Problem 1: More Fun with OWFs and PRGs

Suppose that \( f() \) is a length-preserving one-way function, \( g() \) is a one-way permutation. Let \( G_1, G_2 : \{0,1\}^n \rightarrow \{0,1\}^{2n} \) (length-doubling) and \( G_3 : \{0,1\}^n \rightarrow \{0,1\}^{3n} \) (length-tripling) be PRGs (for every \( n \)), and let \( s \in \{0,1\}^k \). Let \( |x| \) denote the length of the binary string \( x \).

a. For the following functions, use a reduction to prove that the function is one-way, or give a counterexample showing that it is not one-way.
   
   (a) \( f_a(x) = g(f(x)) \)
   
   (b) \( f_b(x) = \begin{cases} \text{0} & |x| \\
                  g(x) & \text{otherwise.}
\end{cases} \)

b. For each of the following, prove that it is a PRG or provide a counterexample to show that it is not a PRG.

   (a) \( G_a(s) = G_1(s) \oplus G_2(s) \)

   (b) \( G_b(s) = G_b(s_1 \circ s_2) = G_3(s_1) \oplus G_3(s_2) \).

   (Note that \( \circ \) is the special operator that denotes dividing \( s \) into two equal-length strings \( s_1 \) and \( s_2 \). If \( s \) is of odd length, then \( \circ \) will split \( s \) into \( s_1, s_2 \) of equal lengths, ignoring the last bit of \( s \). Hence we have that \( |s_1| = |s_2| = \lfloor \frac{|s|}{2} \rfloor \).

Problem 2: Breaking our Assumptions

In both RSA and QR-based cryptosystems, we have implicitly assumed that factoring the public value \( n = pq \) is hard. Specifically, it has been crucial that the adversary cannot efficiently learn \( \varphi(n) = (p-1)(q-1) \). However, it is not even necessary for the adversary to learn \( \varphi \) exactly to break the scheme. There are in fact whole classes of values related to \( \varphi \) that are sufficient to mount an equally powerful attack.

a. RSA Warmup: Given an RSA public modulus \( n = pq \), public exponent \( e \), and the prime factors \( p \) and \( q \) of \( n \), show how to efficiently compute the private exponent \( d \).

b. Given an RSA public modulus \( n = pq \), public exponent \( e \), and \( \lambda(n) = \alpha \varphi(n) \), for some unknown positive integer \( \alpha \), show how to efficiently compute an exponent \( d' \) which will allow you to invert the RSA trapdoor permutation.

c. QR Warmup: Given a Blum integer \( n = pq \) (with \( p \equiv q \equiv 3 \mod 4 \)) and its prime factors \( p \) and \( q \), show how to efficiently distinguish between elements of \( QR_n \) and \( QNR_n \).

d. Given a Blum integer \( n = pq \) (with \( p \equiv q \equiv 3 \mod 4 \)) and \( \lambda(n) = \alpha \varphi(n) \), for some unknown positive odd integer \( \alpha \), show how to efficiently distinguish between elements of \( QR_n \) and \( QNR_n \).
e. Can you extend your scheme from above to efficiently handle even $\alpha$? You may assume that the bit length of $\alpha$ is polynomial in the security parameter $k$. If so, explain how to proceed when you don’t know if $\alpha$ is even or odd. If not, argue why such a scheme is impossible.

**Problem 3: Decisional Diffie-Hellman Assumption and PRGs**

Let $G$ be a cyclic group of prime order $q$ with generator $g$. Recall that a group is cyclic if it can be generated by a single element, which is called the generator. Hence each element $g^x$ for some $x \in \mathbb{Z}_q$. The decisional Diffie-Hellman (DDH) assumption states that for all $a, b, c \in \mathbb{Z}_q$ drawn uniformly, the following two distributions are computationally indistinguishable:

$$(g, g^a, g^b, g^{ab}) \approx (g, g^a, g^b, g^c)$$

a. Let $w = p(\log q)$ where $p()$ is some polynomial. Prove that under the DDH assumption, the following two distributions are indistinguishable:

$$(g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{a_1+b}, g^{a_2+b}, \ldots, g^{a_w+b}) \approx (g^{a_1}, g^{a_2}, \ldots, g^{a_w}, g^{c_1}, g^{c_2}, \ldots, g^{c_w})$$

where $a_1, b, c_1 \sim \mathbb{Z}_q$ are all uniformly random and independent.

b. Using what you proved in the previous part, construct a PRG $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$. What would you choose as your $m, n$ and what would be your seed? Assume that each of the $q$ elements in $G$ can be represented by binary strings of length $\lceil \log q \rceil$. Prove the security of your $G$ with the help of a reduction argument to the DDH assumption.

c. Let $p = 2q + 1$ also be a prime. Prove that DDH assumption does not hold in $\mathbb{Z}_p^*$.

**Problem 4: Variations of Semantic Security**

In class, we saw the definition of semantic security. A cryptosystem $(\text{KeyGen}, \text{Enc}, \text{Dec})$ is semantically secure for variable-length messages if there exists a simulator $\mathcal{S}$ such that for all polynomial length functions $\ell(k)$, for all sequences of messages $\{m^k\}_{k=1}^{\infty}$ where $m^k$ of length $\ell(k)$ the following two distributions are computationally indistinguishable:

$$\text{Real}(1^k, \ell(k), m^k) = \{(pk, sk) \leftarrow \text{KeyGen}(1^k); c \leftarrow \text{Enc}(pk, m^k) : (pk, c, \ell(k), m^k)\}$$

$$\text{Simulated}(1^k, \ell(k), m^k) = \{(pk, sk) \leftarrow \text{KeyGen}(1^k); c \leftarrow \mathcal{S}(pk) : (pk, c, \ell(k), m^k)\}$$

a. Consider a variation on the above definition. Instead of considering the fixed message $m^k$, it hides a randomly chosen message. A cryptosystem $(\text{KeyGen}, \text{Enc}, \text{Dec})$ is random-message-secure for variable-length messages if there exists a simulator $\mathcal{S}$ such that for all polynomial length functions $\ell(k)$, the following two distributions are computationally indistinguishable:

$$\text{Real}(1^k, \ell(k)) = \{(pk, sk) \leftarrow \text{KeyGen}(1^k); m \sim \{0, 1\}^{\ell(k)} ; c \leftarrow \text{Enc}(pk, m) : (pk, c, \ell(k), m)\}$$

$$\text{Simulated}(1^k, \ell(k)) = \{(pk, sk) \leftarrow \text{KeyGen}(1^k); m \sim \{0, 1\}^{\ell(k)} ; c \leftarrow \mathcal{S}(pk) ; (pk, c, \ell(k), m)\}$$

Is random-message security equivalent to semantic security? If your answer is yes, give reductions in both directions. If your answer is no, construct a cryptosystem satisfying one definition but not the other; you may assume that semantically secure encryption exists.
b. Consider another variation. Here, instead of requiring that a simulator exists, we require that the encryption of any message \( m^k \) be indistinguishable from an encryption of a random message. A cryptosystem \((KeyGen, Enc, Dec)\) is midterm-secure for variable-length messages if for all polynomial length functions \( \ell(k) \), for all sequences of messages \( \{m^k\}_{k=1}^{\infty} \) where \( m^k \) is of length \( \ell(k) \), the following two distributions are computationally indistinguishable:

\[
\text{Real}(1^k, \ell(k), m^k) = \{(pk, sk) \leftarrow KeyGen(1^k); c \leftarrow Enc(pk, m^k) : (pk, c, \ell(k), m^k)\}
\]

\[
\text{Random}(1^k, \ell(k), m^k) = \{(pk, sk) \leftarrow KeyGen(1^k); r \leftarrow \{0, 1\}^{\ell(k)}; c \leftarrow Enc(pk, r) : (pk, c, \ell(k), m^k)\}
\]

Is midterm-security equivalent to semantic security? If your answer is yes, give reductions in both directions. If your answer is no, construct a cryptosystem satisfying one definition but not the other; you may assume that semantically secure encryption exists.