Problem 1-1. Hash Function Properties

Let $h : \{0,1\}^{\leq 2n} \to \{0,1\}^n$ be a hash function that is collision resistant. Let $h' : \{0,1\}^{\leq n+1} \to \{0,1\}^{n+1}$ be the hash function given by the rule

$$h'(x) = \begin{cases} 0||x & \text{if } x \in \{0,1\}^n \\ 1||h(x) & \text{otherwise} \end{cases}$$

(a) Prove that h' is not one-way.

Definition 1 *A function $f : X_n \to Y_n$ is said to be one-way if for every efficient adversary $\mathcal{A}$, the probability that $\mathcal{A}$, on input n and $y = f(x)$ for a random $x \in X_n$, outputs any x' such that $f(x') = y$, is negligible.*

Solution: The modified hash function h' is not one-way, since for any hash value y of the form $0||x$, a preimage is x. Therefore, we can find a preimage for at least one half of all possible hash values.

(b) Prove that h' is collision resistant.

Definition 2 *A function $f : X_n \to Y_n$ is said to be collision resistant (CR) if for every efficient adversary $\mathcal{A}$, the probability that $\mathcal{A}$ on input n, outputs any distinct $x, x' \in X_n$ such that $f(x) = f(x')$, is negligible.*

Solution:

Next, we prove that h' inherits collision resistance from h. We show that if we can find a collision for h', then we can easily do so for h. Suppose

$$\exists x_0 \neq x_1 : h'(x_0) \neq h'(x_1)$$

We have two cases:
(a) First bit of $h'(x_0)$ is 0. Impossible as implies $x_0 = x_1$.
(b) First bit of $h'(x_0)$ is 1. Then, $h(x_0) = h(x_1)$ a contradiction, as h is collision resistant.

(c) Prove that h' is target collision resistant if h is target collision resistant. For this problem part, assume that h has the form $h\colon \{0,1\}^{\leq n+1} \to \{0,1\}^n$.

Definition 3 *A function $f : X_n \to Y_n$ is said to be target collision resistant (TCR) if for every efficient adversary $\mathcal{A}$, the probability that $\mathcal{A}$ on input n and a random $x \in X_n$, outputs $x' \in X_n$ such that $x' \neq x$ and $f(x) = f(x')$, is negligible.*

Solution: Given an algorithm $\mathcal{A}$ that breaks TCR for h', we construct an algorithm $\mathcal{B}$ that breaks TCR for h, which is a contradiction.

The algorithm $\mathcal{B}$ operates as follows, given a random TCR challenge $x \in \{0,1\}^{\leq n+1}$:

- If $|x| = n$, output FAIL.
- Otherwise, run $x' \leftarrow \mathcal{A}(n, x)$.
- Output x'.

We first argue that, whenever $\mathcal{B}$ does not fail, it breaks the TCR property of h. Algorithm $\mathcal{A}$ produces a string x' such that $h'(x) = h'(x')$ and $x \neq x'$. Since h' is length-preserving, we know that $|x| = |x'| \neq n$. Then, by the definition of h' it holds that $1\|h(x) = 1\|h(x')$, which implies that $h(x) = h(x')$. Since $x \neq x'$, x' is a valid solution to the TCR challenge.

To complete the argument, we need only to show that $\mathcal{B}$ does not fail often. The algorithm $\mathcal{B}$ only outputs FAIL when the TCR challenge x has length n. (Algorithm $\mathcal{B}$ also fails if $\mathcal{A}$ fails, but the probability that $\mathcal{A}$ fails is non-negligible by assumption.) The probability of this bad event is at most

$$\frac{2^n}{(2^0 + 2^1 + 2^2 + \cdots + 2^{n-1} + 2^{n+1})} \leq \frac{2^n}{2^{n+2}} \leq \frac{1}{4}.$$

Therefore algorithm $\mathcal{B}$ breaks the TCR of h with probability better than negligible and we are done.

Problem 1-2. Message Authentication

(a) Let MAC be a secure message authentication code. Suppose Alice and Bob send authenticated messages to each other. Namely, every time one of them sends a message M they send it together with $\mathsf{MAC}(K, M)$ where K is their shared secret key. On day 1, Alice asks Bob if he wants to go to the movies, and Bob replies "yes". On day 2, Alice asks Bob if he wants to go to ice cream and Bob replies "no". On day 3, Alice asks Bob if he wants to rob a bank and Bob replies "no". Can an adversary Eve observing the communication on the first two days, corrupt Bob's message on the third day (in an authenticated way)? If so, how would you use a secure MAC so that the adversary cannot corrupt Bob's message?

Solution: The adversary can easily corrupt Bob's message since he already saw a MAC corresponding to the message "yes". One can overcome this attack by adding a time stamp to the message.

(b) Recall that the CMAC construction we saw in class is a sequential construction. Namely, to MAC a very long message that consists of L blocks (each of 128 bits), we need to do L sequential steps. Consider the following parallel construction: Let $F : K \times X \to \{0,1\}^k$ be a pseudorandom function (PRF). Let

$$\mathsf{MAC}(K, (M_1, \ldots, M_L)) = \oplus_{i=1}^{L} F(K, M_i),$$

where each $M_i \in X$. Is this a secure MAC (i.e., existentially unforgeable against adaptive chosen message attacks)?

Solution: No! The lack of order makes this MAC insecure, since $\mathsf{MAC}(K, (M_1, M_2)) = \mathsf{MAC}(K, (M_2, M_1))$.

Problem 1-3. Message Signing

Let $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme with message space $\{0,1\}^k$ (where k is the security parameter), and let H be a seeded hash function with domain $\{0,1\}^*$ and range $\{0,1\}^k$. Consider the new signature scheme $(\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$, with message space $\{0,1\}^*$, defined via the following "hash-then-sign" paradigm:

- Gen' runs Gen to generate a pair (sk, vk) and samples a seed s for H. It outputs $sk' = (sk, s)$ and $vk' = (vk, s)$.
- Sig' takes as input a secret key $sk' = (sk, s)$ and a message M, and outputs a $\mathsf{Sig}(sk, H_s(M))$, i.e., it signs the hashed message $H_s(M)$.
- Ver', given the verification key $vk' = (vk, s)$, a message M, and a signature σ, outputs 1 if and only if $\mathsf{Ver}(vk, H_s(M), \sigma) = 1$.

(a) Suppose that $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is secure (existentially against adaptive chosen message attack) then which of the following properties of H do we need to ensure that $(\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$ is also secure?

1. One-wayness.
2. Target collision resistance
3. Collision resistance.

Solution: Collision resistance. If an adversary can find two different messages M_1, M_2 such that $H_s(M_1) = H_s(M_2)$ then it can ask the oracle to sign M_1 and this signature is a valid signature also for M_2. Note that one-wayness and target collision resistance do not suffice, since it may be easy to find collisions in such functions. If H is collision resistant then the resulting signature scheme is secure since an adversary that asks (adaptively) to sign polynomially many messages $M_1, \ldots, M_t$, and obtains signatures $\sigma_i = \mathsf{Sig}(sk, H_s(M_i))$ for every $i \in [t]$, cannot forge a signature to a new message for the following reason:

Suppose he forges a signature to a new message M^*. If $H_s(M^*) \neq H_s(M_i)$ for every $i \in [t]$ then this adversary can be used to break the security of the original signature scheme, and if $H_s(M^*) = H_s(M_i)$ for some $i \in [t]$ then this adversary found a collision thus breaking the collision resistance assumption.

Problem 1-4. Pseudo-Random Functions

Let F be a pseudorandom function (PRF) that takes messages in $\{0,1\}^n$ to messages in $\{0,1\}^n$.

(a) We wish to use F to construct a PRF that takes messages in $\{0,1\}^{2n}$ to messages in $\{0,1\}^{2n}$. Suppose the key to F is also in $\{0,1\}^n$.

Below are four proposals of such a PRF, where $x_0, x_1, K, K_0, K_1 \in \{0,1\}^n$ and where we use $||$ to denote concatenation. Notice that the constructions 1, 3, and 4 use a key in $\{0,1\}^n$ whereas the second construction uses a key in $\{0,1\}^{2n}$.

1. $G_1(K, x_0||x_1) = F(K, x_0)||F(K, x_1)$.
2. $G_2(K_0||K_1, x_0||x_1) = F(K_0, x_0)||F(K_1, x_1)$.
3. $G_3(K, x_0||x_1) = F(K', 0^n)||F(K', 1^n)$, where $K' = F(F(K, x_0), x_1)$
4. $G_4(K, x_0||x_1) = F(F(K, x_0), x_1)||F(F(K, x_0), x_1 \oplus 1^n)$.

Only one of the above four proposals is a secure PRF. Which one is the secure one? For each of the three others, show an attack that distinguishes it from a truly random function (recall the definition of a PRF given in the lecture).

Solution: G_3 is the only PRF.

G_1 is not a PRF since by querying it on a single query of the from $x||x$ for any $x \in \{0,1\}^n$, the outcome will have the form $y||y$ for $y \in \{0,1\}^n$, as opposed to being truly random in $\{0,1\}^{2n}$.

G_2 is not a PRF since by querying it on (x_0, x_1) and (x_0', x_1) the two outcomes will not look random (jointly) since the last n bits will be the same in both!

G_4 is not a PRF since by querying it on $x_0||x_1$ and $x_0||x_1 \oplus 1^n$ one can distinguish the output from uniform.