Cryptographic Hash Functions
Problem Set 1

Summer 2019

Prof. Stefan Lucks
<firstname>.<lastname>(at)uni-weimar.de

Bauhaus-Universität Weimar

April 18, 2019

Task 1: Division Method for Hashing

Inputs $x_i \in \mathbb{Z}_p$, let

$$H(x_1, \ldots, x_m) := \left(\sum_{i=1}^{m} 2^{m-i} \cdot x_i \right) \bmod p$$

For $m \in \{2, 3\}$:

$$H(x_1, x_2) := \left(2^1 \cdot x_1 + 2^0 \cdot x_2\right) \bmod p$$
$$H(x_1, x_2, x_3) := \left(2^2 \cdot x_2 + 2^1 \cdot x_1 + 2^0 \cdot x_2\right) \bmod p$$

Collision and Preimage Resistance

What We Use in The Lecture

Let $H : \mathcal{X} \to \mathcal{Y}$ be a hash function. Let $x \twoheadleftarrow \mathcal{X}$ denote uniformly random sampling of x from a set $\mathcal{X}$.

Definition 1 (Collision Resistance ($\textsc{Coll}$))

H is called **collision-resistant** iff it is it is **infeasible** to find any pair $x, x' \in \mathcal{X}$, $x \neq x'$, with $H(x) = H(x')$.

Definition 2 (Preimage Resistance ($\textsc{Pre}$))

H is called **preimage-resistant** if, given a **random** $y \twoheadleftarrow \mathcal{Y}$, it is **infeasible** to find any $x \in \mathcal{X}$ with $H(x) = y$.

Definition 3 (Second-Preimage Resistance ($\textsc{Sec}$))

H is called **preimage-resistant** if, given a **random** $x \twoheadleftarrow \mathcal{X}$, it is **infeasible** to find any $x' \in \mathcal{X}$, $x \neq x'$, with $H(x) = H(x')$.

Memorize this slide, but note that there exist more than those...

Further Notions

P. Rogaway, T. Shrimpton: Cryptographic Hash-Function Basics . . . , FSE 2004.

Let $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a family of keyed hash functions.

Definition 4 (Preimage Resistance (PRE))

H is called **preimage-resistant** if, given a **random** instance H_K for $K \twoheadleftarrow \mathcal{K}$, and a **random** $y \twoheadleftarrow \mathcal{Y}$, it is **infeasible** to find any $x \in \mathcal{X}$ with $H(x) = y$.

Definition 5 (Always-Preimage Resistance (APRE))

H is called **preimage-resistant** if, given a **fixed** instance H_K, and a **random** $y \twoheadleftarrow \mathcal{Y}$, it is **infeasible** to find any $x \in \mathcal{X}$ with $H(x) = y$.

Definition 6 (Everywhere-Preimage Resistance (EPRE))

H is called **everywhere-preimage-resistant** if, given a **random** instance H_K for $K \twoheadleftarrow \mathcal{K}$, and a **fixed** $y \in \mathcal{Y}$, it is **infeasible** to find any $x \in \mathcal{X}$ with $H(x) = y$.

Optimal Security

Let $H : \mathcal{X} \to \{0,1\}^n$ be a hash function. Let $\mathcal{A}$ be an adversary that asks at most q queries. Then, for H to be optimal resistant, the advantage of any such $\mathcal{A}$ should be upper bounded by

- Collision Resistance:

$$\mathbf{Adv}_H^{\textsc{Coll}}(\mathcal{A}) \leq \binom{q}{2} \cdot \frac{1}{2^n}.$$

- Preimage Resistance:

$$\mathbf{Adv}_H^{\textsc{Pre}}(\mathcal{A}) \leq \frac{q}{2^n}.$$

- Second-Preimage Resistance:

$$\mathbf{Adv}_H^{\textsc{Sec}}(\mathcal{A}) \leq \frac{q}{2^n}.$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$					
b) $H(\text{const}) \oplus x$					
c) $H(x) \,\|\, \overline{H(x)}$					
d) $H(\overline{x})$					
e) $H(x \oplus_* \text{const})$					
f) $H(x) \oplus H(y)$					

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$	✓	–	–	✓	✓
b) $H(\text{const}) \oplus x$					
c) $H(x) \,\|\|\, \overline{H(x)}$					
d) $H(\overline{x})$					
e) $H(x \oplus_* \text{const})$					
f) $H(x) \oplus H(y)$					

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$	✓	–	–	✓	✓
b) $H(\text{const}) \oplus x$	✓	–	–	✓	✓
c) $H(x) \,\|\|\, \overline{H(x)}$					
d) $H(\overline{x})$					
e) $H(x \oplus_* \text{const})$					
f) $H(x) \oplus H(y)$					

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$	✓	–	–	✓	✓
b) $H(\text{const}) \oplus x$	✓	–	–	✓	✓
c) $H(x) \,\|\|\, \overline{H(x)}$	○	○	○	○	○
d) $H(\overline{x})$					
e) $H(x \oplus_* \text{const})$					
f) $H(x) \oplus H(y)$					

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$	✓	–	–	✓	✓
b) $H(\text{const}) \oplus x$	✓	–	–	✓	✓
c) $H(x) \,\|\|\, \overline{H(x)}$	○	○	○	○	○
d) $H(\overline{x})$	✓	✓	✓	✓	✓
e) $H(x \oplus_* \text{const})$					
f) $H(x) \oplus H(y)$					

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$	✓	–	–	✓	✓
b) $H(\text{const}) \oplus x$	✓	–	–	✓	✓
c) $H(x) \,\|\|\, \overline{H(x)}$	○	○	○	○	○
d) $H(\overline{x})$	✓	✓	✓	✓	✓
e) $H(x \oplus_* \text{const})$	–	✓	–	✓	–
f) $H(x) \oplus H(y)$					

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 2: Collision and Preimage Resistance

Construction	Collisions	Preimages		Second Preimages	
		aPre	ePre	aSec	eSec
a) $H(x) \,\|\|\, x$	✓	–	–	✓	✓
b) $H(\text{const}) \oplus x$	✓	–	–	✓	✓
c) $H(x) \,\|\|\, \overline{H(x)}$	○	○	○	○	○
d) $H(\overline{x})$	✓	✓	✓	✓	✓
e) $H(x \oplus_* \text{const})$	–	✓	–	✓	–
f) $H(x) \oplus H(y)$	–	○	–	–	–

✓ = resistance provided, ○ = suboptimal but infeasible, – = no resistance;
$x \oplus y$ assumes $|x| = |y|$.

$$x \oplus_* y := \begin{cases} x \oplus y & \text{if } |x| = |y| \\ (x \,||\, 0^{|y|-|x|}) \oplus y & \text{if } |x| < |y| \\ (y \,||\, 0^{|x|-|y|}) \oplus x & \text{otherwise.} \end{cases}$$

Task 3: Product Hash

Let $p = 61$ and let H be a keyed hash function with $k_1, k_2 \in \mathbb{Z}_p$ for inputs $x_1, x_2 \in \mathbb{Z}_p$:

$$H_{k_1,k_2}(x_1, x_2) := (k_1 + x_1) \cdot (k_2 + x_2) \bmod p.$$

Recover the secret key (k_1, k_2) for the given message-hash pairs:

a) $H_{k_1,k_2}(20, 10) = 34$

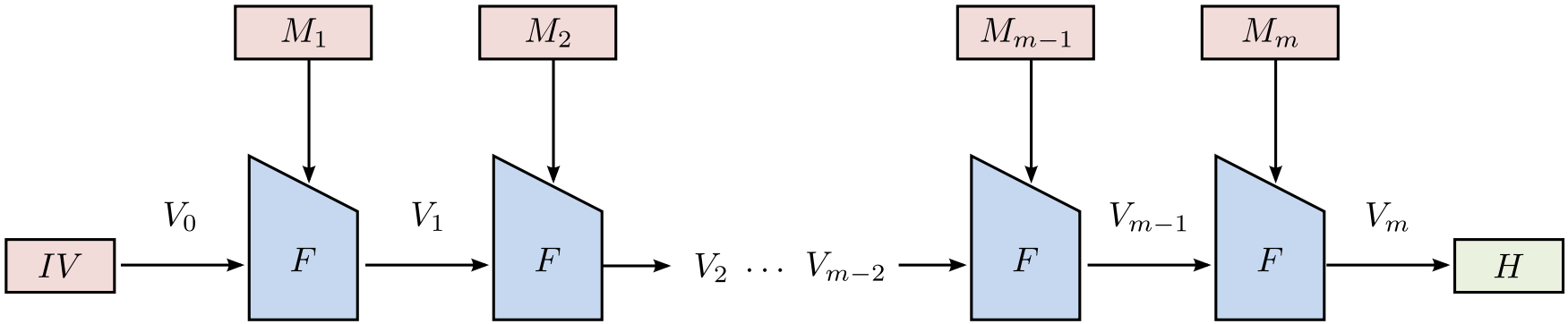
b) $H_{k_1,k_2}(6, 8) = 11$

c) $H_{k_1,k_2}(5, 3) = 49$

Look Ahead

Merkle-Damgård Paradigm:

- Given: Compression function $F : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^\ell$
- Goal: build a hash function $H : \{0,1\}^* \to \{0,1\}^\ell$
  - Append message length and pad M: $M' \leftarrow M \,||\, |M| \,||\, 10^*$
  - Split $M_1, \ldots, M_m \xleftarrow{n} M'$ into n-bit blocks
  - Iteratively compute $V_i \leftarrow F(V_{i-1}, M_i)$ for $i \in 1, \ldots, m$
  - $V_0 := IV$ is a public constant IV
  - $V_m := H(M)$ is the hash



Reductionist Proofs by Contradiction

Hypothesis:
A holds.

Task:
Show that X is impossible.

Reductionist Proofs by Contradiction

Hypothesis:
A holds.

Task:
Show that X is impossible.

$\Longrightarrow$

Antithesis:
X is possible.

Reduced Task:
Show that A does not hold in this case.

Reductionist Proofs by Contradiction

Hypothesis:
A holds.

Task:
Show that X is impossible.

$\Longrightarrow$

Antithesis:
X is possible.

Reduced Task:
Show that A does not hold in this case.

Example:

Hypothesis:
F is some secure primitive.

$\Longrightarrow$

Task:
Show that F' which uses F is also secure.

Reductionist Proofs by Contradiction

Hypothesis:
A holds.

Task:
Show that X is impossible.

$\Longrightarrow$

Antithesis:
X is possible.

Reduced Task:
Show that A does not hold in this case.

Example:

Hypothesis:
F is some secure primitive.

Task:
Show that F' which uses F is also secure.

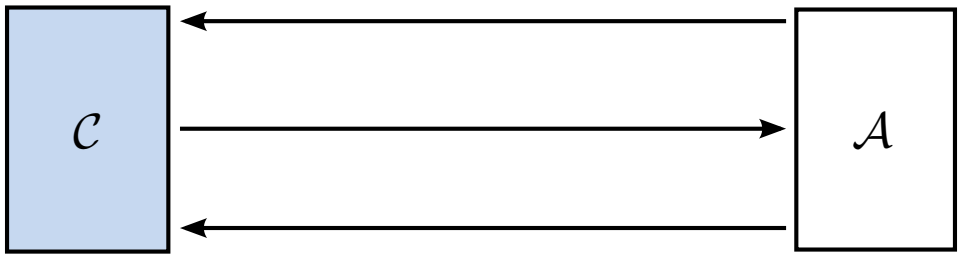
$\Longrightarrow$

Antithesis:
There is an efficient adversary $\mathcal{A}_{F'}$ on F'.

Reduced Task:
Show that then, there exists also an efficient adversary $\mathcal{A}_F$ on F.

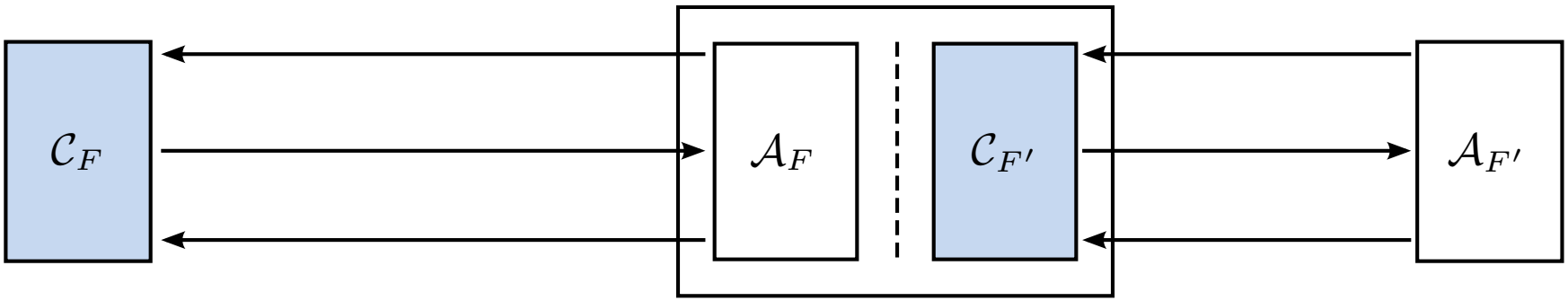
Security as Interaction between Challenger and Adversary

- Security: Modelled as interaction of adversary $\mathcal{A}$ with challenger $\mathcal{C}$
- Adversary $\mathcal{A}$:
  - Polynomial-time (efficient) Turing machine
  - Often: Outputs decision bit at end
- Challenger $\mathcal{C}$:
  - Provides $\mathcal{A}$ with oracle responces to $\mathcal{A}$'s queries
  - Black box to $\mathcal{A}$
  - Often has resources (internal function, secret key) under its control
  - Decides if the result of $\mathcal{A}$ is valid $=$ $\mathcal{A}$ wins



Reduction Proofs

- Security of F' is reduced to the security of F
- Goal: Adversary $\mathcal{A}$ on F
- Uses an adversary $\mathcal{A}_{F'}$ on F'
- $\mathcal{A}_{F'}$ acts as a challenger $\mathcal{C}_{F'}$ in the view of $\mathcal{A}_{F'}$



Questions?