# DD2448 Foundations of Cryptography
# Lecture 5

Douglas Wikström
KTH Royal Institute of Technology
dog@kth.se

March 25, 2020

# Second Pre-Image Resistance

**Definition.** A function $h : \{0,1\}^* \rightarrow \{0,1\}^*$ is said to be **second pre-image resistant** if for every polynomial time algorithm $A$ and a random $x$

$$\Pr[A(x) = x' \wedge x' \neq x \wedge f(x') = f(x)] < \epsilon(n)$$

for a negligible function $\epsilon$.

Note that $A$ is given not only the output of $f$, but also the **input** $x$, but it must find a **second** pre-image.

# Collision Resistance

**Definition.** Let $f = \{f_\alpha\}_\alpha$ be an ensemble of functions. The "function" $f$ is said to be **collision resistant** if for every polynomial time algorithm $A$ and randomly chosen $\alpha$

$$\Pr[A(\alpha) = (x, x') \wedge x \neq x' \wedge f_\alpha(x') = f_\alpha(x)] < \epsilon(n)$$

for a negligible function $\epsilon$.

# Collision Resistance

**Definition.** Let $f = \{f_\alpha\}_\alpha$ be an ensemble of functions. The "function" $f$ is said to be **collision resistant** if for every polynomial time algorithm $A$ and randomly chosen $\alpha$

$$\Pr[A(\alpha) = (x, x') \wedge x \neq x' \wedge f_\alpha(x') = f_\alpha(x)] < \epsilon(n)$$

for a negligible function $\epsilon$.

An algorithm that gets a small "advice string" for each security parameter can easily hardcode a collision for a fixed function $f$, which explains the random index $\alpha$.

# Relations for Compressing Hash Functions

- If a function is not second pre-image resistant, then it is not collision-resistant.

# Relations for Compressing Hash Functions

- If a function is not second pre-image resistant, then it is not collision-resistant.
  1. Pick random $x$.
  2. Request second pre-image $x' \neq x$ with $f(x') = f(x)$.
  3. Output $x'$ and $x$.

# Relations for Compressing Hash Functions

- If a function is not second pre-image resistant, then it is not collision-resistant.
  1. Pick random $x$.
  2. Request second pre-image $x' \neq x$ with $f(x') = f(x)$.
  3. Output $x'$ and $x$.

- If a function is not one-way, then it is not second pre-image resistant.

# Relations for Compressing Hash Functions

- If a function is not second pre-image resistant, then it is not collision-resistant.
  1. Pick random $x$.
  2. Request second pre-image $x' \neq x$ with $f(x') = f(x)$.
  3. Output $x'$ and $x$.

- If a function is not one-way, then it is not second pre-image resistant.
  1. Given random $x$, compute $y = f(x)$.
  2. Request pre-image $x'$ of $y$.
  3. Repeat until $x' \neq x$, and output $x'$.

# Random Oracles

# Random Oracle As Hash Function

A random oracle is simply a randomly chosen function with appropriate domain and range.

A random oracle is the **perfect** hash function. Every input is mapped **independently** and **uniformly** in the range.

Let us consider how a random oracle behaves with respect to our notions of security of hash functions.

# Pre-Image of Random Oracle

We assume with little loss that an adversary always "knows" if it has found a pre-image, i.e., it queries the random oracle on its output.

**Theorem.** Let $H : X \to Y$ be a randomly chosen function and let $x \in X$ be randomly chosen. Then for every algorithm $A$ making $q$ oracle queries

$$\Pr[A^{H(\cdot)}(H(x)) = x' \wedge H(x) = H(x')] \leq 1 - \left(1 - \frac{1}{|Y|}\right)^q \quad .$$

# Pre-Image of Random Oracle

We assume with little loss that an adversary always "knows" if it has found a pre-image, i.e., it queries the random oracle on its output.

**Theorem.** Let $H : X \to Y$ be a randomly chosen function and let $x \in X$ be randomly chosen. Then for every algorithm $A$ making $q$ oracle queries

$$\Pr[A^{H(\cdot)}(H(x)) = x' \wedge H(x) = H(x')] \leq 1 - \left(1 - \frac{1}{|Y|}\right)^q \enspace .$$

**Proof.** Each query $x'$ satisfies $H(x') \neq H(x)$ independently with probability $1 - \frac{1}{|Y|}$.

# Second Pre-Image of Random Oracle

We assume with little loss that an adversary always "knows" if it has found a second pre-image, i.e., it queries the random oracle on the input and its output.

**Theorem.** Let $H : X \to Y$ be a randomly chosen function and let $x \in X$ be randomly chosen. Then for every such algorithm $A$ making $q$ oracle queries

$$\Pr[A^{H(\cdot)}(x) = x' \wedge x \neq x' \wedge H(x) = H(x')] \leq 1 - \left(1 - \frac{1}{|Y|}\right)^{q-1} \quad .$$

# Second Pre-Image of Random Oracle

We assume with little loss that an adversary always "knows" if it has found a second pre-image, i.e., it queries the random oracle on the input and its output.

**Theorem.** Let $H : X \to Y$ be a randomly chosen function and let $x \in X$ be randomly chosen. Then for every such algorithm $A$ making $q$ oracle queries

$$\Pr[A^{H(\cdot)}(x) = x' \wedge x \neq x' \wedge H(x) = H(x')] \leq 1 - \left(1 - \frac{1}{|Y|}\right)^{q-1} \quad .$$

**Proof.** Same as pre-image case, except we must waste one query on the input value to get the target in $Y$.

# Collision Resistance of Random Oracles

We assume with little loss that an adversary always "knows" if it has found a collision, i.e., it queries the random oracle on its outputs.

**Theorem.** Let $H : X \to Y$ be a randomly chosen function. Then for every such algorithm $A$ making $q$ oracle queries

$$\begin{aligned}\Pr[A^{H(\cdot)} = (x, x') \wedge x \neq x' \wedge H(x) = H(x')] &\leq 1 - \prod_{i=1}^{q-1} \left(1 - \frac{i}{|Y|}\right) \\ &\leq \frac{q(q-1)}{2|Y|} \enspace .\end{aligned}$$

# Collision Resistance of Random Oracles

We assume with little loss that an adversary always "knows" if it has found a collision, i.e., it queries the random oracle on its outputs.

**Theorem.** Let $H : X \to Y$ be a randomly chosen function. Then for every such algorithm $A$ making $q$ oracle queries

$$\begin{aligned}\Pr[A^{H(\cdot)} = (x, x') \wedge x \neq x' \wedge H(x) = H(x')] &\leq 1 - \prod_{i=1}^{q-1}\left(1 - \frac{i}{|Y|}\right) \\ &\leq \frac{q(q-1)}{2|Y|} \enspace .\end{aligned}$$

**Proof.** $1 - \frac{i-1}{|Y|}$ bounds the probability that the $i$th query does not give a collision for any of the $i-1$ previous queries, conditioned on no previous collision.

# Iterated Hash Functions

# Merkle-Damgård (1/3)

Suppose that we are given a collision resistant hash function

$$f : \{0,1\}^{n+t} \rightarrow \{0,1\}^n \ .$$

How can we construct a collision resistant hash function

$$h : \{0,1\}^* \rightarrow \{0,1\}^n$$

mapping any length inputs?

# Merkle-Damgård (2/3)

**Construction.**

1. Let $x = (x_1, \ldots, x_k)$ with $|x_i| = t$ and $0 < |x_k| \leq t$.
2. Let $x_{k+1}$ be the total number of bits in $x$.
3. Pad $x_k$ with zeros until it has length $t$.
4. $y_0 = 0^n$, $y_i = f(y_{i-1}, x_i)$ for $i = 1, \ldots, k+1$.
5. Output $y_{k+1}$

Here the total number of bits is bounded by $2^t - 1$, but this can be relaxed.