Homework 6

Due: November 1, 2024

CS 1510: Intro. to Cryptography and Computer Security

1 Hardcore Predicates for One-Way Functions

Let f be a one-way function. Define a function $B : \{0,1\}^* \to \{0,1\}$ for f, where $B(x)$ outputs the inner product modulo 2 of the first $\lfloor |x|/2 \rfloor$ bits of x and the last $\lfloor |x|/2 \rfloor$ bits of x. Prove or disprove that B is necessarily a hard-core predicate for f.

2 One-Way Functions

Let $|x|$ denote the length of the binary string x, let $\|$ denote the concatenation operator, and let $(\|)$ denote the parse operator such that when we parse $x = x_1(\|)x_2$, we get $|x_1| = |x_2|$. Assume for simplicity that all strings to which the parse operator is applied are of even length; this can be accomplished, for example, by appending a 0 to the end of an odd-length string prior to applying the parse operator. Suppose that $g(x)$ is a one-way function that is length-preserving, meaning that $|g(x)| = |x|$.

For the following functions, use a reduction to prove that the function is one-way, or give a counterexample showing that it is not necessarily one-way.

a. $f_a(x) = g(x) \oplus x$.

b. $f_b(x) = g(x_1 \oplus x_2)$, where $x = x_1(\|)x_2$.

c. $f_c(x) = \begin{cases} 0^{|x|} & \text{if exactly 1 bit of } x_1 \text{ is 1} \\ 0^{|x_1|} \| g(x_2) & \text{otherwise} \end{cases}$, where $x = x_1(\|)x_2$.

d. $f_d(x) = \begin{cases} 0^{|x|} & \text{if at least 1 bit of } x_1 \text{ is 1} \\ 0^{|x_1|} \| g(x_2) & \text{otherwise} \end{cases}$, where $x = x_1(\|)x_2$.

3 One-Way Functions Imply that $P \neq NP$

Prove that the existence of one-way functions implies $P \neq NP$.

4 Group Properties

Let $(\mathbb{G}, \cdot)$ be a group. Prove the following:

a. There is a *unique* identity in $\mathbb{G}$.

b. Every element $g \in \mathbb{G}$ has a *unique* inverse.

c. Prove that the RSA group, namely $\mathbb{Z}_N^* = \{a \in \{1, 2, ..., N-1\} \mid \mathsf{gcd}(a, N) = 1\}$, is an abelian group under multiplication modulo N.

5 CRHF from the Discrete Logarithm Assumption

Let $\mathcal{G}$ be a polynomial-time algorithm that, on input 1^n, outputs (the description of) a cyclic group $\mathbb{G}$, its order q (which is prime), and a generator g. Define a fixed-length hash function (Gen, H) as follows:

- Gen: On input 1^n, run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$ and then select uniform random $h_1, \ldots, h_t \xleftarrow{\$} \mathbb{G}$. Output $s := \langle \mathbb{G}, q, g, (h_1, \ldots, h_t) \rangle$ as the key.
- H: Given a key $s = \langle \mathbb{G}, q, g, (h_1, \ldots, h_t) \rangle$ and input $(x_0, x_1, \ldots, x_t)$ with $x_i \in \mathbb{Z}_q$, output $H_s(x_0, x_1, \ldots, x_t) := g^{x_0} \cdot \prod_{i=1}^{t} h_i^{x_i}$.

Prove that if the discrete logarithm problem is hard relative to $\mathcal{G}$, then for any t, this construction is a fixed-length collision-resistant hash function.

6 Summary Question

Summarize the most important insights from this week's material, including from the lectures, notes, textbooks, homework problems, and other resources you find helpful, into a one-page resource. We expect that these summary pages will help you with the take-home midterm and final. Please note this question is graded based on completion—we will not be checking it for correctness. However, we will be checking it for relevance to the week's content and length.