MPRI 2-12-2 Partiel

Morain, Barbaud, Smith

27/11/2018

Time allowed: two hours. Answers may be given in French and/or English. Notes and references on paper are permitted. The six questions are independent, and may be attempted in any order.

Question A: Elementary arithmetic

1. Let u and v two *rational* non integral numbers so that their product N is integral. Show how to find the factorization of N as a product of two *integers.*
2. Suppose that $N = p^2 + q^2 = r^2 + s^2$ with p, q, r, s positive integers and $(r, s) \not\in \{(p, q), (q, p)\}$. Show that we can write N as a product of two non trivial rational numbers. *Hint:* write $p = r + x$, $q = s - y$ and reorganize the equalities.
3. Numerical application: factor $N = 221 = 100 + 121 = 196 + 25$ *using the preceding question* (and not by hand!).

Question B: Elementary discrete logarithms

Let $G = (\mathbb{Z}/p\mathbb{Z})^*$ for an odd prime p, of generator g. Let $a \in G$. We are interested in the discrete logarithm of a in base g, say $x = x_0 + 2x_1 + \cdots + 2^k x_k$ with $x_i \in \{0, 1\}$ and $x_k = 1$. We define the functions $L_i(a) := x_i$.

For simplicity, we assume that $p \equiv 3 \bmod 4$.

1. Show how one can determine x_0 in polynomial time in $\log p$ (in other words we can compute $L_0(a)$ in polynomial time for all a).
2. Show that $L_0(a) \neq L_0(p - a)$ for all a.
3. Suppose we have an oracle that gives us $L_1(b)$ for any $b \in G$. Give an algorithm that computes the whole value of x in polynomial time in $\log p$.

Question C: Coppersmith, Odlyzko, Schroeppel

Let p be a (large) prime number. We want to compute discrete logarithms in $(\mathbb{Z}/p\mathbb{Z})^*$ (generated by a given g).

Put $H = \lfloor\sqrt{p}\rfloor + 1$, $J = H^2 - p$. Consider small integers c_1 and c_2, say $c_i < L(p)^\alpha$ where $L(p)$ is the classical function $\exp((\log p)^{1/2}(\log\log p)^{1/2})$ and $\alpha > 0$.

1. Estimate the size of $(H + c_1)(H + c_2) \bmod p$.
2. Sketch an index calculus method based on this setting.
3. Show that this method can use a sieve to speed up the computations.

Question D: Square roots in $\mathbb{F}_p$

1. Let H be a cyclic group of order 2^s. Give an algorithm to compute the discrete logarithm of a given element $x \in H$ in polynomial time in s.

2. Let p be a prime, let $G = \mathbb{F}_p^*$ and let g be a generator of G. Write $p-1 = 2^s t$ for integers s and t with t odd.

   (a) Let u be the inverse of 2 modulo t. Give a formula for u.

   (b) Let a be a square in G. Show that $a^u/\sqrt{a}$ belongs to the subgroup $H := \{x^t \mid x \in \mathbb{F}_p^*\}$, generated by $h := g^t$.

   (c) Describe an algorithm which, given p, g and a, computes $\sqrt{a}$ in polynomial time in $\log_2 p$.

Question E: Montgomery curves

Let p be a prime > 3. For each $A \neq \pm 2$ in $\mathbb{F}_p$, we have an elliptic curve in Montgomery form defined by

$$\mathscr{E}_A : y^2 = x(x^2 + Ax + 1) \quad \text{over} \quad \mathbb{F}_p .$$

In projective coordinates $(X : Y : Z)$ where $x = X/Z$ and $y = Y/Z$, the defining equation of $\mathscr{E}_A$ becomes

$$\mathscr{E}_A : Y^2 Z = X(X^2 + AXZ + Z^2) .$$

We let $O = (0 : 1 : 0)$ be the "point at infinity". There at least one other obvious point in $\mathscr{E}(\mathbb{F}_p)$, namely $T = (0:0:1)$.

1. What happens if we allow $A = \pm 2$?

2. Write down all of the points in $\mathscr{E}[2](\mathbb{F}_{p^2})$. (Recall that $\mathscr{E}[2]$ is the 2-torsion subgroup of $\mathscr{E}$.)

3. The j-invariant of $\mathscr{E}_A$ is

$$j(\mathscr{E}_A) = 256\frac{(A^2-3)^3}{A^2-4} ,$$

   so $\mathscr{E}_A$ is isomorphic to $\mathscr{E}_{A'}$ (and there is a change of coordinates taking $\mathscr{E}_A$ into $\mathscr{E}_{A'}$) if and only if $A' = \pm A$.

   (a) What is the isomorphism $\mathscr{E}_A \to \mathscr{E}_{-A}$?

   (b) When is $\mathscr{E}_{-A}$ the quadratic twist of $\mathscr{E}_A$?

4. The 4-th division polynomial of $\mathscr{E}$ is

$$\begin{aligned}\Psi_4(x,y) &= 4 \cdot 2y \cdot \left(x^6 + 2Ax^5 + 5x^4 - 5x^2 - 2Ax - 1\right) \\ &= 4 \cdot 2y \cdot (x+1) \cdot (x-1) \cdot (x^4 + 2Ax^3 + 6x^2 + 2Ax + 1) .\end{aligned}$$

   Show that $\#\mathscr{E}_A(\mathbb{F}_p)$ is *always* divisible by 4, for any $A \in \mathbb{F}_p$.

Question F: The Montgomery ladder

Let p, A, $\mathscr{E}_A$, O, and T be defined as above. Suppose $P = (X_P : Y_P : Z_P)$ and $Q = (X_Q : Y_Q : Z_Q)$ are in $\mathscr{E}_A(\mathbb{F}_p) \setminus \{O, T\}$ with $Q \neq \pm P$. We write

$$P \oplus Q = (X_\oplus : Y_\oplus : Z_\oplus) \qquad \text{and} \qquad P \ominus Q = (X_\ominus : Y_\ominus : Z_\ominus) ,$$

and for every $k > 0$ we write

$$(X_{[k]P} : Y_{[k]P} : Z_{[k]P}) = [k]P .$$

If $Q \neq \pm P$ then the pseudo-addition operation xADD: $\big((X_P, Z_P), (X_Q, Z_Q), (X_\ominus, Z_\ominus)\big) \mapsto (X_\oplus, Z_\oplus)$ is defined for $P, Q \notin \{O, T\}$ by the pair of simultaneous equations

$$\begin{cases} X_\oplus = Z_\ominus \left[(X_P - Z_P)(X_Q + Z_Q) + (X_P + Z_P)(X_Q - Z_Q)\right]^2 \\ Z_\oplus = X_\ominus \left[(X_P - Z_P)(X_Q + Z_Q) - (X_P + Z_P)(X_Q - Z_Q)\right]^2 \end{cases} \tag{1}$$

The pseudo-doubling operation xDBL: $(X_P, Z_P) \mapsto (X_{[2]P}, Z_{[2]P})$ is defined for $P \notin \{O, T\}$ by the pair of simultaneous equations

$$\begin{cases} X_{[2]P} = (X_P + Z_P)^2 (X_P - Z_P)^2 \\ Z_{[2]P} = (4X_P Z_P)\left[(X_P - Z_P)^2 + C \cdot (4X_P Z_P)\right] \end{cases} \tag{2}$$

where C is the constant $(A+2)/4$. When calculating, it is useful to remember that $4X_P Z_P = (X_P + Z_P)^2 - (X_P - Z_P)^2$. To compute the map $(m, (X_P, Z_P)) \mapsto (X_{[m]P}, Z_{[m]P})$ for $m > 2$, we use the Montgomery ladder (Algorithm 1).

Algorithm 1: LADDER: The Montgomery ladder

Input: $m = \sum_{i=0}^{k-1} m_i 2^i$ with $m_{k-1} = 1$ and (X_P, Z_P) in $\mathbb{F}_p^2$ for some $P = (X_P : Y_P : Z_P)$ in $\mathcal{E}(\mathbb{F}_p) \setminus \{O, T\}$

Output: $(X_{[m]P}, Z_{[m]P}) \in \mathbb{F}_q^2$.

$(x_0, x_1) \leftarrow ((X_P, Z_P), \mathtt{xDBL}((X_P, Z_P)))$;
for $i = k-2$ **down to** 0 **do**
  if $m_i = 0$ **then**
    $(x_0, x_1) \leftarrow (\mathtt{xDBL}(x_0), \mathtt{xADD}(x_0, x_1, (X_P, Z_P)))$
  else
    $(x_0, x_1) \leftarrow (\mathtt{xADD}(x_0, x_1, (X_P, Z_P)), \mathtt{xDBL}(x_1))$
return x_0

1. What happens if we let $P = O$ or $P = T$ in Algorithm 1 and Equations (1) and (2)?
2. Suppose we have a constant-time conditional swap: that is, a function CSWAP(b, S, T), where $b \in \{0, 1\}$ and $S, T \in \mathbb{F}_p$, which returns (S, T) if $b = 0$ and (T, S) if $b = 1$. Show how to use this function to make the Montgomery ladder uniform and constant-time with respect to its scalar argument m (for scalars of fixed bit-length k). In particular, there should be no branching ("if statements") on bits of m.
3. Let $\mu(k)$, $\sigma(k)$, and $\alpha(k)$ denote the cost (in some unit of time) of computing a multiplication, squaring, and addition (or subtraction) in a k-bit prime finite field $\mathbb{F}_p$ (i.e., $k = \log_2 p$).
   (a) Derive the cost of a single iteration of the loop in the Montgomery ladder.
   (b) What is the cost of a single LADDER call using a $\log_2 p$-bit scalar?
   (c) Given any x in $\mathbb{F}_p$, we can compute x^{-1} as x^{p-2}. When is it worth replacing the input $(X_P : Z_P)$ with $(X_P/Z_P : 1)$ in LADDER?
4. Algorithm 1 requires $m_{k-1} = 1$. How can Algorithm 1 be modified to remove this requirement?