Cryptography – Homework 3

冯诗伟 161220039

1

a

No. Construct a new message $m' = m_2||m_1||m_3||\cdots||m_\ell$ by swapping the first two blocks of m and use m' to query the oracle. We can get $t' = \mathcal{O}(m') = \mathsf{Mac_k}(m') = F_k(m_2) \oplus F_k(m_1) \oplus F_k(m_3) \oplus \cdots \oplus F_k(m_\ell) = \mathsf{Mac_k}(m)$. So we successfully forge a valid tag t' such that $\mathsf{Vrfy_k}(m, t') = 1$ and $(m, t') \notin \mathcal{Q}$.

b

No. Given $m = m1||m2$, we can easily find two messages $m1', m2'(m1' \neq m1,\ m2' \neq m2)$. Then we can construct two new messages m_A and m_B, where $m_A = m1'||m2$, $m_B = m1||m2'$.

Query the oracle with m_A, we can get $\mathcal{O}(m_A) = F_k(m_1')||F_k(F_k(m_2))$. Query the oracle with m_B, we can get $\mathcal{O}(m_b) = F_k(m_1)||F_k(F_k(m_2'))$. By concatenating the former half of $\mathcal{O}(m_b)$ with the latter half of $\mathcal{O}(m_A)$, we can forge a valid tag $t = F_k(m_1)||F_k(F_k(m_2)) = \mathsf{Mac}_k(m)$, where $\mathsf{Vrfy_k}(m, t) = 1$ and $(m, t) \notin \mathcal{Q}$.

c

No. Given $m = m_1||m_2||m_3||\cdots||m_\ell$, we can construct the following three messages:

$$m_A = m_1||m_2||m_2||m_4||m_5||\cdots||m_\ell$$

$$m_B = m_2||m_2||m_2||m_4||m_5||\cdots||m_\ell$$

$$m_C = m_2||m_2||m_3||m_4||m_5||\cdots||m_\ell$$

Then we query the oracle with these three new messages and produce a tag t by XOR the three responses.

$$\begin{aligned}
t &= \mathcal{O}(m_A) \oplus \mathcal{O}(m_b) \oplus \mathcal{O}(m_c) \\
&= F_k(<1>|m_1) \oplus F_k(<2>|m_2) \oplus F_k(<3>|m_2) \oplus \oplus_{i=4}^{\ell} F_k(<i>|m_i) \\
&\oplus F_k(<1>|m_2) \oplus F_k(<2>|m_2) \oplus F_k(<3>|m_2) \oplus \oplus_{i=4}^{\ell} F_k(<i>|m_i) \\
&\oplus F_k(<1>|m_2) \oplus F_k(<2>|m_2) \oplus F_k(<3>|m_3) \oplus \oplus_{i=4}^{\ell} F_k(<i>|m_i) \\
&= F_k(<1>|m_1) \oplus F_k(<2>|m_2) \oplus F_k(<3>|m_3) \oplus \oplus_{i=4}^{\ell} F_k(<i>|m_i) \\
&= F_k(<1>|m_1) \oplus F_k(<2>|m_2) \oplus F_k(<3>|m_3) \oplus \cdots \oplus F_k(<\ell>|m_\ell) \\
&= \mathsf{Mac}_k(m)
\end{aligned}$$

So we successfully forge a valid tag t where $\mathsf{Vrfy_k}(m, t) = 1$ and $(m, t) \notin \mathcal{Q}$.

d

No. Given $m = m_1||m_2||m_3||m_4||m_5||\cdots||m_\ell$, we can first construct the following messages:

$$m_A = m_1||m_2||m_2||m_4||m_5||\cdots||m_\ell$$

Then we query the oracle with m_A and get the response:

$$\begin{aligned}\mathcal{O}(m_A) &= (r, m_r)\\ &= (r,\ F_k(r) \oplus F_k(< 1 > |m_1) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_2) \oplus \oplus_{i=4}^{\ell} F_k(< i > |m_i))\end{aligned}$$

We can parse r as $r =< x > |r'$, where the former half is the $\frac{n}{2}$-bit encoding of the integer x.

Construct the second message:

$$m_B = m_2||m_2||m_2||m_4||m_5||\cdots||r'||\cdots||m_\ell$$

where r' is the x-th block and $\ell = 2^{\frac{n}{2}}$. We can query the oracle with m_A mutiple times until $x \geq 4$.

Then we query the oracle with m_B and get the response:

$$\begin{aligned}\mathcal{O}(m_B) &= (p, m_p)\\ &= (p,\ F_k(p) \oplus F_k(< 1 > |m_2) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_2) \oplus F_k(< x > |r') \oplus \oplus_{i=5}^{2^{\frac{n}{2}}} F_k(< i > |m_i))\end{aligned}$$

Similarly, We can parse p as $p =< y > |p'$, where the former half is the $\frac{n}{2}$-bit encoding of the integer y.

Construct the third message:

$$m_C = m_2||m_2||m_3||m_4||m_5||\cdots||p'||\cdots||m_\ell$$

where p' is the y-th block and $\ell = 2^{\frac{n}{2}}$. We can query the oracle with m_B mutiple times until $y \geq 4$.

Then we query the oracle with m_C and get the response:

$$\begin{aligned}\mathcal{O}(m_C) &= (q, m_q)\\ &= (q,\ F_k(q) \oplus F_k(< 1 > |m_2) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_3) \oplus F_k(< y > |p') \oplus \oplus_{i=5}^{2^{\frac{n}{2}}} F_k(< i > |m_i))\end{aligned}$$

Notice that $F_k(< x > |r') = F_k(r),\ F_k(< y > |p') = F_k(p)$, we can produce the tag $t = (q, M)$ and

$$\begin{aligned}M &= m_r \oplus m_p \oplus m_q\\ &= F_k(r) \oplus F_k(< 1 > |m_1) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_2) \oplus \oplus_{i=4}^{\ell} F_k(< i > |m_i)\\ &\oplus F_k(p) \oplus F_k(< 1 > |m_2) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_2) \oplus F_k(< x > |r') \oplus \oplus_{i=5}^{2^{\frac{n}{2}}} F_k(< i > |m_i)\\ &\oplus F_k(q) \oplus F_k(< 1 > |m_2) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_3) \oplus F_k(< y > |p') \oplus \oplus_{i=5}^{2^{\frac{n}{2}}} F_k(< i > |m_i)\\ &= F_k(q) \oplus F_k(< 1 > |m_1) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_3) \oplus \oplus_{i=4}^{\ell} F_k(< i > |m_i)\\ &\oplus F_k(r) \oplus F_k(< x > |r') \oplus F_k(p) \oplus F_k(< y > |p')\\ &= F_k(q) \oplus F_k(< 1 > |m_1) \oplus F_k(< 2 > |m_2) \oplus F_k(< 3 > |m_3) \oplus \oplus_{i=4}^{\ell} F_k(< i > |m_i)\end{aligned}$$

The discussion above is based on the condition that x and y are greater than 3.

$$\begin{aligned}\Pr[x \geq 4 \wedge y \geq 4] &= \Pr[x \geq 4] \Pr[y \geq 4]\\ &= (1 - 2^{\frac{n}{2}-2})^2\end{aligned}$$

So we can construct an adversary $\mathcal{A}$ conducting the steps mentioned above so that

$$\Pr[\mathsf{Mac-sforge}_{\mathcal{A},\Pi}] = 1 = (1 - 2^{\frac{n}{2}-2})^2$$

which is non-negligible.

With a non-negligible probability we successfully forge a valid tag $t = (q, M)$ where $\mathsf{Vrfy_k}(m, t) = 1$ and $(m, t) \notin \mathcal{Q}$, so this MAC is not strongly secure.

2

a

This is not collision-resistant.

I am going to show that a compression function can be constructed such that even if the compression function is collision-resistant the resulting hash function may not be collision-resistant.

Suppose that $f : \{0,1\}^{2n} \to \{0,1\}^{\frac{n}{2}}$ is a collision-resistant function. Based on f, we can construct another function $h : \{0,1\}^{2n} \to \{0,1\}^n$ as follows:

$$h(x) = \begin{cases} 0^n, & x = 0^n || x^*, \text{ where } x^* \text{ is a specific constant string in } \{0,1\}^n \\ 1^{\frac{n}{2}} || f(x), & \text{otherwise} \end{cases}$$

We can show that h is also collision-resistant. If there exists x_a and $x_b (x_a \neq x_b)$ such that $h(x_a) = h(x_b)$, there must be $1^{\frac{n}{2}} || f(x_a) = 1^{\frac{n}{2}} || f(x_b)$ because x_a and x_b cannot be $0^n || x^*$ at the same time. So we have $f(x_a) = f(x_b)$, contradicting the fact that f is collision-resistant.

Construct two messages x, x':

$$x = x^* || x_2,\ x' = x_2,\ x_2 \in \{0,1\}^n$$

Remember that the IV of Construction 3.5 is 0^n. We have:

$$H^s(x) = H^s(x^* || x_2) = H^s\Big(h^s(0^n || x^*) \,||\, x_2\Big) = h^s(0^n || x_2)$$

$$H^s(x') = h^s(0^n || x_2)$$

So far we have found a collision for H, proving that the resulting hash function is not collision-resistant.

b

This is collision-resistant. We show that for any s, a collision in modified H^s yields a collision in h^s, thereby proving that the modified version is collision-resistant.

Let x and x' be two different strings of length L and L' respectively, such that $H^s(x) = H^s(x')$. Let $x_1, \cdots, x_B$ be the B blocks of the padded x, and let $x'_1, \cdots, x'_B$ be the B' blocks of the padded x'. Recall that $x_{B+1} = L$ and $x'_{B+1} = L'$. There are two cases to consider:

1. Case1: $L \neq L'$. In this case, the last step of the computation of $H^s(x)$ is $z_{B+1} = z_B || L$, and the last step of the computation of $H^s(x')$ is $z'_{B'+1} = z'_{B'} || L'$. Since $H^s(x) = H^s(x')$ it follows that $z_B || L = z'_{B'} || L'$. However, $L \neq L'$ and so $z_B || L$ and $z'_{B'} || L'$ are two different strings that collide under h^s.

2. Case2: $L = L'$. This means that $B = B'$. Let $z_0, \cdots, z_{B+1}$ be the values defined during the computation of $H^s(x)$, let $I_i \overset{def}{=} z_{i-1} || x_i$ denote the i-th input to h^s, $1 \leq i \leq B+1$, and set $I_{B+2} \overset{def}{=} z_{B+1} = z_B || L$. Define

$I'_1, \cdots, I'_{B+2}$ analogously with respect to x'. Let N be the largest index for which $I_N \neq I'_N$. Since $|x| = |x'|$ but $x \neq x'$, there is an i with $x_i \neq x'_i$ and so such that an N certainly exists. Because

$$I_{B+2} = z_{B+1} = H^s(x) = H^s(x') = z'_{B+1} = I'_{B+2}$$

we have $N \leq B+1$. By maximality of N, we have $I_{N+1} = I'_{N+1}$ and in particular $z_N = z'_N$. This means that I_N, I'_N are a collision in h^s.

In conclusion, the modified Merkle-Damgård Tranform is collision-resistant.

c

This is collision-resistant. We show that for any s, a collision in modified H^s yields a collision in h^s, thereby proving that the modified version is collision-resistant.

Let x and x' be two different strings of length L and L' respectively, such that $H^s(x) = H^s(x')$. Let $x_1, \cdots, x_B$ be the B blocks of the padded x, and let $x'_1, \cdots, x'_B$ be the B' blocks of the padded x'. Recall that $x_{B+1} = L$ and $x'_{B+1} = L'$. There are two cases to consider:

1. Case1: $L \neq L'$. In this case, the last step of the computation of $H^s(x)$ is $z_{B+1} = z_B||L$, and the last step of the computation of $H^s(x')$ is $z'_{B'+1} = z'_{B'}||L'$. Since $H^s(x) = H^s(x')$ it follows that $z_B||L = z'_{B'}||L'$. However, $L \neq L'$ and so $z_B||L$ and $z'_{B'}||L'$ are two different strings that collide under h^s.

2. Case2: $L = L'$. This means that $B = B'$. Let $z_0, \cdots, z_{B+1}$ be the values defined during the computation of $H^s(x)$, let $I_i \overset{def}{=} z_{i-1}||x_i$ denote the i-th input to h^s, $2 \leq i \leq B+1$, and set $I_1 = z_1 = x_1$, $I_{B+2} \overset{def}{=} z_{B+1} = z_B||L$. Define $I'_1, \cdots, I'_{B+2}$ analogously with respect to x'. Let N be the largest index for which $I_N \neq I'_N$. Since $|x| = |x'|$ but $x \neq x'$, there is an i with $x_i \neq x'_i$ and so such that an N certainly exists. Because

$$I_{B+2} = z_{B+1} = H^s(x) = H^s(x') = z'_{B+1} = I'_{B+2}$$

we have $N \leq B+1$. By maximality of N, we have $I_{N+1} = I'_{N+1}$ and in particular $z_N = z'_N$. This means that I_N, I'_N are a collision in h^s.

In conclusion, the modified Merkle-Damgård Tranform is collision-resistant.

d

This is not collision-resistant.

I am going to show that a compression function can be constructed such that even if the compression function is collision-resistant the resulting hash function may not be collision-resistant.

Suppose that $f : \{0,1\}^{2n} \to \{0,1\}^{\frac{n}{2}}$ is a collision-resistant function. Based on f, we can construct another function $h : \{0,1\}^{2n} \to \{0,1\}^n$ as follows:

$$h(x) = \begin{cases} < L_0 >, & x = < L_0 + n > || \, x^*, \text{ where } x^* \text{ is a specific constant string in } \{0,1\}^n \\ < \ell \oplus 1^{\frac{n}{2}} > ||f(x), & \text{otherwise, where } \ell \in \{0,1\}^{\frac{n}{2}} \text{ is the former half of } < L_0 > \end{cases}$$

Let me explain it in detail. L_0 is a specific constant interger and is a integer mutiple of the n. $< i > \in \{0,1\}^n$ denotes the n-bit encoding of i. In the second case, $< \ell \oplus 1^{\frac{n}{2}} >$ is used to prevent from colliding with the first case.

We can show that h is also collision-resistant. If there exists x_a and $x_b (x_a \neq x_b)$ such that $h(x_a) = h(x_b)$, there must be $< \ell \oplus 1^{\frac{n}{2}} > ||f(x_a) = < \ell \oplus 1^{\frac{n}{2}} > ||f(x_b)$ because x_a and x_b cannot be $< L_0 + n > ||x^*$ at the same time. So we have $f(x_a) = f(x_b)$, contradicting the fact that f is collision-resistant.

Construct two messages $x,\ x'$, where $|x| = L_0 + n = Bn + n,\ |x'| = L_0 = Bn$:

$$x = x^*||x_1||x_2||\cdots||x_B$$

$$x' = x_1||x_2||\cdots||x_B$$

Remember that the IV here is the length of the input. We have:

$$H^s(x) = H^s(x^*||x_1||x_2||\cdots||x_B) = H^s\Big(h^s(< L_0{+}n > ||x^*)\,||x_1||x_2||\cdots||x_B\Big) = H^s\Big(h^s(< L_0 > ||x_1)\,||x_2||\cdots||x_B\Big)$$

$$H^s(x') = H^s(x_1||x_2||\cdots||x_B) = H^s\Big(h^s(< L_0 > ||x_1)\,||x_2||\cdots||x_B\Big)$$

So far we have found a collision for H, proving that the resulting hash function is not collision-resistant.

Additional 3.26

Define $\Pi = (Gen, H)$, $\Pi_1 = (Gen_1, H_1)$, $\Pi_2 = (Gen_2, H_2)$. Suppose Π is not collision-resistant. Then there exists an PPT adversary $\mathcal{A}$ that $\mathcal{A}$ can found a collision in H with a non-negligible probability $\epsilon(n)$:

$$\Pr[\mathsf{Hash-coll}_{\mathcal{A},\Pi} = 1] = \epsilon(n)$$

Now we can construct $\mathcal{A}_1$ with $\mathcal{A}$:

$\mathcal{A}_1$ is given s_1, s_2.

1. Run $\mathcal{A}(s_1, s_2)$ and obtain $x,\ x'$.

2. Output $x,\ x'$.

$\mathcal{A}_1$ runs in polynomial time since $\mathcal{A}$ does. Whenever $\mathcal{A}$ found a collision in H, $\mathcal{A}_1$ found a collision in H_1. Also, $\mathcal{A}_2$ can find a collision in H_2. So we have

$$\Pr[\mathsf{Hash-coll}_{\mathcal{A}_1,\Pi_1} = 1] > \Pr[\mathsf{Hash-coll}_{\mathcal{A},\Pi} = 1] = \epsilon(n)$$

$$\Pr[\mathsf{Hash-coll}_{\mathcal{A}_2,\Pi_2} = 1] > \Pr[\mathsf{Hash-coll}_{\mathcal{A},\Pi} = 1] = \epsilon(n)$$

So we have that both $\mathcal{A}_1$ and $\mathcal{A}_2$ are not collision-resistant.

In conclusion, if at least one of H_1 and H_2 are collision-resistant, H is collision-resistant.