Master CSI, AGTN/ALGANT 2018-2019

Courbes elliptiques — 4TMA902U

Responsables : G. Castagnos, D. Robert

The Discrete Logarithm Problem

1 Write a function that takes k as input and that outputs (p, q, g) such that q is a prime number of k bits and p is a prime number such that $p - 1 = 2q$ and g is an element of $(\mathbf{Z}/p\mathbf{Z})^\times$ of order q.

2 Implement the naive method for computing discrete logarithms. Test on examples generated by the function of question 1.

The Baby Step / Giant Step method

This algorithm is usually credited to Shanks (1971). It's a time-memory trade-off. Let g be a generator of a cyclic group $(G, \times)$ of order n and $h \in G$. We look for x such that $h = g^x$.

Let $m = \lceil n \rceil$ we write $x = i + mj$ with $0 \leqslant i, j < m$.

3 Show that $h(g^{-1})^i = (g^m)^j$.

The Baby Step / Giant Step method consists in pre-computing a list of $((g^m)^j, j)$ with $j < m$ sorted with respect to the first coordinate. Then we compute $h, hg^{-1}, h(g^{-1})^2, \ldots$ and look for this element in the list.

4 What is the complexity in time and memory of this algorithm (Sorting a list of size ℓ can be done in $\mathcal{O}(\ell\, log(\ell))$ operations and searching for an element in a sorted list of size ℓ can be done in $\mathcal{O}(log(\ell))$ operations)?

5 Implement this algorithm and test on examples generated by the function of question 1.

The Pollard ρ method

This method was proposed by Pollard in 1978. It has the same time complexity as the Baby Step / Giant Step method but uses quasi no memory. However, it is a probabilistic algorithm: sometimes no result is found.

Let g be a generator of a cyclic group $(G, \times)$ of order n and $h \in G$. We look for x such that $h = g^x$.

6 Suppose that we have found integers i, i', j, j' such that $(j' - j)$ is invertible modulo n and $g^i h^j = g^{i'} h^{j'}$. Show that we can find x from these integers.

7 To find these integers, one can store all the tuples $(g^i h^j, i, j)$, sorted with respect to the first coordinates, with random i and j until the same element of G is found two times. Using the birthday paradox, what is the complexity of this method (in time and memory) ?

To get rid of the need of storage, Pollard proposes to iterate a function $f : G \to G$ that « looks like » a random function. We start with a random $X_0 \in G$, then we define $X_m = f(X_{m-1})$. The initial proposal of Pollard consists in partitioning G with three sets S_0, S_1, S_2, and setting $f(X) = X^2$ if $X \in S_0$, $f(X) = hX$ if $X \in S_1$ and $f(X) = gX$ if $X \in S_2$.

8 Show that given $X \in G$ and i, j such that $X = g^i h^j$, one can efficiently compute i', j' s.t. $f(X) = g^{i'} h^{j'}$.

To find the collision, Pollard proposes to use the method of Floyd, «the tortoise and the hare algorithm »: One iteratively compute the position of the tortoise and the hare: $(X_m, X_{2m}) = (f(X_{m-1}), f \circ f(X_{2(m-1)}))$.

The sequence $(X_m)_{m \in \mathbf{N}}$ is ultimately periodic. Let ℓ be the index of the first element of the cycle and c the period. As a result, $X_0, \dots, X_{\ell-1}, X_\ell, \dots, X_{\ell+c-1}$ are all distinct and $X_{m+c} = X_m$ for $m \geqslant \ell$. With the Floyd method, we find the smallest u s.t. $X_u = X_{2u}$.

9 Show that $\ell \leqslant u \leqslant \ell + c$.

10 Deduce from that a probabilistic algorithm that compute discrete logarithms in G in time $\mathcal{O}(\sqrt{n})$ with a few memory.

11 Implement the ρ method of Pollard in $\mathbf{Z}/p\mathbf{Z}$. In order to partition $\mathbf{Z}/p\mathbf{Z}$, one can defined S_0, S_1, S_2 with $S_i = \{X \in \mathbf{Z}/p\mathbf{Z}, X \equiv i \pmod 3\}$.

12 Write a function that takes as inputs two integers k and ℓ with $\ell \leqslant k$ and that outputs E, p, q, P such that E is a random elliptic curve over $\mathbf{F}_p$ with p a k bits prime, q a prime with at least ℓ bits dividing the order of $E(\mathbf{F}_p)$ and P a point of $E(\mathbf{F}_p)$ of order q. Adapt the Pollard ρ method for that curve.