CS 171: Discussion Section 7 (March 4)

1 One-way Functions

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function (OWF), and

$$\text{let } g(x) = f(x) \oplus x$$

Is $g(x)$ necessarily a one-way function? Prove your answer. Note: In your answer, you may use a secure OWF $h : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$.

Solution

Claim 1.1. *$g(x)$ is* not *necessarily a one-way function.*

Proof. We will construct a one-way function f such that when g is constructed from f, then g is insecure. Note that we must actually prove that our construction of f is a secure OWF.

1. Construction of f: Our construction of f will use another OWF $h : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$.

   Next, let the input to f take the form $x = (x_0, x_1) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$. Then,

   $$\text{let } f(x) = 0^{n/2} || h(x_0)$$

2. **Claim 1.2.** *f is a one-way function.*

   Proof.

   (a) Assume toward contradiction that f is not a OWF. Then there is an adversary $\mathcal{A}$ that wins the OWF security game for f with non-negligible probability. We will use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that wins the OWF security game for h with non-negligible probability. This implies that h is not a secure OWF, which is a contradiction. Therefore, our original assumption was false, and in fact, f is a (secure) OWF.

   (b) Let us recall the OWF function security game for f:

      i. The challenger samples $x \leftarrow \{0,1\}^n$ and computes $f(x)$. Then they send $f(x)$ to the adversary $\mathcal{A}$.

      ii. $\mathcal{A}$ outputs x'.

      iii. The adversary wins if $f(x') = f(x)$, and they lose otherwise.

      If f is not a OWF, then there exists an adversary $\mathcal{A}$ that wins the OWF security game for f with probability $\mathsf{non\text{-}negl}(n)$.

   (c) Now we will use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that wins the OWF security game for h with non-negligible probability.

      Construction of $\mathcal{B}$:

      i. $\mathcal{B}$'s challenger samples $x_0 \leftarrow \{0,1\}^{n/2}$ and sends $h(x_0)$ to $\mathcal{B}$.

ii. $\mathcal{B}$ computes the string $0^{n/2}||h(x_0)$ and runs $\mathcal{A}(0^{n/2}||h(x_0))$ to obtain $(x_0', x_1') \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$.

iii. $\mathcal{B}$ outputs x_0' as a preimage of $h(x_0)$.

(d) Analysis: First, note that $\mathcal{B}$ correctly simulates the OWF security game for f with $\mathcal{A}$ as the adversary. $\mathcal{A}$ is supposed to receive $f(x)$, where $x \in \{0,1\}^n$ is sampled uniformly. Since $x_0 \in \{0,1\}^{n/2}$ was sampled uniformly by $\mathcal{B}$'s challenger, then the distribution of $0^{n/2}||h(x_0)$ is the same as the distribution of $f(x)$ for a uniformly random x.

Next, with non-negligible probability, $\mathcal{A}$ will win the simulated security game for f, and in this case $\mathcal{B}$ will win the security game for h. With non-negligible probability, $\mathcal{A}$ will output an (x_0', x_1') such that

$$f(x_0', x_1') = 0^{n/2}||h(x_0)$$

In this case, $h(x_0') = h(x_0)$. Therefore, $\mathcal{B}$'s output, x_0', will win the security game for h.

(e) Since $\mathcal{B}$ wins the security game for h with non-negligible probability, this implies that h is not secure. This is a contradiction because we were told that h is secure. Therefore, our initial assumption was wrong, and in fact, f is also a secure OWF.

□

3.

Claim 1.3. *For the particular choice of f given above, g is not a secure one-way function.*

Proof.

(a) To summarize the constructions above, let $x = (x_0, x_1) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$. Then,

$$\begin{aligned} g(x) &= \big(0^{n/2}||h(x_0)\big) \oplus (x_0, x_1) \\ &= x_0||\big(h(x_0) \oplus x_1\big) \end{aligned}$$

(b) Now we will construct an adversary $\mathcal{C}$ that breaks the OWF security of g.
Construction of $\mathcal{C}$:

i. $\mathcal{C}$'s challenger samples $x \leftarrow \{0,1\}^n$ sends $g(x) = x_0||\big(h(x_0) \oplus x_1\big)$ to $\mathcal{C}$.

ii. From this input, $\mathcal{C}$ learns x_0 and $h(x_0) \oplus x_1$.
Then $\mathcal{C}$ computes $h(x_0)$ and then $x_1 = h(x_0) \oplus x_1 \oplus h(x_0)$.

iii. Finally, $\mathcal{C}$ outputs (x_0, x_1).

(c) $\mathcal{C}$ will successfully compute (x_0, x_1) given $g(x_0, x_1)$, so $\mathcal{C}$ wins the OWF security game for g with probability 1. Therefore, g is not a secure OWF.

□

□

2 Composed Hash Functions

We will show how to compose mulitiple hash functions to increase their compression factor. Let (Gen_1, H_1) and (Gen_2, H_2) be two fixed-length collision-resistant hash functions (CRHFs), where:

- $H_1^{s_1}$ maps $\mathcal{X} \to \mathcal{Y}$, for any seed $s_1 \leftarrow \mathsf{Gen}_1(1^n)$,
- $H_2^{s_2}$ maps $\mathcal{Y} \to \mathcal{Z}$, for any seed $s_2 \leftarrow \mathsf{Gen}_2(1^n)$, and
- $|\mathcal{X}| > |\mathcal{Y}| > |\mathcal{Z}|$

Define a new hash function $(\mathsf{Gen}_{\mathsf{comp}}, H_{\mathsf{comp}})$ to be the composition of H_2 and H_1:

1. $\mathsf{Gen}_{\mathsf{comp}}(1^n)$: Sample $s_1 \leftarrow \mathsf{Gen}_1(1^n)$ and $s_2 \leftarrow \mathsf{Gen}_2(1^n)$, and output $s = (s_1, s_2)$.
2. $H^s_{\mathsf{comp}}(x)$: Let $x \in \mathcal{X}$. Output $H_2^{s_2}(H_1^{s_1}(x))$.

Prove that $(\mathsf{Gen}_{\mathsf{comp}}, H_{\mathsf{comp}})$ is a secure collision-resistant hash function.

Solution

Theorem 2.1. $(\mathsf{Gen}_{\mathsf{comp}}, H_{\mathsf{comp}})$ *is a (secure) collision-resistant hash function.*

Proof.

1. Overview: We will show that if there were an adversary that could break the CRHF security of $(\mathsf{Gen}_{\mathsf{comp}}, H_{\mathsf{comp}})$, by finding a collision with non-negligible probability, then we could use the collision in H_{comp} to find a collision in H_1 or H_2. This would allow us to break the security of H_1 or H_2.

2. The Collision-Finder algorithm below uses a collision in H^s_{comp} to find a collision in $H_1^{s_1}$ or $H_2^{s_2}$. Recall that a collision in H^s_{comp} is two values $x, x' \in \mathcal{X}$ such that $x \neq x'$, and $H^s_{\mathsf{comp}}(x) = H^s_{\mathsf{comp}}(x')$.

   Collision-Finder(s, x, x'):

   (a) Compute $y = H_1^{s_1}(x)$ and $y' = H_1^{s_1}(x')$.
   (b) If $y = y'$, then output (x, x') as the collision in $H_1^{s_1}$.
   (c) If $y \neq y'$, then output (y, y') as the collision in $H_2^{s_2}$.

   Claim 2.2. *If* (x, x') *is a collision in* H^s_{comp}*, then* Collision-Finder(s, x, x') *outputs a collision in* $H_1^{s_1}$ *or a collision in* $H_2^{s_2}$*.*

   Proof. If $y = y'$, then (x, x') are a collision in $H_1^{s_1}$ because $H_1^{s_1}(x) = H_1^{s_1}(x')$, and $x \neq x'$. Next, if $y \neq y'$, then (y, y') are a collision in $H_2^{s_2}$ because

   $$H_2^{s_2}(y) = H^s_{\mathsf{comp}}(x) = H^s_{\mathsf{comp}}(x') = H_2^{s_2}(y')$$

   □

3. Let's recall the CRHF security game for a hash function (Gen, H):
   (a) The challenger samples a key $s \leftarrow \mathsf{Gen}(1^n)$ and sends s to the adversary.
   (b) The adversary outputs two values x, x' in the domain of H^s.
   (c) The adversary wins the game if $x \neq x'$ and $H^s(x) = H^s(x')$, and they lose otherwise.

4. Assume toward contradiction that H_{comp} is insecure. Then there is an adversary $\mathcal{A}$ for H_{comp}'s security game that finds a collision in H_{comp} with non-negligible probability.

   Next, we will construct adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ that try to find collisions in H_1 and H_2, respectively.

   $\underline{\mathcal{B}_1}$:

   (a) The challenger in the security game for H_1 samples a key $s_1 \leftarrow \mathsf{Gen}_1(1^n)$ and sends s_1 to $\mathcal{B}_1$.
   (b) $\mathcal{B}_1$ samples $s_2 \leftarrow \mathsf{Gen}_2(1^n)$ and sets $s = (s_1, s_2)$.
   (c) $\mathcal{B}_1$ runs $\mathcal{A}(s)$, which outputs two values $x, x' \in \mathcal{X}$.
   (d) $\mathcal{B}_1$ runs $\mathsf{Collision\text{-}Finder}(s, x, x')$ to try to find a collision in $H_1^{s_1}$. If successful, $\mathcal{B}_1$ outputs the collision.

   We can also construct an adversary $\mathcal{B}_2$ for the H_2 security game using an almost-identical construction to $\mathcal{B}_1$.

   $\underline{\mathcal{B}_2}$:

   (a) The challenger in the security game for H_2 samples a key $s_2 \leftarrow \mathsf{Gen}_2(1^n)$ and sends s_2 to $\mathcal{B}_2$.
   (b) $\mathcal{B}_2$ samples $s_1 \leftarrow \mathsf{Gen}_1(1^n)$ and sets $s = (s_1, s_2)$.
   (c) $\mathcal{B}_2$ runs $\mathcal{A}(s)$, which outputs two values $x, x' \in \mathcal{X}$.
   (d) $\mathcal{B}_2$ runs $\mathsf{Collision\text{-}Finder}(s, x, x')$ to try to find a collision in $H_2^{s_2}$. If successful, $\mathcal{B}_2$ outputs the collision.

5. Note that $\mathcal{B}_1$ and $\mathcal{B}_2$ correctly simulate the H_{comp} security game with $\mathcal{A}$ as the adversary.

   Therefore, when $\mathcal{B}_1$ or $\mathcal{B}_2$ runs $\mathcal{A}$, $\mathcal{A}$ will output a collision in H_{comp} with non-negligible probability.

6. Next,

   $$\Pr[\mathcal{A} \text{ wins the } H_{\mathsf{comp}} \text{ sec. game}] = \Pr[\mathcal{B}_1 \text{ wins the } H_1 \text{ sec. game}] + \Pr[\mathcal{B}_2 \text{ wins the } H_2 \text{ sec. game}]$$

   This is because whenever $\mathcal{A}$ outputs a collision in H^s_{comp}, it yields either a collision in $H_1^{s_1}$ or a collision in $H_2^{s_2}$.

7. Since $\Pr[\mathcal{A}$ wins the H_{comp} sec. game$]$ is non-negligible, then either $\Pr[\mathcal{B}_1$ wins the H_1 sec. game$]$ is non-negligible or $\Pr[\mathcal{B}_2$ wins the H_2 sec. game$]$ is non-negligible. That means that

either H_1 is insecure or H_2 is insecure[1]. In either case, this is a contradiction because H_1 and H_2 are secure CRHFs. Therefore, our initial assumption was false, and in fact, H_{comp} is also a secure CRHF.

$\square$

[1] We can't say which one of the hash functions is insecure; it depends on the particular algorithm for $\mathcal{A}$.