**CS195-7: Introduction to Cryptography** DUE: December 2, 2002

Problem Set 8

*Instructor: Anna Lysyanskaya*

## Problem 1: Insecure Signatures

All of the signature schemes presented in this problem are broken. A signature scheme can be broken in a number of ways: it can be shown to be existentially forgeable, it can be target-message forgeable, or its secret key can be completely recoverable. In order to break a signature scheme, different flavors of attacks can be launched: a public-key only attack, or an interactive attack where the adversary asks for signatures on messages of his choice.

In this problem, you are asked to break each in the strongest sense possible, using the weakest attack you can.

On the other hand, each of these signature schemes is secure in a weak sense: namely, on input a public key and a message, it is infeasible to compute the signature. Prove this weak security property for each signature scheme.

**(a)** Suppose a family of trapdoor permutations $\{p_{PK}\}$ over $\{0,1\}^k$ is given. The message space is $\{0,1\}^k$. A public key is the public key $PK$ of a member of the TDP; the secret key is the correspoding $SK$. The signature on message $m$ is the value $x$ such that $p_{PK}(x) = m$.

**(b)** Assume that factoring is hard. The public key of this signature scheme consists of a Blum integer $n = pq$. (Recall that Blum integers have the property that $-1$ is a quadratic non-residue with Jacobi symbol 1). The secret key is $n$'s factorization. The message space is $J_1(n)$, i.e. the set of elements of $\mathbb{Z}_n^*$ that have Jacobi symbol 1. A signature $s$ on message $m \in J_1(n)$ is computed as follows: if $m$ is a qudratic residue, then $s$ is some arbitrary square root of $m$. Otherwise, $s$ is some arbitrary square root of $-m$.

**(c)** Let $(G, \mathit{Sign}, \mathit{Verify})$ be the Lamport one-time secure signature. What happens if we use it more than once? (Recall the Lamport signature: Let $f$ be a one-way function. A secret key consists of the values $\{(x_i^0, x_i^1) \,:\, 1 \leq i \leq L\}$, where $L$ is the length of the messages in the message space, and each $x_i^b$ is chosen uniformly at random from $\{0,1\}^k$, $k$ being the security parameter. The corresponding public key consists of the values $\{(f(x_i^0), f(x_i^1)) \,:\, 1 \leq i \leq L\}$, and the signature on message $m = m_1 \circ m_2 \circ \ldots \circ m_L$ consists of the values $\{x_i^{m_i} \,:\, 1 \leq i \leq L\}$.)

## Problem 2: Collision-resistant hash functions

Let $\{G, h_K\}$ be a family of collision-resistant hash functions, where $G$ is the key generation algorithm that generates a key $K$, and $h_K \,:\, D_K \mapsto \{0,1\}^*$ is the function indexed by key $K$, with domain $D_K$. Recall that a function is collision-resistant if the following are satisfied:

**Non-triviality** : for all keys $K \in G$, for any input $x$, $|x| > k$, $h_K(x)$ is always shorter than $x$.

**Collision-resistance** : for all PPT adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[K \leftarrow G(1^k); (x_1, x_2) \leftarrow A_k(K) : h_K(x_1) = h_K(x_2) \wedge x_1, x_2 \in D_K] = \nu(k)$$

(recall that $D_K$ is the domain of the function $h_K$.)

**(a)** Let $\{G, h_K\}$ be a family of collision-resistant hash functions, where $D_K$, the domain of the function, consists of all strings of length up to $L(k)$, and $L(k) > k$ is a polynomial. Let $h'_K(x) = h_K(h_K(x))$. Is $\{G, h'_K\}$ a family of collision-resistant hash functions?

**(b)** Suppose you have a collision-resistant hash function that reduces its input by a little bit, for example from $k+1$ bits to $k$ bits. Your task is to design a hash function that reduces an $L > k$ bit string into a $k$ bit string, in such a way that it is hard to find collisions. More precisely, let $\{G, h_K\}$ be a family of collision-resistant hash functions, $h_K : \{0,1\}^{k+1} \mapsto \{0,1\}^k$. Let $L(k) > k$ be any given polynomial. Design a collision-resistant hash function family $\{G, h^L_K\}$ such that $h^L_K : \{0,1\}^{L(k)} \mapsto \{0,1\}^k$.

**(c)** Suppose you have a collision-resistant hash function for fixed-length inputs (for example, as in part (b)): $h_K : \{0,1\}^{L(k)} \mapsto \{0,1\}^k$. But we need to be able to hash strings of arbitrary length, so you need to construct a variable-length-input collision-resistant hash function. (Of course, you are safe to assume that your input string will never be longer than $2^k$, since at this point, the running time is no longer polynomial in the security parameter, and so all bets are off as far as security is concerned.) Let $h'_K(x)$ be as follows: $h'_K(x) = h^{|x|}_K(x)$, where $h^L_K(x)$ is your construction for part (b). Either explain why this will not result in a collision-resistant family (i.e., explain how to find collisions), or show that your construction for part (b) works in this context as well.

**(d)** Assuming that $\{G, h_K\}$ is a collision-resistant hash function family as in (b), give a construction of a secure variable-length-input collision-resistant hash function. You may use the fact that it is safe to assume that $2^k$ is an upper bound on the length of the input string $x$. (You do not need to do this if you construction in part (c) is already secure.)