Exercice 1.  

**Building a PRF from a PRG**

Let \( n \in \mathbb{N} \) be a security parameter. Let \( G : \{0,1\}^n \to \{0,1\}^{2^m} \) denote a length-doubling Pseudo-Random Generator (PRG). We define \( G_0 : \{0,1\}^n \to \{0,1\}^n \) and \( G_1 : \{0,1\}^n \to \{0,1\}^n \) as the functions that evaluate \( G \) and keep the \( n \) left-most bits and \( n \) right-most bits, respectively.

We consider the following keyed function

\[
F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n
\]

\[
k, x \mapsto G_x (G_{x-1} (\ldots (G_{x_1}(k)) \ldots )) ,
\]

where \( x = x_1 \ldots x_{n-1} x_n \). Our aim is to show that \( F \) is a Pseudo-Random Function (PRF).

1. Recall the security definition of a PRF and the advantage of a PRF adversary.

We now consider \( n+1 \) functions defined as follows, for \( i \in \{0, \ldots , n-1\} \):

\[
F_i : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n
\]

\[
k, x \mapsto G_x (G_{x-1} (\ldots (G_{x_{i+1}}(u_{x,x_{i-1} \ldots x_1})) \ldots )) ,
\]

where each \( u_{x,x_{i-1} \ldots x_1} \) is chosen uniformly and independently in \( \{0,1\}^n \), and fixed once and for all (it is hardwired in the definition of \( F_i \)). For \( i = 0 \), we define \( u_k = k \). For \( i = n-1 \), we let \( F^n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \) be a uniformly sampled function.

2. Show that if there is a PRF adversary \( \mathcal{A} \) against \( F \), then \( \mathcal{A} \) distinguishes between an oracle access to \( F_i \) and an oracle access to \( F_{i+1} \), for some \( i \in \{0, \ldots , n-1\} \).

For \( t \geq 1 \), we consider the function

\[
G^i : (\{0,1\}^n)^t \to (\{0,1\}^{2^n})^t
\]

\[
(k_1, \ldots , k_t) \mapsto (G(k_1), \ldots , G(k_t)).
\]

3. Show that any PRG adversary \( \mathcal{B}^t \) against \( G^i \) leads to a PRG adversary against \( G \).

Let \( i \) and \( \mathcal{A} \) be as above. Let \( t \) denote the run-time of \( \mathcal{A} \). We are going to show that \( \mathcal{A} \) may be used to mount an attack against \( G^i \). We consider the following algorithm \( \mathcal{B}^t \).

- It takes as input a string \( (y_1,0, y_1,1, y_2,0, y_2,1, \ldots , y_t,0, y_t,1) \in (\{0,1\}^{2^n})^t \).
- It maintains a list \( L \) of triples that is initially empty.
- It interacts with Algorithm \( \mathcal{A} \).
- Each time \( \mathcal{A} \) makes a function query \( x_1 \ldots x_n \), it checks whether \( x_1 \ldots x_i = x'_1 \ldots x'_i \) for a previously queried input \( x'_1 \ldots x'_n \).
* If this is not the case, then it computes the length \( j \) of \( L \), and it adds \((x_1 \ldots x_i, y_{j+1}, 0, y_{j+1})\) to the list \( L \).
* Else, it finds the triple \((x_1 \ldots x_i, y_{j+1}, 0, y_{j+1})\) in \( L \).
* In both cases, it replies \( G_{x_n} \ldots G_{x_{i+2}}(y_{j+1}) \ldots \) if \( x_{i+1} = 0 \) and \( G_{x_n} \ldots G_{x_{i+2}}(y_{j+1}) \ldots \) if \( x_{i+1} = 1 \). If \( i = n - 1 \), it replies \( y_{j+1} \) if \( x_n = 0 \) and \( y_{j+1} \) if \( x_n = 1 \).
* Eventually, Algorithm \( \mathcal{A} \) outputs a bit \( b \in \{0, 1\} \), which \( B^\ell \) forwards as its own output.

4. Show that if the \( y_{j,0} \)'s are uniformly and independently random, then the view of \( \mathcal{A} \) is exactly the same as if it were given oracle access to \( F^{n+1} \).

5. Show that if the \( y_{j,0} \)'s are uniformly and independently random \( k_j \)'s, then the view of \( \mathcal{A} \) is exactly the same as if it were given oracle access to \( F^t \).

6. Conclude. In particular, give bounds on the run-time and advantage of the adversary against PRG \( G \) as functions of the run-time and advantage of the adversary against PRF \( F \).

**Exercise 2. Pseudo-random synthesizers**

Let \( n \in \mathbb{N} \) be a security parameter. Let \( \mathbb{G} \) be a cyclic group of prime order \( q > 2^n \) with a generator \( g \in \mathbb{G} \). Recall that the Decisional Diffie-Hellman (DDH) assumption says that the following distributions

\[
D_0 := \{(g^a, g^b, g^{ab}) \mid a, b \leftarrow U(\mathbb{Z}_q)\}, \quad D_1 := \{(g^a, g^b, g^c) \mid a, b, c \leftarrow U(\mathbb{Z}_q)\}
\]

are computationally indistinguishable.

A **synthesizer** \( G : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{G}^{n \times n} \) is a length-squaring function which takes as input a random seed made of \( 2n \) scalars \( \vec{a} = (a_1, \ldots, a_n) \leftarrow U(\mathbb{Z}_q^n), \vec{b} = (b_1, \ldots, b_n) \leftarrow U(\mathbb{Z}_q^n) \) and outputs a \( n \times n \) matrix

\[
G((a_1, \ldots, a_n), (b_1, \ldots, b_n)) = \left( g^{a_i b_j} \right)_{i,j \in \{1, \ldots, n\}} = \begin{bmatrix} g^{a_1 b_1} & \ldots & g^{a_1 b_n} \\ g^{a_2 b_1} & \ldots & g^{a_2 b_n} \\ \vdots & \ddots & \vdots \\ g^{a_n b_1} & \ldots & g^{a_n b_n} \end{bmatrix}
\]

1. Show that an unbounded adversary (which can compute discrete logarithms in \( \mathbb{G} \)) can distinguish an output of \( G \) from a truly random matrix in \( \mathbb{G}^{n \times n} \).

2. Show that \( G : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{G}^{n \times n} \) is a pseudo-random generator under the DDH assumption in the group \( \mathbb{G} \).

**Hint** (but you may choose not to read it): Consider a sequence of \( n^2 \) hybrid experiments \( \text{Exp}_{k, \ell} \) for \( k, \ell \in \{1, \ldots, n\} \), where the output of \( G((a_1, \ldots, a_n), (b_1, \ldots, b_n)) \) is replaced by a matrix of the form

\[
G^{(k, \ell)}((a_1, \ldots, a_n), (b_1, \ldots, b_n)) = \left( g^{u_{ij}} \right)_{i,j \in \{1, \ldots, n\}}
\]

where \( u_{ij} = a_i b_j \) if \( i > k \) or \( (i = k) \land (j > \ell) \) and \( u_{ij} \leftarrow U(\mathbb{Z}_q) \) otherwise. Define \( G^{(0,0)} \) to be actual function of \( 1 \).
Exercice 3. Pseudo-random functions from the DDH assumption

Let \( n \in \mathbb{N} \) be a security parameter. Let \( \mathbb{G} \) be a cyclic group of prime order \( q > 2^n \) which is generated by \( g \in \mathbb{G} \) and for which DDH is presumably hard.

For a public \( g \in \mathbb{G} \), we define the function \( F_K : \{0, 1\}^n \rightarrow \mathbb{G} \) which is keyed by a random vector \( K = (a_0, a_1, \ldots, a_n) \in U(\mathbb{Z}_q^{n+1}) \) and takes as input a bitstring \( x = x_1 \ldots x_n \in \{0, 1\}^n \) to output

\[
F_K(x) = g^{a_0} \prod_{j=1}^n a_j^{x_j}.
\]

Our goal is to prove that the function \( F_K : \{0, 1\}^n \rightarrow \mathbb{G} \) is a pseudo-random function under the DDH assumption in \( \mathbb{G} \).

For an index \( i \in \{1, \ldots, n\} \), we consider an experiment where the adversary is given oracle access to a hybrid function \( F_K^{(i)} : \{0, 1\}^n \rightarrow \mathbb{G} \) defined as

\[
F_K^{(i)}(x) = g^R(x[1 \ldots i]) \prod_{j=i+1}^n a_j^{x_j},
\]

where \( R : \{0, 1\}^i \rightarrow \mathbb{Z}_q \) is a truly random function and \( x[1 \ldots i] = x_1 \ldots x_i \in \{0, 1\}^i \) denotes the \( i \)-th prefix of the input \( x \in \{0, 1\}^n \).

1. Prove that \( F_K^{(0)}(x) \) coincides with the function \( F_K(\cdot) \) of (2) if we define the length-0 prefix of \( x \in \{0, 1\}^n \) to be the empty string \( \epsilon \) and \( R(\epsilon) \) to be a non-zero constant. How does the function \( F_K^{(n)}(x) \) behave in the adversary’s view?

2. Let \((g^a, g^b, g^c)\) be a DDH instance, where \( a, b \leftarrow U(\mathbb{Z}_q) \), and we have to decide if \( c = ab \) or if \( c \leftarrow U(\mathbb{Z}_q) \). Describe a probabilistic polynomial-time algorithm that creates \( Q \) randomized DDH instances

\[
\{ (g^a, g^b, g^c_k) \}_{k=1}^Q,
\]

where \( \{b_k\}_{k=1}^Q \) are random and independent over \( \mathbb{Z}_q \), with the properties that

- If \( c = ab \) then \( c_k = ab_k \) for each \( k \in \{1, \ldots, Q\} \).
- If \( c \leftarrow U(\mathbb{Z}_q) \), then \( \{c_k\}_{k=1}^Q \) are independent and uniformly distributed over \( \mathbb{Z}_q \).

3. For each \( i \in \{0, \ldots, n\} \), we define the experiment \( \text{Exp}_i \) where the adversary \( \mathcal{A} \) is given oracle access to the function \( F_K^{(i)}(x) \) and eventually outputs a bit \( b' \in \{0, 1\} \) after \( Q \) evaluation queries. Prove that, for each \( i \in \{0, \ldots, n-1\} \), experiment \( \text{Exp}_i \) is computationally indistinguishable from \( \text{Exp}_{i+1} \) under the DDH assumption in \( \mathbb{G} \). Namely, prove that \( \mathcal{A} \) outputs \( b' = 1 \) with about the same probabilities in \( \text{Exp}_i \) and \( \text{Exp}_{i+1} \) unless the DDH assumption is false.

4. Give an upper bound on the advantage of a PRF distinguisher as a function of the maximal advantage of a DDH distinguisher.