Advanced cryptography: Pairing-based cryptography
winter term 2012/13

DANIEL LOEBENBERGER AND MICHAEL NÜSKEN

3. Exercise sheet
Hand in solutions until Monday, 12 November 2012, 23:59:59

Exercise 3.1 (Hands on: Miller). (20+3 points)

Implement Miller's algorithm computing $\frac{f_P(Q_1)}{f_P(Q_2)}$ where $\operatorname{div}(f_P) = \ell[P+R] - \ell[P] - \ell[R] + [\mathcal{O}]$ given $P, R, Q_1, Q_2 \in E$ and the desired index ℓ in a programming language/computer algebra system of your choice. Hint: Employ a system that can handle finite field and polynomial arithmetic. Then, first implement elliptic curve arithmetic (or find an appropriate library for that task) and afterwards realize Miller's algorithm. $\boxed{20}$

Bonus task: Show (on paper) that running the algorithm is only a constant factor slower than a scalar multiplication. $\boxed{+3}$

Exercise 3.2 (Divisors of functions). (10+3 points)

(i) Consider the function $f\colon \mathbb{P}^1\mathbb{C} \to \mathbb{P}^1\mathbb{C},\ x \mapsto \frac{(x-1)^2(x-5)}{x^3(x-2)^2(x-3)^2}$, where $\mathbb{P}^1\mathbb{C} = \mathbb{C} \cup \{\infty\}$.

  (a) Compute the value $f(\infty)$. If f has a zero or a pole compute its multiplicity. Hint: Consider $f(1/x)$ and evaluate at $x = 0$. $\boxed{2}$

  (b) Compute $\operatorname{div}(f)$. $\boxed{1}$

(ii) Consider now the function $g\colon \mathbb{R} \to \mathbb{R},\ x \mapsto x^2 + 1$.

  (a) Compute $\operatorname{div}(g)$. Hint: It is not zero. $\boxed{+2}$

  (b) What is the divisor of g when we replace $\mathbb{R}$ by $\mathbb{P}^1\mathbb{R}$? $\boxed{+1}$

(iii) You are given the divisor $D = [1]-2[2]+3[3]-4[4]+2[\infty]$. Find a function $h\colon \mathbb{P}^1\mathbb{C} \to \mathbb{P}^1\mathbb{C},\ x \mapsto h(x)$ with $\operatorname{div}(h) = D$. $\boxed{2}$

(iv) Explain why finding functions on the curve E with a predescribed divisor may be more difficult than over $\mathbb{P}^1\mathbb{C}$. $\boxed{5}$

Exercise 3.3 (Security estimate). (4+6 points)

The ElGamal signature scheme works over some publicly known group of (often prime) order ℓ, where ℓ has length n. In many cases this is a subgroup of some $\mathbb{Z}_p^\times$ with another (larger) prime p; then $\ell|(p-1)$. However, it is necessary for its security that it is difficult to compute a discrete logarithm in the group and also, if applicable, in the surrounding group $\mathbb{Z}_p^\times$. The best known discrete logarithm algorithms achieve the following (heuristic, expected) running times:

method	year	time for a group size of n-bit
brute force (any group)	$-\infty$	$\mathcal{O}^\sim(2^n)$
Baby-step Giant-step (any group)	1971	$\mathcal{O}^\sim\left(2^{n/2}\right)$
Pollard's ϱ method (any group)	1978	$\mathcal{O}\left(n^2 2^{n/2}\right)$
Pohlig-Hellman (any group)	1978	$\mathcal{O}^\sim\left(2^{n/2}\right)$
Index-Calculus for $\mathbb{Z}_p^\times$	1986	$2^{(\sqrt{2}+o(1))n^{1/2}\log_2^{1/2} n}$
Number-field sieve for $\mathbb{Z}_p^\times$	1990	$2^{((64/9)^{1/3}+o(1))n^{1/3}\log_2^{2/3} n}$

It is not correct to think of $o(1)$ as zero, but for the following rough estimates just do it. Estimate the time that would be needed to find a discrete logarithm in a group whose order has n-bits assuming the (strongest of the) above estimates are accurate with $o(1) = 0$ (which is wrong in practice!)

+1 (i) for $n = 1024$ (standard size),
+1 (ii) for $n = 2048$ (as required for Document Signer CA),
+1 (iii) for $n = 3072$ (as required for Country Signing CA).

Repeat the estimate assuming that for the given group only Pollard's ϱ method is available, for example in case the group is a ℓ-element subgroup of $\mathbb{Z}_p^\times$ or an elliptic curve,

+1 (iv) for $n = 160$,
+1 (v) for $n = 200$,
+1 (vi) for $n = 240$.

In April 2001 Reynald Lercier reported (`http://perso.univ-rennes1.fr/reynald.lercier/file/nmbrJL01a.html`) that they can solve a discrete logarithm problem modulo a 397-bit prime p within 10 weeks on a 525MHz computer.

4 (vii) Which bit size for the prime p is necessary to ensure that they cannot solve the DLP problem in $\mathbb{Z}_p^\times$ given —say— 10'000 10GHz computers and 1 year (disregarding memory requirements).

[Note: The record for computing discrete logs in $\mathbb{F}_{2^n}^\times$ lies at $n = 613$, see Antoine Joux `http://perso.univ-rennes1.fr/reynald.lercier/file/nmbrJL05a.html`.]