CAS CS 538. Problem Set 2

Due in class Tuesday, September 18, 2012, *before* the start of lecture

Problem 1. (42 points, at 6 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 $\mathbb{Z}_p^* \to \mathbb{Z}_p^*$ given by $x \mapsto g^x \bmod p$ 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 $(g^x)^2 \equiv a \pmod{p}$, then $\left(g^{x+(p-1)/2}\right)^2 \equiv a \pmod{p}$, 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. (33 points, at 3 each for the first four parts, and 7 each for the last three parts) In his monograph Pseudorandomness, Vadhan defines statistical distance differently from how we defined it in class: see Definition 6.2 of `http://people.seas.harvard.edu/~salil/pseudorandomness/extractors.pdf`. Prove Lemma 6.3, which shows useful properties of statistical distance and, in particular, shows that the definition we gave in class is equivalent.

(Hint for part 6, which is probably the hardest part: use the triangle inequality you proved in part 5. That is, use the fact that $\Delta((X_1, X_2), (Y_1, Y_2)) \leq \Delta((X_1, X_2), Z) + \Delta(Z, (Y_1, Y_2))$, where the Z is the random variable obtained by sampling X_1 and Y_2 independently (you can write Z as $(X_1 \times Y_2)$). Now prove that $\Delta((X_1, X_2), (X_1 \times Y_2)) \leq \Delta(X_2, Y_2)$ and $\Delta((X_1 \times Y_2), (Y_1, Y_2)) \leq \Delta(X_1, Y_1)$. These two have the same proofs—so proving just one of the statements is enough. Note that you are just tacking on an independent variable.)

(Hint/explanation for part 7: let $\delta_u = \Pr[X = u] - \Pr[Y = u]$. The value $|X - Y|_1$ is defined as $\sum_{u \in \mathcal{U}} |\delta_u|$.)

Problem 3. (25 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 that for any poly-time algorithm A, there exists a negligible function η such that, if you generate random k-bit p and its generator g and select a random $x, y \in \mathbb{Z}_p^*$, $\Pr[A(p, g, g^{xy} \bmod p, g^y \bmod p) = x] \leq \eta(k)$.