CAS CS 538. Problem Set 2

Due via websubmit on Thursday, September 17, 2015, at 11:59pm

Optional Reading: Shoup’s A Computational Introduction to Number Theory and Algebra from sections 6.1, 6.2, and 6.5, as needed, to get comfortable with cyclic groups.

Problem 1. (49 points, at 7 each) Let $p > 2$ be a prime. Recall that $\mathbb{Z}_p$ is a field of integers modulo $p$, and $\mathbb{Z}_p^*$ is the multiplicative group of that field. Recall Fermat’s little theorem: for all $a \in \mathbb{Z}_p^*$, $a^{p-1} \equiv 1 \pmod{p}$. You may also use the following fact (without proof—see any number theory textbook or me if you’d like to see the proof): there exists a generator $g$ such that $g, g^2, g^3, \ldots, g^{p-1}$ (all modulo $p$) are all distinct and, therefore, cover all of $\mathbb{Z}_p^*$. In other words, there exists $g \in \mathbb{Z}_p^*$ such that the map $\mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*$ given by $x \mapsto g^x \pmod{p}$ is a bijection.

(a) Show that exponents work modulo $p - 1$: in other words, if $x \equiv y \pmod{p - 1}$ then, for any $a$, $a^x \equiv a^y \pmod{p}$ (hint: write $x$ as $(p - 1)k_x + r$ and $y$ as $(p - 1)k_y + r$).

(b) Show that if $g$ is a generator, then $g^x \equiv 1$ if and only if $(p - 1)|x$ (Hint: let $r$ be the remainder of $x$ modulo $p - 1$. Consider $g^r$.)

(c) Show that if $g$ is a generator, then the converse of part (a) also holds: if $g^x \equiv g^y \pmod{p}$, then $x \equiv y \pmod{p - 1}$. (Hint: use the previous part.)

(d) Let $a = g^x \pmod{p}$, where $g$ is again a generator. Show that if $x$ is even, then $a$ has a square root modulo $p$. Now show the converse: if $a$ has a square root modulo $p$, then $x$ is even. (Hint: suppose $x$ is odd. Represent $r$ as $g^y$. Then $g^{2y} \equiv g^x \pmod{p}$.) Now use the previous part to show that $x$ is even.) Thus, we can tell from the exponent whether or not the value is a square.

(e) We’d also like to be able to tell if a value is a square modulo $p$ without having to know its discrete logarithm. Show that if $a$ is a square, then $a^{(p-1)/2} \equiv 1 \pmod{p}$. Now show that if $a$ is a non-square, then $a^{(p-1)/2} \not\equiv 1 \pmod{p}$. (Hint: first write $a$ as $g^x$; then use the previous part.)

We have thus shown that exactly half the values in $\mathbb{Z}_p^*$ have square roots, and we know how to identify them: by raising to $(p - 1)/2$. Note also that values that do have square roots have exactly two of them: if $r$ is a square root of $a$, then, trivially, so is $-r$ (each square cannot have more than two square roots by a simple counting argument: there wouldn’t be enough squares otherwise).

(f) Show that if $(g^r)^2 \equiv a \pmod{p}$, then $(g^{x+(p-1)/2})^2 \equiv a \pmod{p}$, as well. Fact (you don’t need to prove it): these are two distinct square roots of $a$ (because $x \not\equiv x + (p - 1)/2 \pmod{p - 1}$). Another fact (you don’t need to prove it): if $g^x$ is a square root of $a$, then so is $-g^x$ (verified by squaring); hence $g^x \equiv -g^{x+(p-1)/2} \pmod{p}$ (because as explained above, there are at most two roots, so we must have $-g^x \equiv g^{x+(p-1)/2}$). Now show from here that $g^{(p-1)/2} \equiv -1 \pmod{p}$, and, in fact, if $b$ is a non-square, then $b^{(p-1)/2} \equiv -1 \pmod{p}$.

This refines our previous test for squares: to test if something is a square, you raise it to $(p - 1)/2$ and check if the result is 1 or −1. By the way, the value of $a^{(p-1)/2}$ is called the Legendre symbol of $a$ and is often written as $\left(\frac{a}{p}\right)$.
(g) Show that if \( p \equiv 3 \pmod{4} \), and \( a \) has a square root, then the value \( a^{(p+1)/4} \) is a square root of \( a \). (We thus have a simple algorithm to compute square roots for half the primes. The algorithm to compute square roots in other case, if \( p \equiv 1 \pmod{4} \), is a little bit more complex, but still very efficient.)

**Problem 2.** (25 points) Suppose a lottery is designed with a security parameter \( k \) (the designers can choose \( k \); we are interested in what happens as \( k \) grows). The chances of winning by playing once are \( f(k) \), where \( f \) is a negligible function in \( k \). You decide to play \( p(k) \) times, where \( p \) is a positive polynomial. Show that your chances of winning at least one of those games are still negligible in \( k \).

**Problem 3.** (26 points) Prove that if the discrete logarithm assumption (formally described in Section 2.3 of the notes) holds, then it is hard to compute \( x \) given \( g^{xy} \) and \( g^y \). Formally, prove (via a reduction) that for any poly-time algorithm \( A \), there exists a negligible function \( \eta \) such that, if you generate random \( k \)-bit \( p \) and its generator \( g \) and select a random \( x, y \in \mathbb{Z}_p^* \), \( \Pr[A(p, g, g^{xy} \mod p, g^y \mod p) = x] \leq \eta(k) \).