Cryptography Homework 4

2024 Spring Semester

21 CST H3Art

Exercise 5.1

Define a toy hash function $h : (\mathbb{Z}_2)^7 \to (\mathbb{Z}_2)^4$ by the rule $h(x) = xA$ where all operations are modulo 2 and

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Find all preimages of $(0, 1, 0, 1)$.

> **Solution**:
>
> This is equivalent to solving the following system of equations:
>
> $$\begin{cases} (x_1 + x_2 + x_3 + x_4) \bmod 2 = 0 \\ (x_2 + x_3 + x_4 + x_5) \bmod 2 = 1 \\ (x_3 + x_4 + x_5 + x_6) \bmod 2 = 0 \\ (x_4 + x_5 + x_6 + x_7) \bmod 2 = 1 \end{cases}$$
>
> Finally, there are 8 solutions for $(x_1, \ldots, x_7)$: $(1, 1, 1, 1, 0, 0, 0)$, $(1, 1, 0, 0, 0, 0, 1)$, $(1, 0, 1, 0, 0, 1, 0)$, $(1, 0, 0, 1, 0, 1, 1)$, $(0, 1, 1, 0, 1, 0, 0)$, $(0, 1, 0, 1, 1, 0, 1)$, $(0, 0, 1, 1, 1, 1, 0)$, $(0, 0, 0, 0, 1, 1, 1)$.

Exercise 5.6

(This exercise is based on an example from the *Handbook of Applied Cryptography* by A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone.) Suppose g is a collision resistant hash function that takes an arbitrary bitstring as input and produces an n-bit message digest. Define a hash function h as follows:

$$h(x) = \begin{cases} 0 \,||\, x & \text{if } x \text{ is a bitstring of length } n \\ 1 \,||\, g(x) & \text{otherwise.} \end{cases}$$

(a) Prove that h is collision resistant.

(b) Prove that h is not preimage resistant. More precisely, show that preimages (for the function h) can easily be found for half of the possible message digests.

> **Solution**:
>
> (a) Suppose $h(x) = h(x')$ for some $x \neq x'$. We divide the proof into three cases:
>
> **First Case**: Assume $|x| = |x'| = n$. Then $h(x) = 0 \,||\, x$ and $h(x') = 0 \,||\, x'$. Since $h(x) = h(x')$, it implies $x = x'$, which leads to a contradiction.

Second Case: Assume $|x| = n$ and $|x'| \neq n$. Here, the first bit of $h(x)$ is 0 and the first bit of $h(x')$ is 1, therefore $h(x) \neq h(x')$.

Final Case: Assume $|x| \neq n$ and $|x'| \neq n$. In this scenario, $h(x) = 1 \,||\, g(x)$ and $h(x') = 1 \,||\, g(x')$. Given $h(x) = h(x')$, it follows that $g(x) = g(x')$. This indicates a collision in g, contradicting the assumption that g is collision-resistant.

(b) For the image y where $y = (0, y_2, y_3, \cdots, y_{n+1})$, let $x = (y_2, y_3, \cdots, y_{n+1})$, the length of x is n, so x is the preimage of y because $h(x) = 0 \,||\, x = y$. Therefore, we have 2^n possible n-bit strings, the preimages are easily found in half of the possible message digests, h is not preimage resistant.

Exercise 5.7

If we define a hash function (or compression function) h that will hash an n-bit binary string to an m-bit binary string, we can view h as a function from $\mathbb{Z}_{2^n}$ to $\mathbb{Z}_{2^m}$. It is tempting to define h using integer operations modulo 2^m. We show in this exercise that some simple constructions of this type are insecure and should therefore be avoided.

(a) Suppose that $n = m > 1$ and $h : \mathbb{Z}_{2^m} \to \mathbb{Z}_{2^m}$ is defined as

$$h(x) = x^2 + ax + b \bmod 2^m$$

Prove that it is (usually) easy to solve **Second Preimage** for any $x \in \mathbb{Z}_{2^m}$ without having to solve a quadratic equation.

HINT: Show that it is possible to find a linear function $g(x)$ such that $h(g(x)) = h(x)$ for all x. This solves **Second Preimage** for any x such that $g(x) \neq x$.

(b) Suppose that $n > m$ and $h : \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^m}$ is defined to be a polynomial of degree d:

$$h(x) = \sum_{i=0}^{d} a_i x^i \bmod 2^m$$

where $a_i \in \mathbb{Z}$ for $0 \leq i \leq d$. Prove that it is easy to solve **Second Preimage** for any $x \in \mathbb{Z}_{2^n}$ without having to solve a polynomial equation.

HINT: Make use of the fact that $h(x)$ is defined using reduction modulo 2^m, but the domain of h is $\mathbb{Z}_{2^n}$, where $n > m$.

Solution:

(a) Suppose that a is even; then $a2^{m-1} \equiv 0 (\bmod\ 2^m)$. Also, $2^{2m-2} \equiv 0 (\bmod\ 2^m)$ because $m \geq 2$. Define $x' = x + 2^{m-1} \bmod 2^m$; then

$$\begin{aligned} h(x') &= (x + 2^{m-1})^2 + a(x + 2^{m-1}) + b \bmod 2^m \\ &= x^2 + 2^m x + 2^{2m-2} + ax + a2^{m-1} + b \bmod 2^m \\ &= x^2 + ax + b \bmod 2^m \\ &= h(x) \end{aligned}$$

Now suppose that a is odd. Define $x' = -x - a \bmod 2^m$; note that $x' \neq x$ because $2x + a$ is odd. Now, we have that

$$\begin{aligned} h(x') &= (-x - a)(-x) + b \bmod 2^m \\ &= (x + a)x + b \bmod 2^m \\ &= h(x) \end{aligned}$$

Therefore, given any x, we can find $x' \neq x$ such that $h(x') = h(x)$.

(b) Define $x' = x + 2^m \bmod 2^n$. We can find that x' is a valid solution to the second problem:

$$\begin{aligned}
h(x') &= \sum_{i=0}^{d} a_i x'^i \bmod 2^m \\
&= a_0 + a_1(x+2^m) + a_2(x+2^m)^2 + \cdots + a_d(x+2^m)^d \bmod 2^m \\
&= a_0 + a_1(x+2^m) + a_2(x^2 + 2^{m+1}x + 2^{2m}) + \cdots + a_d(x^d + \cdots + 2^{dm}) \bmod 2^m \\
&= ((a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d) + (a_1 2^m + a_2 2^{m+1} x + a_2 2^{2m} + \cdots + a_d 2^{dm})) \bmod 2^m \\
&= a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d \bmod 2^m \\
&= \sum_{i=0}^{d} a_i x^i \bmod 2^m \\
&= h(x)
\end{aligned}$$

Exercise 5.8

Suppose that $f : \{0,1\}^m \to \{0,1\}^m$ is a preimage resistant bijection. Define $h : \{0,1\}^{2m} \to \{0,1\}^m$ as follows. Given $x \in \{0,1\}^{2m}$, write

$$x = x' \mid\mid x''$$

where x', $x'' \in \{0,1\}^m$. Then define

$$h(x) = f(x' \oplus x'')$$

Prove that h is not second preimage resistant.

Solution:

We are given $x = x' \mid\mid x''$. Let $x_0 \in \{0,1\}^m$, $x_0 \neq \{0\}^m$.

Define $x_1' = x' \oplus x_0$, $x_1'' = x'' \oplus x_0$ and $x_1 = x_1' \mid\mid x_1''$.

Then $x \neq x_1$, we have

$$\begin{aligned}
h(x_1) &= f(x_1' \oplus x_1'') \\
&= f(x' \oplus x_0 \oplus x'' \oplus x_0) \\
&= f(x' \oplus x'' \oplus x_0 \oplus x_0) \\
&= f(x' \oplus x'') \\
&= h(x)
\end{aligned}$$

Thus, h is not second preimage resistant.

Exercise 5.12

Suppose $h_1 : \{0,1\}^{2m} \to \{0,1\}^m$ is a collision resistant hash function.

(a) Define $h_2 : \{0,1\}^{4m} \to \{0,1\}^m$ as follows:

1. Write $x \in \{0,1\}^{4m}$ as $x = x_1 \mid\mid x_2$, where $x_1, x_2 \in \{0,1\}^{2m}$.
2. Define $h_2(x) = h_1(h_1(x_1) \mid\mid h_1(x_2))$.

Prove that h_2 is collision resistant (i.e., given a collision for h_2, show how to find a collision for h_1).

(b) For an integer $i \geq 2$, define a hash function $h_i : \{0,1\}^{2^i m} \to \{0,1\}^m$ recursively from h_{i-1}, as follows:

1. Write $x \in \{0,1\}^{2^i m}$ as $x = x_1 \mid\mid x_2$, where $x_1, x_2 \in \{0,1\}^{2^{i-1} m}$.
2. Define $h_i(x) = h_1(h_{i-1}(x_1) \mid\mid h_{i-1}(x_2))$.

Prove that h_i is collision resistant.

> **Solution**:
>
> (a) Suppose that we have found a collision for h_2, say $h_2(x) = h_2(x')$ where $x \neq x'$. Denote $x = x_1 \| x_2$ and $x' = x'_1 \| x'_2$. First, suppose that $h_1(x_1) \neq h_1(x'_1)$. Then
>
> $$h_1(x_1) \| h_1(x_2) \neq h_1(x'_1) \| h_1(x'_2)$$
>
> and
>
> $$h_1(h_1(x_1) \| h_1(x_2)) = h_1(h_1(x'_1) \| h_1(x'_2))$$
>
> Therefore, we have found a collision for h_1.
>
> If $h_1(x_2) \neq h_1(x'_2)$, then by a similar argument, we have a collision for h_1.
>
> Therefore, we can assume that $h_1(x_1) = h_1(x'_1)$ and $h_1(x_2) = h_1(x'_2)$.
>
> Because $x \neq x'$, it follows that $(x_1, x_2) \neq (x'_1, x'_2)$. Therefore, $x_1 \neq x'_1$ or $x_2 \neq x'_2$. In either of these two cases, we have a collision for h_1.
>
> We conclude that given a collision for h_2, we can always find a collision for h_1.
>
> (b) Suppose that we have found a collision for h_i, say $h_i(x) = h_i(x')$ where $x \neq x'$. Denote $x = x_1 \| x_2$ and $x' = x'_1 \| x'_2$.
>
> First, suppose that $h_{i-1}(x_1) \neq h_{i-1}(x'_1)$. Then
>
> $$h_{i-1}(x_1) \| h_{i-1}(x_2) \neq h_{i-1}(x'_1) \| h_{i-1}(x'_2)$$
>
> and
>
> $$h_1(h_{i-1}(x_1) \| h_{i-1}(x_2)) = h_1(h_{i-1}(x'_1) \| h_{i-1}(x'_2))$$
>
> Therefore, we have found a collision for h_1.
>
> If $h_{i-1}(x_2) \neq h_{i-1}(x'_2)$, then by a similar argument, we have a collision for h_1.
>
> Therefore, we can assume that $h_{i-1}(x_1) = h_{i-1}(x'_1)$ and $h_{i-1}(x_2) = h_{i-1}(x'_2)$. Because $x \neq x'$, it follows that $(x_1, x_2) \neq (x'_1, x'_2)$. Therefore, $x_1 \neq x'_1$ or $x_2 \neq x'_2$. In either of these two cases, we have a collision for h_{i-1}.
>
> We conclude that given a collision for h_i, we can always find a collision for at least one of h_1 or h_{i-1}.

Exercise 5.13

In this exercise, we consider a simplified version of the Merkle-Damgård construction. Suppose

$$\mathbf{compress} : \{0,1\}^{m+t} \to \{0,1\}^m$$

where $t \geq 1$, and suppose that

$$x = x_1 \mid\mid x_2 \mid\mid \cdots \mid\mid x_k$$

where

$$|x_1| = |x_2| = \cdots = |x_k| = t$$

We study the following iterated hash function:

> $\textbf{Algorithm}: \text{SIMPLIFIED MERKLE-DAMGÅRD}(x, k, t)$
>
> $\textbf{external compress}$
> $z_1 \leftarrow 0^m \mid\mid x_1$
> $g_1 \leftarrow \textbf{compress}(z_1)$
> $\textbf{for } i \leftarrow 1 \textbf{ to } k-1$
> $\quad \textbf{do} \begin{cases} z_{i+1} \leftarrow g_i \mid\mid x_{i+1} \\ g_{i+1} \leftarrow \textbf{compress}(z_{i+1}) \end{cases}$
> $h(x) \leftarrow g_k$
> $\textbf{return } (h(x))$

Suppose that $\textbf{compress}$ is collision resistant, and suppose further that $\textbf{compress}$ is ***zero preimage resistant***, which means that it is hard to find $z \in \{0,1\}^{m+t}$ such that $\textbf{compress}(z) = 0^m$. Under these assumptions, prove that h is collision resistant.

> **Solution**:
>
> Suppose that $h(x) = h(x')$ where $x \neq x'$. We consider two cases:
>
> **First case**: If $|x| = |x'| = kt$ for some positive integer k, we have $g_k = g'_k$, but if $z_k \neq z'_k$, then we have a collision for $\textbf{compress}$, otherwise we assume that $z_k = z'_k$. This implies that $g_{k-1} = g'_{k-1}$ and $x_k = x'_k$.
>
> After that, we work backward of the above algorithm, finally either we find a collision for $\textbf{compress}$, or we have $x_i = x'_i$ for $i = k, k-1, \ldots, 1$. But then $x = x'$, a contradiction.
>
> **Second case**: If $|x| = kt$ and $|x'| = \ell t$, where k and ℓ are positive integers such that $\ell > k$. If $g_k = g'_\ell$. If $z_k \neq z'_\ell$, then we have a collision for $\textbf{compress}$ and we're done, so we assume that $z_k = z'_\ell$. This implies that $g_{k-1} = g'_{\ell-1}$ and $x_k = x'_\ell$.
>
> We work backward like first case, finally either we find a collision for $\textbf{compress}$, or we eventually reach the situation where $z_1 = z'_{\ell-k+1}$. Then $0^m = g'_{\ell-k}$ =$\textbf{compress}(z'_{\ell-k})$, so $\textbf{compress}$ is not ***zero preimage resistant***. Therefore we either find a collision or a zero preimage for $\textbf{compress}$ in this case.
>
> Through the above proof, we can know that the given condition is collision, then we can find the collision of the $\textbf{compress}$ function, because the $\textbf{compress}$ function is collision resistant, then our given condition is wrong, and h is collision resistant.

Exercise 5.14

Message authentication codes are often constructed using block ciphers in CBC mode. Here we consider the construction of a message authentication code using a block cipher in CFB mode. Given a sequence of plaintext blocks, $x_1, \ldots, x_n$, suppose we define the initialization vector IV to be x_1. Then encrypt the sequence $x_2, \ldots, x_n$ using key K in CFB mode, obtaining the ciphertext sequence $y_1, \ldots, y_{n-1}$ (note that there are only $n-1$ ciphertext blocks). Finally, define the MAC to be $e_K(y_{n-1})$. Prove that this MAC actually turns out to be identical to CBC-MAC, as presented in Section 5.5.2.

> **Solution**:
>
> Using CFB mode, we obtain the encryption procedure as follows:

$$\begin{aligned}
\text{IV} &= x_1 \\
y_1 &= e_K(x_1) \oplus x_2 \\
y_2 &= e_K(y_1) \oplus x_3 \\
y_3 &= e_K(y_2) \oplus x_4 \\
&\vdots \\
y_{n-1} &= e_K(y_{n-2}) \oplus x_n \\
\text{MAC} &= e_K(y_{n-1})
\end{aligned}$$

Using CBC mode with $\text{IV} = 0\,0 \cdots 0$, we obtain the following:

$$\begin{aligned}
\text{IV} &= 0\,0 \cdots 0 \\
y_1' &= e_K(x_1) \\
y_2' &= e_K(y_1' \oplus x_2) \\
y_3' &= e_K(y_2' \oplus x_3) \\
&\vdots \\
y_n' &= e_K(y_{n-1}' \oplus x_n) \\
\text{MAC}' &= y_n'
\end{aligned}$$

Next we need to prove $y_i = y_i' \oplus x_{i+1}, 1 \leq i \leq n-1$ by induction:

Base case($i = 1$):

Since $\text{IV} = x_1$ in CFB mode, $y_1 = e_K(x_1) \oplus x_2$, and $y_1' = e_K(x_1)$ in CBC mode, we can find $y_1 = y_1' \oplus x_2$.

Induction($i > 1$):

Suppose $y_i = y_i' \oplus x_{i+1}$ holds for $1 < i \leq n-2$, since $y_i = e_K(y_{i-1}) \oplus x_{i+1}$, therefore

$$\begin{aligned}
y_{i+1} &= e_K(y_i) \oplus x_{i+2} \\
&= e_K(y_i' \oplus x_{i+1}) \oplus x_{i+2}
\end{aligned}$$

and

$$y_{i+1}' = e_K(y_i' \oplus x_{i+1})$$

so $y_{i+1} = y_{i+1}' \oplus x_{i+2}$.

Finally, we have

$$\begin{aligned}
\text{MAC} &= e_K(y_{n-1}) \\
&= e_K(y_{n-1}' \oplus x_n) \\
&= y_n' \\
&= \text{MAC}'
\end{aligned}$$

Therefore the same MAC is produced by both methods.

Exercise 5.15

Suppose that $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is a cryptosystem with $\mathcal{P} = \mathcal{C} = \{0,1\}^m$. Let $n \geq 2$ be a fixed integer, and define a hash family $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$, where $\mathcal{X} = (\{0,1\}^m)^n$ and $Y = \{0,1\}^m$, as follows:

$$h_K(x_1, \ldots, x_n) = e_K(x_1) \oplus \cdots \oplus e_K(x_n)$$

Suppose that $(x_1, \ldots, x_n)$ is an arbitrary message. Show how an adversary can then determine $h_K(x_1, \ldots, x_n)$ by using at most one oracle query. (This is called a ***selective forgery***, because a specific message is given to the adversary and the adversary is then required to find the tag for the given message.)

HINT: The proof is divided into three mutually exclusive cases as follows:

- **case 1**: In this case, we assume that not all of the x_i's are identical. Here, one oracle query suffices.
- **case 2**: In this case, we assume n is even and $x_1 = \cdots = x_n$. Here, no oracle queries are required.
- **case 3**: In this case, we assume $n \geq 3$ is odd and $x_1 = \cdots = x_n$. Here, one oracle query suffices.

Solution:

First, suppose that $x_i \neq x_j$ for some i and j. Define

$$x'_k = \begin{cases} x_k & \text{if } k \neq i, j \\ x_i & \text{if } k = j \\ x_j & \text{if } k = i \end{cases}$$

Request the MAC for $(x'_1, \ldots, x'_n)$, denoted as y_0. Then y_0 is a forged MAC for the original message $(x_1, \ldots, x_n)$.

Next, suppose that $x_1 = x_2 = \cdots = x_n$.

- If n is even, then $h_K(x_1, \ldots, x_1) = 0$ (i.e., we have a $(1, 0)$-forgery).
- If n is odd, select $x' \neq x_1$ and request the MAC for $(x', \ldots, x', x_1)$, denoted as y_0. Then y_0 is a forged MAC for the original message $(x_1, \ldots, x_1)$.