Homework 3

Written Solutions

Written Part

6. Let's prove some properties of the discrete logarithm.

   (a) Let g be a primitive root of $\mathbb{F}_p^*$. Fix $a, b \in \mathbb{Z}$ and suppose that $g^a \equiv g^b \mod p$. Show that $a \equiv b \mod (p-1)$.

   Proof. We recall that we showed in class that if $g \in \mathbb{F}_p^*$ has order d, and $g^k \equiv 1 \mod p$, then $d|k$. Since g is a primitive root, its order is $p-1$. Since $g^a \equiv g^b$ we know that $g^{b-a} \equiv 1 \mod p$, so that by what we just said, $p-1$ divides $b-a$, completing the proof. □

   (b) Use part (a) to prove that the discrete log map $\log_g : \mathbb{F}_p^* \longrightarrow \mathbb{Z}/(p-1)\mathbb{Z}$ is well defined.

   Proof. Suppose that a and b both solve $x = \log_g h$. This means $g^a \equiv h \equiv g^b \mod p$ so that by part (a) $a \equiv b \mod p-1$ so that they define the same element of the target. □

   (c) Show that the map $\log_g$ from part (b) is *bijective*. (Hint, can you construct an explicit inverse?).

   Proof. We build an expoential map $g^x : \mathbb{Z}/(p-1)\mathbb{Z} \to \mathbb{F}_p^*$. It is defined in the obvious way, for $a = \{1, 2, ..., p-1\}$ we let,

$$g^a = \underbrace{g \cdot g \cdots g}_{a \text{ times}}$$

   Then one checks that $\log_g g^a = a$ and $g^{\log_g a} = a$ by definition. □

   (d) Show that $\log_g(ab) = log_g(a) + \log_g(b)$ for all $a, b \in \mathbb{F}_p^*$. (For those of you have seen group theory, this means $\log_g$ is a homomorphism, and in light of (c) an *isomorphism*!)

   Proof. Let $x = \log_g(a)$ and $y = \log_g(b)$. This means $g^x = a$ and $g^y = b$. Therefore $ab = g^x g^y = g^{x+y}$ so that $x + y = \log_g(ab)$. □

   (e) Let p be an odd prime and g a primitive root of $\mathbb{F}_p^*$. Prove that $a \in \mathbb{F}_p^*$ has a square root if and only if $\log_g(a)$ is even.

   Proof. This is just a rephrasing of HW2 Problem 8(d), where we showed that if g is a primitive root and $a = g^k$, then a has a square root if and only if k is even. But k is precisely $\log_g a$. □

(f) (BONUS:) We've talked about how the Discrete Log Problem is rather secure. That is, given an odd prime p, a primitive root $g \in \mathbb{F}_p^*$, and some $x = g^a \mod p$, it should be hard to find a. Nevertheless, it is easy to tell whether a is even or odd. Describe a fast algorithm to do so and prove it's correct. (This is often referred to as saying the *least significant bit* of the discrete log problem is insecure).

Proof. We propose the following algorithm.

i. Compute $a^{\frac{p-1}{2}} \mod p$ (using fast powering).
ii. If it is 1, then $\log_g a$ is even, otherwise $\log_g a$ is odd.

We first notice that raising to the $\frac{p-1}{2}$ makes sense as p is an odd prime, so at the very least the algorithm will run. We second observation is that this algorithm is fast. Indeed, it just does one fast powering which is $\mathcal{O}\left(\log \frac{p-1}{2}\right) = \mathcal{O}(\log p)$. We defer revealing the proof of correctness until HW5 Problem 6. $\square$