Homework 4

Total of 130 points

The following two questions show that a PRF does *not* necessarily yield a cryptographic hash function that can be used as a proof of work or provide collision resistance.

1. (15 points) Show (under the PRF conjecture) that there exists a PRF $\{f_k\}$ mapping n bits to n bits and an efficient algorithm A such that $A(k) = x$ such that $f_k(x) = 0^\ell$.
2. (15 points) Show (under the PRF conjecture) that there exists a PRF $\{f_k\}$ mapping n bits to n bits and an efficient algorithm A such that $A(k) = (x, x')$ such that $f_k(x) = f_k(x')$.

3. (20 points) This question studies the security implications of including a unique "salt" value per user when hashing passwords. Suppose that H is a random oracle, and an adversary is given a database $(y_1, \ldots, y_N)$ of the hashes of passwords of N users, with each user choosing their password at random from a dictionary D. Compute (up to an order of magnitude) the expected number of queries (as a function of N and $|D|$) an adversary needs to make to recover the passwords for all users in the case (a) that the i^{th} entry in the database is simply $H(p_i)$ where p_i is the password of the i^{th} user and the case (b) that the i^{th} entry is $s_i \| H(s_i \| p_i)$ where s_i is a "salt" value chosen at random in $\{0,1\}^n$.

4. (30 points) Consider the following construction (S, V) for a message authentication code: given some hash function collection $\{h_k\}$, to sign the message m the signing algorithm $S_k(m)$ outputs a `C` program P that on input k, m outputs the constant $y = h_k(m)$. The verification algorithm V on key k and pair (m, P) outputs 1 iff $P(k, m) = h_k(m)$.

   a. (15 points) Prove that (S, V) is secure in the random oracle model.

That is, prove that with high probability if H is a random function and the function $h_k(m)$ is defined as $H(k\|m)$ then there is no adversary that succeeds in a chosen message attack against (S, V).

b. (15 points) Prove that (S, V) is *insecure* no matter *what* hash function collection $\{h_k\}$ we use as long as the map $(k, m) \mapsto h_k(m)$ is efficiently computable.

This is one example of the potential dangers in the "random oracle heuristic". Stronger examples were given by this paper of Canneti, Goldreich and Halevi. The conclusions (Section 6) of this paper are particularly worth reading. While the technical contents of the paper are unambiguous, the three authors each had different opinions on their *meaning* and so each author wrote his own conclusions.

5. (20 points) This question studies the need to use *min entropy* as opposed to *Shannon entropy* in cryptographic applications:

   a. (10 points) Show that there exists a distribution X over $\{0,1\}^\ell$ such that $H_{Shannon}(X) \geq \ell/100$ but $H_\infty(X) \leq 5$.

   b. (10 points) Show that for *every* $n > 100$, distribution $\frac{X}{\{0,1\}^{10n}}$ with $H_\infty(X) \leq 5$ and for *every* salt value $s \in \{0,1\}^n$, and *every* efficiently computable function $h : \{0,1\}^{10n} \to \{0,1\}^n$ there is an efficient attacker A and a pair of messages $m_0, m_1 \in \{0,1\}^n$ such that given the salt value s, A can distinguish between a sample from $h(s\|X) \oplus m_0$ and a sample from $h(s\|X) \oplus m_1$ with advantage at least $1/100$. That is, it is impossible to use a distribution with small min entropy to obtain a secure instantiation of the one time pad.

6. (30 points) We now show a few properties of min entropy, many of those hold for other entropies as well:

   a. (10 points) Show a "concavity like" property of the min entropy distribution: for every two distributions X and Y and $p \in [0,1]$ with $H_\infty(X) = H_\infty(Y) = k$, $H_\infty(pX + (1-p)Y) \geq k$. (The min entropy function can also be shown to be Schur concave , it is also known (and is a nice exercise to prove using Farkas Lemma that $H_\infty(X) \geq k$ iff X is a convex combination of flat sources with support 2^k.)

   b. (10 points) We have mentioned that in practical applications sometimes operating systems accumulates an *entropy pool* before refreshing the generator state. One approach is to simply XOR the hash of the new measurements into the old pool. This can sometimes be problematic but let us show that under the (not always realistic) assumption of *independence* it at least does not hurt: show that for every two distributions X and Y over $\{0,1\}^n$, $H_\infty(X \oplus Y) \geq \max\{H_\infty(X), H_\infty(Y)\}$ where $X \oplus Y$ denotes the distribution obtained by picking independently $x \leftarrow_R X$ and $y \leftarrow_R Y$ and outputting $x \oplus y$.

c. (10 points) Give an example of two distributions X and Y over $\{0,1\}^n$ with min entropy $n/2$ such that $H_\infty(X \oplus Y) = n/2$.