CS195-7: Introduction to Cryptography DUE: September 17, 2002

Problem Set 2

Instructor: Anna Lysyanskaya

In lecture, we defined one-way permutations and gave an application for password authentication. In this problem set, we will define a weaker notion, namely that of one-way *functions* and explore applications.

A one-way function is a function that is easy to compute, but hard to invert. (So a one-way permutation is a one-way function that also happens to be a permutation.) More formally:

Definition: An efficiently computable function $f : \{0,1\}^* \mapsto \{0,1\}^*$ is a *one-way function* if for all probabilistic polynomial-time families of adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[x \leftarrow \{0,1\}^k; y = f(x); x' \leftarrow A_k(y) : f(x') = y] = \nu(k)$$

Definition: A one-way permutation is a one-way function that is a permutation.

Problem 1

The definition above captures the intuition that a one-way function should be easy to compute, but hard to invert. But there may be many ways to define the same concept.

Are hard-to-invert functions (defined below in Definition 1a) equivalent to one-way functions? What about hard-to-find-preimage functions (defined below in Definition 1b)?

Definition 1a: An efficiently computable function $f : \{0,1\}^* \mapsto \{0,1\}^*$ is a *hard-to-invert function* if for all probabilistic polynomial-time families of adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[x \leftarrow \{0,1\}^k; y = f(x); x' \leftarrow A_k(y) : x' = x] = \nu(k)$$

Definition 1b: An efficiently computable function $f : \{0,1\}^* \mapsto \{0,1\}^*$ is a *hard-to-find-preimage function* if for all probabilistic polynomial-time families of adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[y \leftarrow \{0,1\}^k; x \leftarrow A_k(y) : f(x) = y] = \nu(k)$$

Problem 2

Assume that f is a one-way function. Let "$\circ$" denote concatenation. If x is a binary string, let $|x|$ denote its length. For each of the functions below, either prove that it is a one-way function (by reduction that, in case g is not one-way, will give an algorithm that inverts f), or give an attack.

(a) A function g that ignores half of its input: $g(x_1 \circ x_2) = f(x_1)$, where $x_1 \circ x_2$ is a $2k$ or $2k-1$-bit input string, and x_1 denotes the first k bits of it.

(b) A function g that appends a string of zeroes to its output: $g(x) = f(x) \circ 0^{|f(x)|}$.

(c) A function g that is equivalent to f on all of its input strings x except those that end in $|x|/2$ zeroes:

$$g(x) = \begin{array}{ll} 0^{|x|} & \text{if } x = y \circ 0^{|x|/2} \\ f(x) & \text{otherwise} \end{array}$$

Problem 3

(This is what used to be Problem 2 on the last problem set. You may need to consult Dana Angluin's notes posted on the course webpage.)

Suppose p is a prime and g is a generator modulo p.

Experiment 1: Pick x at random in $\{1, \ldots, p-1\}$. Output g^x.

Experiment 2: Pick x, y at random in $\{1, \ldots, p-1\}$. Output g^{xy}.

Prove or disprove: Experiment 1 and Experiment 2 produce identically distributed outputs.