CS60088 Foundations of Cryptography, Spring 2013–2014

Class Test 1

12–February–2014 | F-127, 6:00–7:00pm | Maximum marks: 20

Roll no: ____________ **Name:** ________________________

[*Write your answers in the question paper itself. Be brief and precise. Answer all questions.*]

1. Let $p \equiv 3 \pmod 4$ be a suitably large prime, and g a generator of $\mathbb{Z}_p^*$. We want to compute discrete logarithms to the base g. Let $x = \operatorname{ind}_g a$. The least significant bit of x is 0 if and only if a is a quadratic residue modulo p, and this can be easily checked by computing the Legendre symbol $\left(\frac{a}{p}\right)$. However, the question of computing the second least significant bit of x is subtle.

(a) Let $\alpha \in \mathrm{QR}_p$. Prove that the two square roots of α modulo p are $\pm\alpha^{(p+1)/4} \pmod p$. **(5)**

Solution By Euler's criterion, $\alpha^{(p-1)/2} \equiv 1 \pmod p$, so $(\alpha^{(p+1)/4})^2 \equiv \alpha^{(p+1)/2} \equiv \alpha^{(p-1)/2}\alpha \equiv \alpha \pmod p$.

(b) Let $\alpha \equiv g^{2^i y} \pmod p$ for some $i \geqslant 2$. Prove that $\alpha^{(p+1)/4} \equiv g^{2^{i-1}y} \pmod p$. **(5)**

Solution Exponentiation to the $(p+1)/4$-th power gives $\alpha^{(p+1)/4} \equiv g^{(p+1)2^{i-2}} \equiv g^{(p-1+2)2^{i-2}} \equiv (g^{p-1})^{2^{i-2}y} g^{2^{i-1}y} \equiv g^{2^{i-1}y} \pmod p$, since $g^{p-1} \equiv 1 \pmod p$.

(c) Suppose that there is an oracle SLSB that, upon the input of p, g and $\alpha \in \mathbb{Z}_p^*$, returns the second least significant bit of $\operatorname{ind}_g \alpha$. Prove that the SLSB oracle can be used to design an efficient algorithm to compute discrete logarithms in $\mathbb{Z}_p^*$. **(5)**

Solution Suppose that we want to compute $x = \operatorname{ind}_g a = (x_{l-1}x_{l-2}\ldots x_2x_1x_0)_2$. The least significant bit x_0 can be determined by computing the Legendre symbol $\left(\frac{a}{p}\right)$. A query to the SLSB oracle with a as input gives x_1. Suppose now that for some $i \geqslant 2$, we have already computed $x_0, x_1, \ldots, x_{i-1}$, and we want to compute x_i. We compute $b \equiv g^{x_0+2x_1+2^2x_2+\cdots+2^{i-1}x_{i-1}} \pmod{p}$, and take $\alpha \equiv ab^{-1} \pmod{p}$. We have $\alpha \equiv g^{2^i y} \pmod{p}$, where $y = (x_{l-1}x_{l-2}\ldots x_i)_2$. Applying Part (b) $i-1$ times gives $\alpha' \equiv (\alpha^{(p+1)/4})^{i-1} \equiv g^{2y} \equiv g^{(x_{l-1}x_{l-2}\cdots x_i 0)_2} \pmod{p}$. A query to the SLSB oracle then gives us x_i.

(d) Design a pseudorandom bit generator (PRBG) as follows. Given a seed $s \in \mathbb{Z}_p^*$, set $x_0 = s$, and then compute $x_i \equiv g^{x_{i-1}} \pmod{p}$ for $i = 1, 2, 3, \ldots$. Let b_i be the second least significant bit of x_i. The output of the PRBG is the bit sequence $b_0, b_1, b_2, b_3, \ldots$. Prove that this PRBG is cryptographically secure. **(5)**

Solution For proving the previous-bit security of this PRBG, let an oracle exist that, given the bit sequence $b_0, b_1, b_2, \ldots$ from the PRBG, outputs b_{-1}, that is, the second least significant bit of $x_{-1} = \mathrm{ind}_g(x_0)$. Using this, we can build the SLSB oracle of Part (c), which in turn violates the DL assumption.

Suppose that we want to compute the second least significant bit of $\mathrm{ind}_g(x_0)$. We generate the bit sequence $b_0, b_1, b_2, \ldots$ using the PRBG with x_0 as the seed. We then query the oracle for obtaining b_{-1}. But this is precisely the second most significant bit of $\mathrm{ind}_g a$.