Homework 5

1 Problem 1 (20 points)

In this problem, we will see how to using hashing with message authentication codes where the hash function need not be collision resistant.

Let $(\mathsf{MAC}, \mathsf{Ver})$ be a secure message authentication code with message space $\{0,1\}^{n+r}$. Let H be a keyed hash function with domain $\{0,1\}^N$ and range $\{0,1\}^n$, where $N \gg n$. Suppose the key space for H is $\{0,1\}^r$.

Let $(\mathsf{MAC}', \mathsf{Ver}')$ be the following MAC with message space $\{0,1\}^N$.

- $\mathsf{MAC}'(k, m)$: choose a random hash key hk, and let $h \leftarrow H(\mathsf{hk}, m)$, and $\sigma \leftarrow \mathsf{MAC}(k, (\mathsf{hk}, h))$. Output $\sigma' = (\mathsf{hk}, \sigma)$.
- $\mathsf{Ver}'(k, m, \sigma')$: write $\sigma' = (\mathsf{hk}, \sigma)$. Compute $h \leftarrow H(\mathsf{hk}, m)$. Then run $\mathsf{Ver}(k, (\mathsf{hk}, h), \sigma)$, and output whatever Ver outputs.

Prove that this scheme is correct, and prove that it is secure assuming H is second pre-image resistant (aka target collision resistant).

2 Problem 2 (10 points)

Suppose you have a commitment scheme (with setup) $(\mathsf{Setup}, \mathsf{Com})$ that is computationally binding and computationally hiding, and has message space $\{0,1\}^n$.

(a) Explain how to use a collision resistant hash function (with appropriate domain and range) to get a commitment scheme $(\mathsf{Setup}', \mathsf{Com}')$ with a message space $\{0,1\}^N$ for $N \gg n$.

(b) Explain why an approach using second pre-image resistance as in **Problem 1** will not work for commitments.

3 Problem 3 (20 points)

(a) Let $(\mathsf{Setup}, \mathsf{Com})$ be a commitment scheme that is *perfectly binding*, and computationally hiding (for honest receivers).

Show how, given such a scheme, to construct a commitment scheme Com' without setup that is computationally hiding and perfectly binding. (Since there is no more setup in Com', there is no longer any distinction between malicious receiver and honest-but-curious receiver)

(b) Let $(\mathsf{Setup}, \mathsf{Com})$ be a commitment scheme that is *computationally binding* and computationally hiding. Suppose we additionally required that the scheme remains secure in the following scenario. Bob (the receiver) wants to let Alice (the sender) to run Setup to get the commitment key k. However, Alice is malicious, and may try to devise a bad key k that allows her to break binding. For a scheme where Alice can devise k however she wants, but for which (computational) binding still holds, we say the scheme is computationally binding for *malicious senders*.

Show, given such a scheme, how to construct a commitment scheme Com' without setup that is computationally hiding and computationally binding. (Since there is no more setup in Com', there is no longer any distinction between a malicious sender and an honest-but-curious sender)

4 Problem 4 (10 points)

Let $\mathbb{G}$ be a cyclic finite group of prime order p with generator g. Consider the following commitment scheme:

- The message space is $\mathbb{Z}_p$.
- $\mathsf{Setup}()$: choose a random $a \in \mathbb{Z}_p$, $a \neq 0$, and compute $h = g^a$. The commitment key is h.
- $\mathsf{Com}(h, m; r)$: output $g^m h^r$, where r is a random element in $\mathbb{Z}_p$.

(a) Show that the scheme is perfectly hiding.

(b) Show that the scheme is computationally binding, assuming the discrete log problem is hard for $\mathbb{G}$. Hint: show that if you know two openings (m_0, r_0) and (m_1, r_1) of the same commitment, then you can compute a, the discrete log of h.

5 Problem 5 (40 points)

(a) Let $\mathbb{G}$ be a cyclic finite group of order $2p$ where p is a prime. Show that the decisional Diffie Hellman problem does not hold in $\mathbb{G}$. Hint: given a tuple (g, h, u, v), try raising g, h, u, v to the power p.

(b) A number N is t-smooth if all of its prime factors are at most t. Let $\mathbb{G}$ be a cyclic finite group of order N, where N is the product of distinct prime factors and N is t-smooth for some small t (say, $t = \lambda^c$ for some constant c). Show that the discrete log problem is easy in $\mathbb{G}$: given any g and g^a, it is possible in polynomial time to recover a. The Chinese Remainder Theorem will be helpful here.