G(k)_i] \geq 0.999$ for all k?

5. What condition on ϵ do we need to ensure that $\mathcal{A}'$ runs in polynomial time?

Let now $\mathcal{A}$ be an attacker such that

$$\mathrm{Adv}(\mathcal{A}) = Pr_{k \leftarrow \mathcal{U}(\{0,1\}^s)}[\mathcal{A}(G(k)_{1..i-1}) = G(k)_i] \geq \frac{1}{2} + \epsilon$$

Note that we are now looking at the definition of advantage given in class, where the probability also depends on the uniform choice of k. We want to show that in this case, we cannot always amplify the success probability of the attacker by repeating the computation.

In the following, we write $Pr[\mathcal{A}(G(k)_{1..i-1}) = G(k)_i]$ when we only consider the probability over the internal randomness of $\mathcal{A}$ (and k is fixed) and $Pr_{k \leftarrow \mathcal{U}(\{0,1\}^s)}[\mathcal{A}(G(k)_{1..i-1}) = G(k)_i]$ when we consider the probability over the choice of k and the internal randomness of $\mathcal{A}$.

Suppose that $s \geq 2$ and define

$$G(k) = \begin{cases} 00\cdots 0, & \text{if } k_0 = k_1 = 0 \\ G_0(k), & \text{otherwise,} \end{cases}$$

where G_0 is a secure PRG from $\{0,1\}^s$ to $\{0,1\}^n$.

6. Show that there exists a PPT attacker $\mathcal{A}$ with non negligible advantage (for the unpredictability definition) against G.

7. Show on the contrary that there is no PPT attacker $\mathcal{A}$ with $\mathrm{Adv}(\mathcal{A}) \geq \frac{7}{8}$ (assuming that G_0 is a secure PRG).