CS195-7: Introduction to Cryptography	DUE: October 10, 2002
Midterm Exam	
Instructor: Anna Lysyanskaya	

Please note that you are not allowed to collaborate with others on this exam.

Problem 1: One-way functions and permutations

Let $f : \{0,1\}^* \mapsto \{0,1\}^*$ be a one-way function. Let $p : \{0,1\}^* \mapsto \{0,1\}^*$ be a one-way permutation.

For each of the suggested implications below, prove or disprove that they are valid. That is to say, if an implication is valid, give a reduction. If it is not valid, give an example of a one-way function f and a one-way permutation p for which the implication is false. You may assume existence of one-way functions permutations.

Problem 2 from Problem Set 2 may serve as a helpful hint for a couple of these problems.

Example. Does it follow that $f(x)$ is a permutation?

Solution. It does not. Let $f'(x)$ be a one-way function. Let $f(x) = g(x)$ where $g(x)$ is as defined in Problem 2a of problem set 2. Then $f(x)$ is a one-way function (that's what is shown in that problem) but it cannot be a permutation because it ignores half of its input bits.

(a) Does it follow that $g(x) = f(f(x))$ is a one-way function?

(b) Does it follow that $g(x) = p(p(x))$ is a one-way permutation?

(c) Does it follow that $g(x) = f(x) \circ p(x)$ is a one-way function? (Recall that $\circ$ denotes concatenation.)

(d) Does it follow that, on input $p(x)$, one can efficiently compute $f(x)$?

Problem 2: The Blum-Rabin trapdoor permutation

Recall the definition of a family of trapdoor permutations. A trapdoor permutation family consists of algorithms $(G, M_{PK}, f_{PK}, f^{-1}_{PK})$. G generates a member of the family, that is to say, a public key PK that allows to efficiently evaluate the permutation f_{PK}, and the secret key SK that allows to efficiently invert f_{PK}. M_{PK} is the algorithm that efficiently samples the domain of the permutation f_{PK}.

For example, in RSA, the procedure G generates the modulus $n = pq$ and the exponent e, and sets $PK = (n, e)$ and $SK = d$, where $de \equiv 1 \bmod \phi(n)$. Furthermore, $M_{(n,e)} = \mathbb{Z}_n^*$, $f_{(n,e)}(x) = x^e \bmod n$, and $f^{-1}_{(n,e)}(y) = y^d \bmod n$.

$(G, M_{PK}, f_{PK}, f^{-1}_{PK})$ constitute a trapdoor permutation if f_{PK} is hard to invert. More formally, for all probabilistic polynomial-time adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[(PK, SK) \leftarrow G(1^k); y \leftarrow M_{PK}; x \leftarrow A_k(y) \ : \ f_{PK}(x) = y] = \nu(k)$$

Consider the following collection of algorithms:

Key generation The procedure $G(1^k)$ generates two k-bit primes, p and q, such that $p \equiv q \equiv 3 \bmod 4$. It outputs $PK = n = pq$, and $SK = (p, q)$. (Such a modulus n is called a *Blum integer*.)

Domain The domain M_n of the permutation f_n consists of all the quadratic residues modulo n. More formally,

$$M_n = \{x \mid x \in \mathbb{Z}_n^* \wedge \exists u \text{ such that } x \equiv u^2 \bmod n\}$$

To sample from the domain, pick $u \leftarrow \mathbb{Z}_n^*$, and output $x = u^2 \bmod n$.

Computing the function The permutation f_n is squaring: $f_n(x) = x^2 \bmod n$.

Inverting the function To compute $f_n^{-1}(y)$, one must compute the value $x \in M_n$ such that $x^2 = y \bmod n$.

In this problem, you will prove that the algorithms given above constitute a family of trapdoor permutations.

(a) Show that f_n is a permutation. (Hint: work modulo p and q first, and then combine using the Chinese remainder theorem.)

(b) Suppose that $p = 4m + 3$ is a prime and that a is a quadratic residue modulo p. Prove that a^{m+1} is a square root of a modulo p.

(c) Devise an efficient algorithm that, on input (p, q, y), computes $x = f_{pq}^{-1}(y)$, i.e., x such that $x^2 = y \bmod n$, where $n = pq$ is a Blum integer.

(d) Devise an efficient algorithm that, on input (n, a, b), where $a, b \in \mathbb{Z}_n^*$, $a \not\equiv \pm b \bmod n$, and $a^2 \equiv b^2 \bmod n$, outputs a non-trivial divisor of n.

(e) Let us assume that factoring Blum integers is infeasible. More precisely, assume that for all probabilistic polynomial-time adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[(n, (p, q)) \leftarrow G(1^k); p \leftarrow A_k(n) \ : \ p \mid n \wedge 1 < p < n] = \nu(k)$$

Show that under this assumption, it is infeasible to invert f_n. More precisely, show that for all probabilistic polynomial-time adversaries $\{A_k\}$, there exists a negligible funtion $\nu'(k)$ such that

$$\Pr[(n, (p, q)) \leftarrow G(1^k); y \leftarrow M_n; x \leftarrow A_k(n) \ : \ x^2 = y \bmod n] = \nu'(k)$$

(This fact is due to Michael Rabin.)