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

Solution 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

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

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

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

Solution:

Neither of the two definitions are equivalent to one-way functions.

For the first definition, consider the following function: $f(x) = 0^{|x|}$. This function satisfies Definition 1a: any k-bit string x satisfies that $f(x) = 0^k$, and so on input 0^k, it is hard to guess which input string it was. However, f is not a one-way function, since for any k-bit x, $x' = 0^k$ satisfies $f(x) = f(x')$.

For the second definition, consider function f such that $f(x) = x \circ 0^{|x|}$. This is not a one-way function since x can easily be computed from $f(x)$. (And for the same reason, it is not a hard-to-invert function either.) Yet, if y is a k-bit string chosen at random, most likely no x exists that satisfies $f(x) = y$: it will exist only if k is even, and the last $k/2$ bits of y are all zeroes, which happens with probability $2^{-k/2}$ for a randomly chosen y.

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.

Solution:

Suppose that an adversary $\{A_k\}$ inverts g with non-negligible probability. Then let us construct an algorithm $\{B_k\}$ that will invert f. On input $y = f(x)$ (where x is k bits long), our algorithm B_k must compute some x' such that $f(x') = y$.

B_k works as follows: on input y, run A_k. With non-negligible probability, A_k outputs some x_1' and x_2' such that $y = g(x_1' \circ x_2') = f(x_1')$. Our algorithm B_k will then simply output the value x_1'.

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

Solution:

Suppose that an adversary $\{A_k\}$ inverts g with non-negligible probability. Then let us construct an algorithm $\{B_k\}$ that will invert f. On input $y = f(x)$ (where x is k bits long), our algorithm B_k must compute some x' such that $f(x') = y$.

B_k works as follows: run A_k on input $y \circ 0^{|y|}$. With non-negligible probability, A_k outputs some x' such that $y \circ 0^{|y|} = g(x') = f(x') \circ 0^{|f(x')|}$. Our algorithm B_k will then simply output the value 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:

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

Solution:

This is a one-way function, by the following reduction: suppose we have an algorithm $\{A_k\}$ that computes $g^{-1}(z)$, where $z = g(x)$ for a randomly chosen x, non-negligibly often, that is to say with probability $\epsilon(|x|)$.

Written in our notation:

$$\Pr[x \leftarrow \{0,1\}^k; z = g(x); x' \leftarrow A_k(z) : g(x') = z] = \epsilon(|x|)$$

Let us try to use A_k to invert $f(x)$. Note that there are two cases when A succeeds in inverting g: the case that A succeeds and x does not end in $|x|/2$ zeroes, and the case when A succeeds and x does end in that many zeroes. Note that the probability of the second case is at most $2^{-|x|/2}$, because x is chosen uniformly at random to begin with.

Therefore:

$$\Pr[x \leftarrow \{0,1\}^k; z = g(x); x' \leftarrow A_k(z) : g(x') = z \wedge x \neq y \circ 0^{|x|}] \leq \epsilon(k) - 2^{-k/2}$$

But whenever x does not end in $|x|/2$ zeroes, $f(x) = g(x)$, and so

$$\Pr[x \leftarrow \{0,1\}^k; z = f(x); x' \leftarrow A_k(z) : f(x') = z] \leq \epsilon(k) - 2^{-k/2}$$

and $\epsilon(k) - 2^{-k/2}$ is non-negligible, because ϵ is non-negligible while $2^{-k/2}$ is negligible.

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.

Solution:

The two experiments do not produce the same outcome. As a counter-example, consider the groups $\mathbb{Z}_p^*$.