Problem 1: Fun with Hash Functions

Recall that \((\text{ParamGen}, \text{Hash})\) is a family of collision-resistant hash functions (CRHF) if

**Collision-resistance:** \(\forall \text{ PPT } \mathcal{A} \exists \text{ negl. } \nu()\) such that

\[
\Pr[\text{params} \in \text{ParamGen}(1^k) ; x, y \leftarrow \mathcal{A}(1^k, \text{params}) : x \neq y \text{ and } \text{Hash}_{\text{params}}(x) = \text{Hash}_{\text{params}}(y)] = \nu(k)
\]

**Length-reducing:** For all \(\text{params} \in \text{ParamGen}(1^k)\),

\[
\text{Hash}(\text{params}, \cdot) : \{0,1\}^\ast \rightarrow \{0,1\}^k
\]

Once \(\text{params}\) are fixed, let \(H\) denote \(\text{Hash}_{\text{params}}\). Let \(H_1, H_2\) be length-reducing functions.

a. Use \(H\) to construct a hash function \(H_a\) over the same domain and range that is also collision resistant, but if one truncates the LSB of output of \(H_a\) then the new hash function is no longer collision resistant. More formally, let \(H_a(x) = y = y_1y_2 \ldots y_{k-1}y_k\). Design \(H_a\) such that it is collision resistant, but \(H'_a(x) = y_1y_2 \ldots y_{k-1}\) (output of \(H_a(x)\) with LSB truncated) is not.

b. We want to build a CRHF \(H_b\) using \(H_1\) and \(H_2\), so that if just one of \(H_1\) or \(H_2\) is not collision-resistant then \(H_b\) is still a CRHF. Let

\[
H_b(m) = (H_1(m), H_2(m))
\]

Show that \(H_b\) is a secure CRHF if either \(H_1\) or \(H_2\) is a CRHF.

c. Show that \(H_c(x) = H_1(H_2(x))\) might not be a secure CRHF even if one of \(H_1\) or \(H_2\) is CRHF.

d. Let \(F\) be a secure PRF. Show that \(F_d(k, m) = F(k, H(m))\) is a secure PRF. This shows that a collision-resistant hash function can be used to extend the domain of a PRF.

Problem 2: Broken Signatures

A signature scheme can be broken in a number of ways. It can be shown to be existentially forgeable, which means that the adversary can find some message for which it can forge a signature. It can be target-message forgeable, which means that an adversary can forge a signature for any given message. Or, in the worst case, its secret key can be completely recoverable.

In order to break a signature scheme, different flavors of attacks can be launched: a public-key only attack where the adversary only gets to see the public-key, or an interactive attack where the adversary asks for signatures on messages of his choice.

Consider the following signature scheme: the public key of this signature scheme consists of a Blum integer \(n = pq\) (recall that Blum integers have the property that \(-1\) is a nonsquare mod \(p\) and mod \(q\)). The secret key is \(n\)’s factorization. The message space is \(QR_n \cup QNR_n\). A signature \(\sigma\) on message \(m\) is computed as follows: if \(m\) is a quadratic residue, then \(\sigma\) is some arbitrary square
root of \( m \). Otherwise, \( \sigma \) is some arbitrary square root of \(-m\).

Prove the following facts about this signature scheme:

a. With a public-key only attack (i.e., without access to the signer, only the signer’s public key), a target-message attack on this signature scheme is as hard as factoring. (Give a reduction.)

b. This signature scheme is existentially forgeable with a public-key only attack. (Describe an attack.)

c. The secret key of this signature scheme can be recovered by an adversary making interactive queries to the signer. (Describe an attack.)

Problem 3: GHR Signature

Let’s look at a signature scheme that is both efficient and provably secure. This signature scheme is due to Gennaro, Halevi and Rabin.

Let \( P_k \) be the set of primes of length \( k \). Suppose we have a family of collision-resistant hash functions \( h_{pk} : \{0, 1\}^* \to P_k \), with key generation algorithm \( G \).

Consider the following scheme:

\[
\text{KeyGen}(1^k): \text{Choose random } k\text{-bit safe primes } p_1, p_2, q \text{ (a prime } p \text{ is safe if there exists prime } p' \text{ such that } p = 2p' + 1). \text{ Compute } n = p_1 p_2. \text{ Choose } s \leftarrow \mathbb{Z}_n^*, \text{ Choose } g_1, g_2 \leftarrow QR_q, \text{ such that } g_1, g_2 \neq 1 \text{ and } g_1 \neq g_2. \text{ Run } G(1^k) \text{ to obtain hash function key } pk.
\]

Output \( \text{PK} = (n, s, q, pk, g_1, g_2) \), \( \text{SK} = (p_1, p_2) \).

\[
\text{Sign}((n, s, q, pk), (p_1, p_2), m): \text{Choose random } r \leftarrow \mathbb{Z}^*_q. \text{ Let } H = h_{pk}(g_1^m g_2^r). \text{ Finally, use } p_1, p_2 \text{ to compute } \sigma = s^H. \text{ Output signature } (\sigma, r).
\]

\[
\text{Verify}((n, s, q, (\sigma, r), m): \text{Let } H = h_{pk}(g_1^m g_2^r). \text{ Accept iff } \sigma H = s.
\]

We will use the following assumptions:

**Discrete Log Assumption** Let \( \text{Gen}(1^k) \) be an algorithm that generates a \( k \)-bit prime \( q \), and a generator \( g \) of \( \mathbb{Z}_q^* \). We assume that for all PPT adversaries \( A \) there exists negligible \( \nu(\cdot) \) such that

\[
\Pr[(q, g) \leftarrow \text{Gen}(1^k); i \leftarrow \mathbb{Z}_q; i' \leftarrow A(q, g, g^i \mod q) : g^i = g^{i'} \mod q] = \nu(k).
\]

**Strong RSA Assumption** Let \( \text{Gen}(1^k) \) be an algorithm that generates 2 \( k \)-bit primes \( p_1, p_2 \), and outputs \( n = p_1 p_2 \). We assume that for all PPT adversaries \( A \) there exists negligible \( \nu(\cdot) \) such that

\[
\Pr[n \leftarrow \text{Gen}(1^k); y \leftarrow \mathbb{Z}_n^*; (x, e) \leftarrow A(n, y) : y = x^e \land e \neq 1] = \nu(k).
\]

We will assume that these assumptions hold when \( q, p_1, p_2 \) are safe primes.

We can show that if these assumptions hold, and if \( h \) is a family of collision-resistant hash functions, then the above signature scheme is existentially unforgeable even when the adversary is allowed to make an interactive attack in which he can request signatures for arbitrary messages.

Suppose there exists an adversary that can make such an attack and produce a valid forgery. Then we consider the following cases (note that in each one, you must describe a reduction which (1) processes its input and gives input to the adversary, (2) answers the adversary’s signing queries, and (3) processes the adversary’s forgery):
a. Suppose that with some non-negligible probability, after querying for signatures on messages $m_1, \ldots, m_l$, $A$ produces a successful forgery $(m^*, (\sigma^*, r^*))$ such that for some $i \in \{1, \ldots, l\}$, $m^* \neq m_i$ and $g_1^{m^*} g_2^{r^*} = g_1^{m_i} g_2^{r_i}$. Show that if such an adversary exists then the discrete logarithm assumption does not hold.

b. Suppose that with some non-negligible probability, after querying for signatures on messages $m_1, \ldots, m_l$, $A$ produces a successful forgery $(m^*, (\sigma^*, r^*))$ such that for some $i \in \{1, \ldots, l\}$, $g_1^{m^*} g_2^{r^*} \neq g_1^{m_i} g_2^{r_i}$ and $h_{pk}(g_1^{m^*} g_2^{r^*}) = h_{pk}(g_1^{m_i} g_2^{r_i})$. Show that if such an adversary exists then $h$ is not collision-resistant.

c. Suppose that with some non-negligible probability, after querying for signatures on messages $m_1, \ldots, m_l$, $A$ produces a successful forgery $(m^*, (\sigma^*, r^*))$ such that for all $i \in \{1, \ldots, l\}$, $h_{pk}(g_1^{m^*} g_2^{r^*}) \neq h_{pk}(g_1^{m_i} g_2^{r_i})$, but such that $\text{Verify}(PK, m^*, (\sigma^*, r^*))$ accepts. Then, as we will see in class, the strong RSA assumption does not hold.

Using this information and what you proved in the previous two parts, conclude that if the strong RSA and discrete logarithm assumptions hold, and if $h$ is a family of collision resistant hash functions, then the above signature scheme is existentially unforgeable against an interactive attack.

**Problem 4: Digital Signatures**

Prove that the existence of secure digital signature schemes implies the existence of one-way functions.