**Exercise 1.** *Let $\mathbb{G}$ be an abelian group with an element, $g \in \mathbb{G}$, of order $n$. A Schnorr signature of a message, $m$, by a private key, $x$, is a pair $(s, R)$ such that $s = k + H(R, m) \cdot x \in \mathbb{Z}/n\mathbb{Z}$ and $R = g^k \in \mathbb{G}$ is the nonce ($k$ is a random element of $\mathbb{Z}/n\mathbb{Z}$ generated for this signature). Intuitively, this works because $s$ commits to the message using a hash, uses the private key, but it does not reveal the private key because of the random value $k$. A Schnorr signature can be verified by checking the equation $g^s == R \cdot X^{H(R,m)}$, where $X = g^x$ is the signer's public key.*

*Use proof by reduction to show that if we assume that the discrete log problem is hard in $\mathbb{G}$, then we can conclude that it is hard to forge two Schnorr signatures for the same nonce and key, that is, if we are given a public key $X = g^x$ and nonce $R$ (without being given $x$ or $k$) it is hard to compute two different messages $m_1$ and $m_2$ and values $s_1$ and $s_2$ such that $(s_1, R)$ and $(s_2, R)$ are valid signatures for $X$ of $m_1$ and $m_2$, respectively.*

*This will later be an important part of the proof of the stronger statement that if the discrete log problem is hard, then Schnorr signatures are unforgable in a much stronger sense.*

PROOF:-

Let $G_1$ be a cyclic group of prime order $n$, and let $g \in G_1$ be a fixed generator. Let $X = g^x$ for some secret $x \in \mathbb{Z}_n$, which serves as a public key in the Schnorr signature scheme.

Assume, we are given:

- a group element $R = g^k \in G_1$ for some random unknown $k \in \mathbb{Z}_n$,
- two distinct messages $m_1, m_2 \in \{0,1\}^*$ with $m_1 \neq m_2$
- and two integers $s_1, s_2 \in \mathbb{Z}_n$

such that the pairs $(s_1, R)$ and $(s_2, R)$ constitute a valid Schnorr signature on $m_1$ and $m_2$ respectively, under public key $X$.

Since these are valid signatures, we have:

$$g^{s_1} = R \cdot X^{H(R, m_1)} \quad \text{---------- (1)}$$

$$\Rightarrow g^{s_1} = g^k \cdot g^{x H(R, m_1)} = g^{k + x h_1} \quad \text{---------- (2)}$$

where $h_1 = H(R, m_1)$

and

$$g^{s_2} = R \cdot X^{H(R, m_2)} \quad \text{---------- (3)}$$

$$\Rightarrow g^{s_2} = g^k \cdot g^{x H(R, m_2)} = g^{k + x h_2} \quad \text{---------- (4)}$$

where $h_2 = H(R, m_2)$

Since exponentiation in $G$ is injective (as $G$ is a cyclic group of prime order), (2) and (4) becomes,

$$s_1 \equiv k + x h_1 \bmod n \quad \text{---------- (5)}$$

$$\& \quad s_2 \equiv k + x h_2 \bmod n \quad \text{---------- (6)}$$

Substracting (6) from (5), we get

$$s_1 - s_2 \equiv x(h_1 - h_2) \pmod n \quad \text{---------- (7)}$$

Since $m_1 \neq m_2$ and $H$ is assumed to be collision resistant, the probability that $H(R, m_1) = H(R, m_2)$ is negligible.

$\therefore$ we have $h_1 \neq h_2$

Since $n$ is prime and $h_1 \neq h_2$, $h_1 - h_2 \in \mathbb{Z}_n^{\times}$ is invertible in the finite field $\mathbb{Z}_n$.

$\therefore$ to solve $x$, we divide both sides of the congruence by $h_1 - h_2$:

$$\Rightarrow \quad x \equiv (s_1 - s_2)(h_1 - h_2)^{-1} \pmod{n}$$

Thus private key $x$ is uniquely determined modulo $n$ and can be computed from publicly known values $s_1, s_2, h_1, h_2$

This shows that, given two Schnorr signatures on distinct messages that uses the same nonce $R$, one can compute the private key $x$, thereby solving the discrete logarithm problem for $X = g^x$.

This contradicts the assumption that DLP is hard. Hence if DLP is hard in $G_1$, then forging two Schnorr signatures for the same key and nonce is hard.

HENCE THE PROOF //

**Exercise 2.** *Use proof by reduction and the precise definitions above to show that*

(a) *If the DDH assumption holds, then the CDH assumption holds.*

(b) *If The CDH assumption holds, then the DL assumption holds.*

*In order to solve this part, you will need two facts that we will prove later:*

(1) *If you have two independent events $A$ and $B$, then $Pr[A \text{ and } B] = Pr[A] \cdot Pr[B]$.*

(2) *If $F(\lambda)$ is not negligible, then $F(\lambda)^2$ is also not negligible.*

Let $\mathcal{G}(\lambda)$ be a PPT (probabilistic polynomial time) algorithm that, on input the security parameter $\lambda \in \mathbb{N}$, it outputs a cyclic group $G$ of prime order $p > 2^\lambda$ with generator $g \in G$. $\mathcal{G}(\lambda) \to (G, p, g)$

For any PPT algorithm $\mathcal{A}$, we denote

$$Adv_{\mathcal{A}}^{dl}(\lambda) := Pr\left[\mathcal{A}(g, g^x) = x\right],$$

$$Adv_{\mathcal{A}}^{cdh}(\lambda) := Pr\left[\mathcal{A}(g, g^x, g^y) = g^{xy}\right],$$

$$Adv_{\mathcal{A}}^{ddh}(\lambda) := \left| Pr\left[\mathcal{A}(g, g^x, g^y, g^{xy}) = 1\right] - 1/2 \right|$$

A real valued function of $\lambda$, $f(\lambda)$ is negligible if $\forall c > 0 \ \exists \lambda_0 \in \mathbb{N}$ such that $\forall \lambda > \lambda_0$, we have

$$f(\lambda) < 1/\lambda^c$$

*(a) If the DDH assumption holds, then the CDH assumption holds.*

CLAIM: If DDH advantage of every PPT algorithm is negligible, then the CDH advantage of every PPT algorithm is also negligible.

We prove the contrapositive.

Suppose $\exists$ a PPT algorithm $\mathcal{C}$ and a non-negligible function $\varepsilon(\lambda)$ such that $Adv_{\mathcal{C}}^{cdh}(\lambda) = \varepsilon(\lambda)$ and $\exists c > 0$ and infinitely many values of $\lambda \in \mathbb{N}$ such that $\varepsilon(\lambda) \geq 1/\lambda^c$.

We build a PPT distinguisher $\mathcal{D}$ for the DDH game as follows:

on input $(g, X, Y, Z)$ with $X = g^x$, $Y = g^y$, it runs $\hat{Z} \leftarrow \mathcal{C}(g, X, Y)$ once and outputs $1 \iff \hat{Z} = Z$.

Let $b \in \{0,1\}$ be the hidden bit selected by the DDH challenger.

CASE 1: If $b = 1$ then $Z = g^{xy}$, so $\mathcal{D}$ outputs 1 with probability $\varepsilon(\lambda)$

$$Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1] = Pr[\mathcal{C} \text{ returns } g^{xy}] = Adv_{\mathcal{C}}^{cdh}(\lambda) \geq \varepsilon(\lambda)$$

CASE 2: If $b=0$, then $Z=g^z$ for a random $z$ that is independent of $(X,Y)$. Hence $\hat{Z}=g^{xy}$ and $Z=g^{xy}$ are independent.

By independence,

$$Pr[\mathcal{D} \text{ outputs } 1 \mid b=0] = Pr[\hat{Z}=g^{xy}]\, Pr[Z=g^{xy}]$$

$$= Adv_{\mathcal{A}}^{cdh}(\lambda) \cdot 1/p \leq \frac{\varepsilon(\lambda)}{p}$$

Consequently, the DDH advantage of $\mathcal{D}$ is

$$Adv_{\mathcal{D}}^{ddh}(\lambda) = \left| Pr[1 \mid b=1] - Pr[1 \mid b=0] \right| = \varepsilon(\lambda)(1-1/p)$$

Because $p > 2\lambda$, the factor $(1-1/p) \geq 1/2$

Hence

$$Adv_{\mathcal{D}}^{ddh}(\lambda) \geq \frac{\varepsilon(\lambda)}{2}$$

Since $\varepsilon(\lambda)$ is non-negligible, so is $\varepsilon(\lambda)/2$.

Thus a PPT adversary with non-negligible DDH advantage exists, contradicting DDH hardness.

$\therefore$ DDH hardness $\Longrightarrow$ CDH hardness

(b) *If The CDH assumption holds, then the DL assumption holds.*

CLAIM: If CDH advantage of every PPT algorithm is negligible, the DL advantage of every PPT algorithm is also negligible.

We prove the contrapositive.

Assume a PPT algorithm $\mathcal{L}$ and a non-negligible function $\delta(\lambda)$ satisfy $Adv^{dl}_{\mathcal{L}}(\lambda) = \delta(\lambda)$ for some non-negligible $\delta(\lambda)$.

From $\mathcal{L}$, we construct a CDH solver $\mathcal{C}^{\mathcal{L}}$ as follows:

on input $(g, X, Y) = (g, g^x, g^y)$ it computes $\hat{x} := \mathcal{L}(g, X)$.

if $\hat{x} \neq x$, it outputs a "fail", else it returns

$$Y^{\hat{x}} = g^{y\hat{x}} = g^{xy}$$

So the solver outputs the correct Diffie-Hellman,

Hence $Adv^{cdh}_{\mathcal{C}^{\mathcal{L}}}(\lambda) = \delta(\lambda)$ for some non-negligible $\delta(\lambda)$

which contradicting the assumed hardness of CHD.

$\therefore$ CDH hardness $\Rightarrow$ DL hardness.

**Exercise 3.** *Suppose there is a room with 30 people, and assume that the probabilities of all assignments of one of the 365 possible birthdays to each of the 30 people are equal (this is of course not true, but assuming a uniform distribution will make things simpler). Compute the probability that no two people share a birthday, and then compute the probability that at least two people share a birthday.*

*(If instead of people and birthdays we think of inputs and hash values, this "birthday problem" shows that finding hash collisions is easier than finding hash pre-images).*

Let $n = 30$ be the number of people in a room.
Let the number of possible birthdays be $d = 365$.
Assuming all birthday assingments are uniform and independent.

i.e., each person's birthday is choosen uniformly from $\{1, 2, \ldots 365\}$ and each assigment is equally likely.

We have to compute:

i) $P_{\text{no collision}} = Pr[\text{no two people share a birthday}]$

ii) $P_{\text{collision}} = Pr[\text{at least two people share a birthday}]$.

Let $\Omega$ be the sample space of all birthday assigments for $n = 30$ people.
Since each person independently chooses 365 options, the total number of assignments is $|\Omega| = 365^{30}$

We now count the number of injective fnctions from a set of size 30 into a set of size 365.

i.e., birthday assignments where no two people share a birthday.

Let us denote the number of such assigments by $N$.

We assume that the first person can have any of the 365 birthdays, the second one can have any different one (364 choices) and so on.

So the number of valid (distinct) birthday asigmments is

$$N = 365 \cdot 364 \cdot \ldots (365-30-1)$$

$$= \prod_{i=0}^{30} (365-i-1) \quad (\text{or}) \quad \prod_{i=0}^{29} (365-i)$$

i) Pr[no two people share a birthday] :-

$$P_{\text{no collision}} = \frac{N}{|\Omega|} = \frac{\prod_{i=0}^{29} (365-i)}{365^{30}}$$

$$= \frac{365-0}{365} \cdot \frac{365-1}{365} \cdots \frac{365-29}{365}$$

$$= \prod_{i=0}^{29} \left(1 - \frac{i}{365}\right)$$

$\approx 0.293683$

∴ The chance that all 30 people have different birthdays is about 29.4 %

ii) Pr[at least two people share a birthday] :-

$$P_{collision} = 1 - P_{no\ collision} \approx 1 - 0.293683$$

$$\approx 0.706316$$

∴ There is about 70.6 % chance for at least two out of 30 people share a birthday.

**Exercise 4.** *Explain why the following is true using the definition of probability:*

$$Pr\left[\bigcup_{i=1}^{n} E_i\right] \leq \sum_{i=1}^{n} Pr[E_i] .$$

*When are these two values equal? (Hint: first try the case where $n = 2$).*

PROOF USING INCLUSION - EXCLUSION:-

Case $n=2$ :

Let $E_1$ and $E_2$ be two events.

By inclusion-exclusion:

$$Pr[E_1 \cup E_2] = Pr[E_1] + Pr[E_2] - Pr[E_1 \cap E_2]$$

Since $Pr[E_1 \cap E_2] \geq 0$, it follows:

$$Pr[E_1 \cup E_2] \leq Pr[E_1] + Pr[E_2]$$

Thus the union bound holds for $n=2$.

Case $n$ (general):

By inclusion-exclusion principle, for any finite collection $E_1, \ldots E_n$,

$$Pr\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{k=1}^{n} (-1)^{k+1} \sum_{1 \leq i_1 < \ldots < i_k \leq n} Pr[E_{i_1} \cap \ldots \cap E_{i_k}]$$

i.e., $Pr(\bigcup E_i) = \sum_i Pr[E_i] - \sum_{i<j} Pr[E_i \cap E_j]$

$+ \sum_{i<j<k} Pr[E_i \cap E_j \cap E_k]$

$- \ldots\ldots$

$+ (-1)^{n+1} Pr[E_1 \cap \ldots \cap E_n]$

Since each of these intersection terms is non-negative, we have

$$Pr\left(\bigcup_{i=1}^{n} E_i\right) \leq \sum_{i=1}^{n} Pr[E_i]$$

Hence the result //

PROOF BY INDUCTION :-

BASE CASE :- $n = 1$

$$Pr(E_1) \leq Pr(E_1) \quad \text{---------- (1)}$$

This is trivially true.

INDUCTION HYPOTHESIS:-

Assume the result holds for some $n=k$.

i.e., $$Pr\left(\bigcup_{i=1}^{k} E_i\right) \leq \sum_{i=1}^{k} Pr[E_i] \quad \text{---------- (2)}$$

INDUCTIVE PROOF:-

We now want to prove for $n=k+1$

Let $A = \bigcup_{i=1}^{k} E_i$ and consider,

$$Pr\left(\bigcup_{i=1}^{k+1} E_i\right) = Pr(A \cup E_{k+1})$$

$$\Rightarrow Pr(A \cup E_{k+1}) = Pr(A) + Pr(E_{k+1}) - Pr[A \cap E_{k+1}]$$

Since $Pr[A \cap E_{k+1}] \geq 0$, we get,

$$Pr(A \cup E_{k+1}) \leq Pr(A) + Pr(E_{k+1})$$

Applying inductive hypothesis to $Pr(A)$,

$$Pr(A) = Pr\left(\bigcup_{i=1}^{k} E_i\right) \leq \sum_{i=1}^{k} Pr[E_i]$$

So,

$$Pr\left[\bigcup_{i=1}^{k+1} E_i\right] \leq \sum_{i=1}^{k} Pr[E_i] + Pr[E_{i+1}] = \sum_{i=1}^{k+1} Pr[E_i]$$

$\therefore$ By the principle of mathematical induction,

$$P_H\left(\bigcup_{i=1}^{n} E_i\right) \leq \sum_{i=1}^{n} P_H[E_i] \quad \forall\, n \in \mathbb{N}$$

**Exercise 5.** *Using the definitions above, show that two events, $E_1$ and $E_2$, are independent if and only if $P(E_1 \cap E_2) = P(E_1) \cdot P(E_2)$.*

$\Rightarrow$ NECESSITY

Suppose $E_1$ and $E_2$ are independent,

By the definition of independence, we have

$$Pr[E_1 | E_2] = Pr[E_1] \quad \text{---------- (1)}$$

By the definition of conditional probability (given $Pr[E_2] > 0$), we know,

$$Pr[E_1 | E_2] = \frac{Pr[E_1 \cap E_2]}{Pr[E_2]} \quad \text{---------- (2)}$$

Equating (1) & (2), we get,

$$\frac{Pr[E_1 \cap E_2]}{Pr[E_2]} = Pr[E_1]$$

$$\Rightarrow Pr[E_1 \cap E_2] = Pr[E_1] \cdot Pr[E_2]$$

Hence the identity holds.

$\Leftarrow$ SUFFICIENCY

Conversely, suppose $Pr[E_1 \cap E_2] = Pr[E_1] \cdot Pr[E_2]$ holds.

Again assuming $Pr[E_2] > 0$, we have

$$Pr[E_1 | E_2] = \frac{Pr[E_1 \cap E_2]}{Pr[E_2]} = \frac{Pr[E_1] \cdot \cancel{Pr[E_2]}}{\cancel{Pr[E_2]}} = Pr[E_1]$$

Thus $Pr[E_1 / E_2] = Pr[E_1]$,

so by definition,

$E_1$ and $E_2$ are independent

$\therefore$ We conclude that

$E_1$ and $E_2$ are independent

$\iff Pr[E_1 \cap E_2] = Pr[E_1] \cdot Pr[E_2]$

**Exercise 6.** *We now have all of the tools we need to prove that the one-time pad is secure!*

(a) *Let us begin with the example of the single-bit encryption case (where $\lambda = 1$). Show that for all adversary programs, $\mathcal{A}$, it is always true that* $\mathsf{Adv}_{\mathcal{A}}^{\text{distinguishcipher}}(\lambda) = 0$.

CLAIM:

For any adversary $\mathcal{A}$, and for $\lambda = 1$, we have

$$Adv_{\mathcal{A}}^{distinguishcipher}(1) = 0$$

i.e., $P_H[\mathcal{A}(c) = b] = 1/2$

where $c = E(k, m_b) = k \oplus m_b$ with $k \in \{0,1\}$ choosen uniformly at random and $b \in \{1,2\}$ choosen uniformly at random.

PROOF:

Let us consider the one-time pad encryption scheme where messages, keys and ciphertexts are all elements of $\{0,1\}$, since $\lambda = 1$.

The encryptim fnctim is defined as $E(k,m) = k \oplus m$

Let $\mathcal{A}$ be an arbitrary adversary where he submits two messages $m_1, m_2 \in \{0,1\}$. The challenger selects a random key $k \in \{0,1\}$ and a random bit $b \in \{1,2\}$ each with uniform probability $1/2$. Then the challenger returns $c = k \oplus m$ to $\mathcal{A}$ and adversary outputs a guess $\hat{b}$. The adversary wins if $\hat{b} = b$.

Our goal is to compute $P_R[A(c)=b]$ and show it is exactly $1/2$, implying that the adversary has no advantage over random guessing.

Let the sample space be $\Omega = K \times \{1,2\} \times R$ where $K=\{0,1\}$ is the key space and $R$ is the randomness used internally by adversary $A$.

There are 4 possible combinations of key and bit, $(k,b) \in \{0,1\} \times \{1,2\}$ each with probability $1/4$.

CASE 1: $(k=0, b=1)$

Then $c = 0 \oplus m_1 = m_1$

$\Rightarrow$ The adversary receives $c = m_1$ and the event $A(c)=b$ becomes $A(m_1)=1$.

$$P_{m_1}^{(1)} = P_R[A(m_1)=1]$$

CASE 2: $(k=0, b=2)$

Then $c = 0 \oplus m_2 = m_2$ and $P_R[A(c)=b] = P_R[A(m_2)=2]$

$$P_{m_2}^{(2)} = P_R[A(m_2)=2]$$

CASE 3 : $(k=1, b=1)$

Then $c = 1 \oplus m_1 = \sim m_1$ and $Pr[\mathcal{A}(c)=b] = Pr[A(\sim m_1)=1]$

$$P^{(1)}_{\sim m_1} = Pr[\mathcal{A}(\sim m_1)=1]$$

CASE 4 : $(k=1, b=2)$

Then $c = 1 \oplus m_2 = \sim m_2$ and $Pr[\mathcal{A}(c)=b] = Pr[A(\sim m_2)=2]$

$$P^{(2)}_{\sim m_2} = Pr[\mathcal{A}(\sim m_2)=2]$$

By the law of total probability, we get

$$Pr[\mathcal{A}(c)=b] = \frac{1}{4}\left(P^{(1)}_{m_1} + P^{(2)}_{m_2} + P^{(1)}_{\sim m_1} + P^{(2)}_{\sim m_2}\right)$$

Since, we know that for any cipher text $x \in \{0,1\}$, the adversary must either output 1 or 2, we have,

$$P^{(1)}_x + P^{(2)}_x = 1 \Rightarrow P^{(i)}_x + P^{(i)}_{\sim x} = 1 \text{ for any } i$$

$\therefore$

$$P^{(1)}_{m_1} + P^{(1)}_{\sim m_1} = 1 \text{ and } P^{(2)}_{m_2} + P^{(2)}_{\sim m_2} = 1$$

$$\Rightarrow Pr[\mathcal{A}(c)=b] = \frac{1}{4}(1+1) = \frac{1}{4}(2) = \frac{1}{2}$$

∴ If $m_1 = m_2$, then cipher texts are identical in both cases. So $\mathcal{A}$ receives the same input no matter which bit $b$ is used. Then $\mathcal{A}$'s sucess probability is exactly $1/2$.

And if $m_1 \neq m_2$, we calculated $1/2$.

∴ This holds regardless of whether $m_1 = m_2$ or $m_1 \neq m_2$, the adversary receives no information distinguishing the challenge bit $b$.

Thus $\mathcal{A}$ has no advantage.

$$\therefore Pr[\mathcal{A}(c) = b] = 1/2 \Rightarrow Adv_{\mathcal{A}}^{distinguish cipher}(1) = 0$$

Thus one-time pad for $\lambda = 1$ is perfectly secure in indistinguishability based encryption.

QED //

(b) *Using the same general argument as in the previous part, show that* $\mathsf{Adv}_{\mathcal{A}}^{\text{distinguishcipher}}(\lambda) = 0$ *for all* $\lambda$ *and for all adversaries,* $\mathcal{A}$.

Let $\mathcal{D}_1$ be the distribution of $c = k \oplus m$, and $\mathcal{D}_2$ be that of $c = k \oplus m_2$ with $k$ uniformly distributed over $\{0,1\}^\lambda$.

So $\mathcal{D}_1 = \mathcal{D}_2$

Since the distributions are identical, the adversary gain nothing regardless of its internal randomness.

$\therefore \Pr[\hat{b} = b] = 1/2$

$\Rightarrow \mathrm{Adv}_{\mathcal{A}}^{\text{distinguishcipher}}(\lambda) = \Pr[\hat{b} - b] - 1/2$

$= 1/2 - 1/2$

$= 0$

$\therefore$ For all security parameters $\lambda \in \mathbb{N}$ and all adversaries $\mathcal{A}$, the one-time pad has zero distinguishing advantage.

i.e., $\mathrm{Adv}_{\mathcal{A}}^{\text{distinguishcipher}}(\lambda) = 0$

$\therefore$ One-time pad is perfectly secure under the definition of indistinguishability.

Q.E.D

**Exercise 7.** *Using the security definitions we have seen so far as a model, define an attack game that calls on an adversary to distinguish between an output of $G$ and a truly random element. Define the advantage an adversary has in this game, and define what it means for a PRG, $G$, to be secure in terms of advantages.*

Let $G: \{0,1\}^{\ell} \longrightarrow \{0,1\}^{L}$ be a deterministic function with $L > \ell$.

The goal of the adversary is to distinguish between the output of $G$ (on a random seed) and a truly random string from $\{0,1\}^{L}$.

For a fixed security parameter $\lambda$, we define the adversary $\mathcal{A}$ is given access to a challenge string $y \in \{0,1\}^{L}$ which is generated as follows:

- A bit $b \in \{0,1\}$ is sampled uniformly at random.
- If $b=0$, the challenger selects a seed $s \in \{0,1\}^{\ell}$ uniformly at random and sets $y := G(s)$
- If $b=1$, the challenger samples $y$ uniformly at random from $\{0,1\}^{L}$.
- The string $y$ is then given to $\mathcal{A}$, which outputs a guess $\hat{b} \in \{0,1\}$.
- adversary wins if $\hat{b} = b$.

Let us define the probability that the adversary wins as

$$Pr[\text{DistinguishPRG}_{\mathcal{A}}(\lambda) = \text{true}] := Pr[\hat{b} = b]$$

where the probability is taken over the randomness used to select $b$, the randomness of the seed $s$ and any internal randomness of the adversary $\mathcal{A}$.

Then the advantage of the adversary is defined as

$$Adv_{\mathcal{A}}^{PRG}(\lambda) = \left| Pr[\hat{b} = b] - \frac{1}{2} \right|$$

This expression quantifies how much better the adversary is at distinguishing the PRG's output from truly random strings than random guessing.

ABOUT SECURITY:

We say $G$ is secure if for all probabilistic polynomial time adversaries $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{PRG}(\lambda)$ is negligible in $\lambda$.

i.e., $\exists$ a negligible function $\mu(\lambda)$ such that

$$Adv_{\mathcal{A}}^{PRG}(\lambda) \leq \mu(\lambda) \quad \forall \lambda \in \mathbb{N}.$$

This means the output of $G$ is computationally indistinguishable from uniform random strings. //

**Exercise 8.** *Using the definition of a secure cipher given in Figure 2, and your definition of a secure PRG from the previous exercise, use proof by reduction to show that if $G$ is a secure PRG, then $E_G$ is a secure encryption function. You may use the fact, that we will prove next week, that if $F(\lambda)$ is not negligible, then $\frac{F(\lambda)}{2}$ is also not negligible.*

*Hint: Given a program $\mathcal{A}$ that has a non-negligible advantage against the game in Figure 2 for $E_G$, construct an adversary program $A'(L, r)$ that plays your game from the previous exercise by using $r$ as a one-time pad key to encrypt a random message given by $A(L)$. To argue that $A'$ has a non-negligible advantage, use the fact that*

$$Pr[\mathcal{A}' \text{ wins}] = \frac{1}{2} \cdot Pr[\mathcal{A}' \text{ wins} \mid r \text{ is random}] + \frac{1}{2} \cdot Pr[\mathcal{A}' \text{ wins} \mid r = G(s)] .$$

Let us assume for contradiction that $E_G$ is not secure. Then by definition, there exist a PPT adversary $\mathcal{A}$ and a non-negligible function $\varepsilon(\lambda)$ such that

$$Adv_{\mathcal{A}}^{\text{distinguish cipher}}(\lambda) = \left| P_r(b' = b) - \frac{1}{2} \right| \geq \varepsilon(\lambda)$$

where $b'$ is the adversary's guess.

Let $c = E_G(k, m_b) = G(k) \oplus m_b$ with $k \in \{0,1\}^l$ chosen uniformly at random.

We will try to use this $\mathcal{A}$ to construct new $\mathcal{A}'$ that breaks the pseudorandomness of $G$ with non-negligible advantage.

$\mathcal{A}'$ participates in PRG distinguishing experiment where it receives an input $r \in \{0,1\}^L$ which is either $r = G(s)$ for some $s \in \{0,1\}^l$ or $r$ is uniformly random from $\{0,1\}^L$

Let the behaviour of $\mathcal{A}'$ as follows:

1. $\mathcal{A}'$ runs $\mathcal{A}(\lambda)$ to obtain $(m_1, m_2) \in \{0,1\}^L \times \{0,1\}^L$
2. It selects a random bit $b \in \{1,2\}$ and computes the ciphertext $c := r \oplus m_b$
3. $\mathcal{A}'$ gives $c$ to $\mathcal{A}$, receives a guess $\hat{b} \in \{1,2\}$ and output 0 if $\hat{b} = b$ or 1 otherwise.

Here $\mathcal{A}'$ uses $r$ as a keystream in one-time pad, limitating the behavior of $E_a$ if $r = G(k)$ or acting truly random.

Let $W$ be the event that $\mathcal{A}$ guesses $b$ correctly. The the probability of $\mathcal{A}'$ that outputs 0 is

$$Pr[\mathcal{A}' \text{ outputs } 0] = Pr[W]$$
$$= \frac{1}{2} Pr[W / r = G(s)] + \frac{1}{2} \cdot Pr[W / r \text{ choosen uniformly random from } \{0,1\}^L]$$

If $r = G(s)$, then $c = G(s) \oplus m_b$ is valid encryption under $E_a$, so $\mathcal{A}$ guesses $b$ with advantage $\varepsilon(\lambda)$.

Hence $Pr[W / r = G(s)] = \frac{1}{2} + \varepsilon(\lambda)$

If $r$ chosen uniformly random, then $c$ is a uniform random string, independent of $b$, so

$$Pr\left[W \,/\, r \text{ chosen uniformly random from } \{0,1\}^{\lambda}\right] = 1/2$$

Therefore, $Pr[W] = 1/2 \cdot (1/2 + \varepsilon(\lambda)) + 1/2 \cdot 1/2$

$$= 1/2 \left[ 1/2 + 1/2 + \frac{2\varepsilon(\lambda)}{2} \right]$$

$$= \frac{1}{2}\left[ \frac{2 + 2\varepsilon(\lambda)}{2} \right] = 1/2\left(1 + \varepsilon(\lambda)\right)$$

∴ The advantage of $A'$ is distinguishing $G$ from random is $Adv^{PRG}_{A'}(\lambda) = \left| Pr[A' \text{ output } 0] - 1/2 \right| = \varepsilon(\lambda)/2$

Since $\varepsilon(\lambda)$ is non-negligible and square of a non-negligible function is also non-negligible, then $\frac{\varepsilon(\lambda)}{2}$ is non-negligible.

This contradicts our assumption that $G$ is secure PRG.

∴ Our initial assumption that $A$ breaks the encryption scheme must be false.

∴ If $G$ is secure PRG, then encryption scheme $E_G$ defined by $E_G(k, m) = G(k) \oplus m$ is secure in indistinguishability based definition.

//