CS195-7: Introduction to Cryptography DUE: October 1, 2002

Problem Set 3-4

Instructor: Anna Lysyanskaya

Problem 1

In this problem, we show that if for a given RSA public key (n, e), the RSA function $f_{n,e}(x) = x^e \bmod n$ can be inverted on an ϵ fraction of $\mathbb{Z}_n^*$ in time t, then $f_{n,e}^{-1}(y)$ can be computed for any y in expected time t/ϵ.

(a) Show that for all RSA public keys (n, e), for all values $y \in \mathbb{Z}_n^*$, the following experiments produce identically distributed outcomes (recall that $x \leftarrow X$ notation means that x was chosen uniformly at random from set X):

Experiment 1: Pick $r \leftarrow \mathbb{Z}_n^*$, and output r.
Experiment 2: Pick $r \leftarrow \mathbb{Z}_n^*$, and output r^e.
Experiment 3: Pick $r \leftarrow \mathbb{Z}_n^*$, and output $r^e y$.

(b) Let (n, e) be fixed. Suppose that algorithm A has the following property: there exists a set $Y \subseteq \mathbb{Z}_n^*$, $|Y| = \phi(n)\epsilon$, such that for all $y \in Y$, $A(y)^e = y$, i.e., A inverts RSA for an ϵ fraction of $\mathbb{Z}_n^*$. Suppose that A's running time is t steps. (And assume that it always halts after t steps, even if its input is $y \notin Y$.) Show that

$$\Pr[r \leftarrow \mathbb{Z}_n^*; v = A(r) \ : \ v^e = r] \geq \epsilon$$

.

(c) Assume algorithm A as descibed in part (b). Give an algorithm that runs in expected time t/ϵ (up to a multiplicative factor that is independent on (n, e)) and inverts $f_{n,e}$ for all $y \in \mathbb{Z}_n^*$.

Problem 2

Suppose that Alice uses the same password x to log into three different machines A, B, and C. Each machine uses RSA with exponent 3 for password authentication. That is to say, each machine has generated its own RSA modulus, but happens to use $e = 3$. By n_X let us denote the modulus of machine $X = A, B, C$. The value $y_X = x^3 \bmod n_X$ is stored in a world-readable file at machines $X = A, B, C$. Give an algorithm that computes Alice's password x using the values n_A, n_B, n_C and y_A, y_B, y_C.

Problem 3

Recall the GM definition of security for one bit for public-key cryptosystems that you saw in lecture:

Definition. A cryptosystem (G, E, D) is GM-secure for one bit if: (1) it is a faithful cryptosystem where decryption retrieves the encrypted message for all choices of the public and secret keys; and (2) it is secure: for all probabilistic polynomial-time families of adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[(PK, SK) \leftarrow G(1^k); b \leftarrow \{0,1\}; c \leftarrow E(PK, b); b' \leftarrow A_k(PK, c) \ : \ b = b'] \leq 1/2 + \nu(k)$$

Note that in the above definition, the bit b is chosen at random, and then encrypted, and then the adversary cannot tell what it was better than by random guessing.

(a) Do we get an equivalent definition of a secure cryptosystem if we replace condition (2) of the definition above by the following:

For all probabilistic polynomial-time families of adversaries $\{A_k\}$, for $b = 0, 1$, there exists a negligible function $\nu(k)$ such that

$$\Pr[(PK, SK) \leftarrow G(1^k); c \leftarrow E(PK, b); b' \leftarrow A_k(PK, c) \ : \ b = b'] \leq 1/2 + \nu(k)$$

(b) (Extra credit — do this problem last!) Let the notation $b \leftarrow^q \{0,1\}$ denote that b is a *biased* bit: it is 0 with probability q, and 1 with probability $1 - q$, for $1/2 \leq q < 1$.

For what values of q do we get an equivalent definition of a secure cryptosystem if we replace condition (2) of the definition above by the following requirement:

For all probabilistic polynomial-time families of adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[(PK, SK) \leftarrow G(1^k); b \leftarrow^q \{0,1\}; c \leftarrow E(PK, b); b' \leftarrow A_k(PK, c) \ : \ b = b'] \leq q + \nu(k)$$