ETH Zürich, D-INFK
Spring 2019

Prof. Ueli Maurer
Fabio Banfi
Daniel Jost
Jiamin Zhu

Cryptography Foundations

Exercise 7

7.1 The Merkle-Damgård Hash-Function Construction

Goal*: Learn about the Merkle-Damgård construction of a hash function from a compression function.*

Let $f\colon \{0,1\}^{m+n+1} \to \{0,1\}^n$ with $m, n \geq 1$. f is called a *compression function*. We want to use f to construct a hash function $h : \{0,1\}^* \to \{0,1\}^n$. The idea is to split a message $x \in \{0,1\}^*$ into blocks of length m and iteratively feed the concatenation of the current block and the output of the previous stage into the compression function. Hence we must first pad x to a multiple of m bits. So let

$$x \mapsto \hat{x} : \{0,1\}^* \to \{0,1\}^*$$

be a padding function sucht that $|\hat{x}| = mt$ for some $t \geq 1$ and write $\hat{x} = \hat{x}_1 \parallel \cdots \parallel \hat{x}_t$ with $\hat{x}_k \in \{0,1\}^m$. Then set

$$h_1 = f((0,\ldots,0) \parallel \hat{x}_1)$$

and iteratively for $2 \leq k \leq t$,

$$h_k = f(h_{k-1} \parallel 1 \parallel \hat{x}_k).$$

Now define the hash $h(x)$ of x to be the output h_t of the last compression stage, i.e., $h(x) = h_t$. Such constructions of a hash function from a compression function are called Merkle-Damgård constructions.

a) Assume that $m \geq 2$ and consider the straightforward padding $\hat{x} = x \parallel (0,\ldots,0)$ which just appends zero or more 0's to fill the last block. Find a collision of h with this padding. So give two different message $x \neq y$ such that $h(x) = h(y)$.

b) Now let the padding be given by $\hat{x} = x \parallel (0,\ldots,0) \parallel \langle d \rangle$ where we append $d \geq 0$ zeros such that $|x| + d$ is a multiple of m, and the number d of appended zeros written as an m-bit binary string. Show that if you have an algorithm that wins the collision-finding game for h, you can use it in order to win the collision-finding game for f.

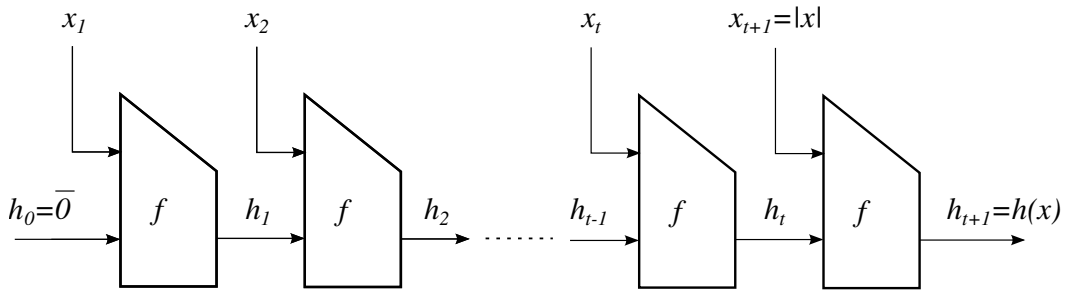

Figure 1: A graphical illustration of the Merkle-Damgård Construction with the padding of task 7.1 b).

7.2 Search Problems

Goal: *Understand the random experiment of an algorithm trying to solve a search problem.*

a) Describe the setting of a (probabilistic) algorithm A trying to solve a search problem $(\mathcal{X}, \mathcal{W}, Q, \mathsf{P}_X)$ as a random experiment. What are the random variables in this random experiment? How is the success probability of A defined?

b) Let A be an algorithm with success probability $\alpha \in (0, 1]$ for some search problem $(\mathcal{X}, \mathcal{W}, Q, \mathsf{P}_X)$ such that Q can (efficiently) be computed by an algorithm. Let the algorithm A' be defined as follows: Given an instance $x \in \mathcal{X}$, it first invokes A on input x to retrieve w. If $Q(x, w) = 1$, A' returns w. Otherwise, it invokes A again on input x to retrieve w' and returns w'. Find the best lower bound on the success probability of A'.

c) Suppose there is an algorithm A that solves the discrete logarithm problem with probability $\alpha \in (0, 1]$. Describe an algorithm A' (that uses A) such that the success probability of A' exceeds the one of A.

Hint: Given a group element g^x how can you obtain a "uniform but related" element g^y?

d) Why does the technique used in part **c)** not apply in general?

7.3 Properties of the Statistical Distance

Goal: *We show (1) that a probabilistic function (or algorithm) cannot increase the statistical distance of two random variables, and (2) that uniform distributions over two intervals of similar size are statistically close.*

a) Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets and let X and X' be random variables over $\mathcal{X}$. Further let A be a random variable over the set of functions $\mathcal{X} \to \mathcal{Y}$ such that A and X are independent and A and X' are independent. Show that

$$\delta\big(A(X), A(X')\big) \leq \delta\big(X, X'\big).$$

b) Let I be some set and $J \subseteq I$. Further let X be uniformly distributed over I and let Y be uniformly distributed over J. Show that

$$\delta(X, Y) = 1 - \frac{|J|}{|I|}.$$