Master CSI, AGTN/ALGANT 2019-2020

Courbes elliptiques — 4TMA902U

Responsables : G. Castagnos, D. Robert

Terminal Exam — December 13, 2019

3h
Documents are not allowed
Answer the two parts on separate sheets

G. Castagnos' Part

$\boxed{1}$ Let $(G, \times)$ be a finite cyclic group of order ω. We suppose that ω factors in a product of ℓ small distinct primes $p_i : \omega = \prod_i^\ell p_i$ with $p_i < C\log(\omega)$ where $i = 1, \ldots, \ell$ and $C \in \mathbf{N}$ is a constant. Let $g \in G$ of order ω and $h = g^d$ with $1 < d < \omega$.

(a) Give an efficient algorithm (in pseudo code) that takes G, g, h and $p_1, \ldots, p_\ell$ as inputs and that outputs the integer d. What is its approximate complexity?

In the following, we consider an elliptic curve E of equation $y^2 = x^3 + ax + b$ over the finite field $\mathbf{F}_p$ where p is a large prime. Let P be a point of order n of $E(\mathbf{F}_p)$ where n is a large prime, $p \neq n$. We denote $(G, +) = \langle P \rangle$ the group of points generated by P. We consider a variant of the Elgamal encryption algorithm, adapted to elliptic curves. We denote EC-Elgamal this public-key encryption scheme.

The secret key is an integer s with $1 < s < n$. The public key is the point $Q = sP$. The parameters $p, (a, b), n, P$ are also public and we suppose in the following that there are implicit inputs to all the algorithms.

In order to encrypt a message $m \in \mathbf{F}_p$ with the public key Q, one takes a random integer r with $1 < r < n$. Denoting $R := (x_R, y_R) := rQ$, the encryption of m is $c := (c_1, c_2) := (rP, x_R + m) \in G \times \mathbf{F}_p$.

(b) What is the goal of a public-key encryption scheme?

(c) Give a decryption algorithm for EC-Elgamal (in pseudo code).

(d) Let d be an integer with $1 < d < n$. We denote $H = (x_H, y_H) = dP$. Suppose in this question only that you have access to an oracle that solve the discrete logarithm problem in G: you can give points $T \in G$ to this oracle which gives you back the integer f such that $T = fP$.

Show that with access to this oracle, given $p, (a, b), n, P$ and x_H, you can compute efficiently $\pm d \pmod{n}$.

(e) Let Q be a public key for EC-Elgamal and s the corresponding secret key.

Suppose in this question that you know the public key but not the secret key. However, suppose that you have black box access to a machine that implements the decryption algorithm using this secret key: you can give ciphertexts c to this machine that gives you back the output of the decryption algorithm using the secret key s. Moreover, instead of giving ciphertexts c such that c_1 is a point of $E(\mathbf{F}_p)$, we suppose that you can give, to this machine, points c_1 of an elliptic curve E' over $\mathbf{F}_p$ with an equation $y^2 = x^3 + ax + b'$ where $b' \neq b$. We suppose that in this case the machine executes the decryption algorithm without noticing that the input is not well-formed.

Give an efficient attack, using well chosen curves E' and the previous questions, that uses this machine to recover the secret key s, knowing only the public parameters and the public key Q.

(f) This question is independent of the previous ones. In this question only, we suppose that the elliptic curve E used in EC-Elgamal is supersingular. Propose an efficient attack that allows to test if a given ciphertext c encrypts a given message m, knowing only the public parameters and the public key Q.

$\boxed{2}$ Let p_1 and p_2 be two large prime numbers of λ bits, with $p_1 \neq p_2$ and $n = p_1p_2$. Let $(G, +)$ and $(G_t, \times)$ be two cyclic groups of order n. We denote by P a generator of G and $Q \in G$ an element of order p_1.

We denote by $e : G \times G \rightarrow G_t$ a cryptographic pairing (of type I).

We consider the following public-key encryption scheme. The public key is (P, Q, n). Let B be an integer and $m \in \{0, \dots, B\}$. To encrypt m, one chooses a random r with $1 < r < n$ and compute the ciphertext $C = mP + rQ$.

(a) Give a secret key and a decryption algorithm for this encryption scheme. How to choose the bound B in order to have an efficient decryption procedure ?

(b) We denote by C a ciphertext of m and by C′ a ciphertext of m'. Show that without knowing the private key, one can build from C and C′ a ciphertext of $m + m'$.

(c) Same question to build a ciphertext of $m \times m'$ (this ciphertext can be of a different form than C and C′ but must be still decipherable to give $m \times m'$ knowing the private key).

(d) What problems an opponent must resolve in order to compute the private key from the public key ?

(e) Give an algorithm (in pseudo code) that takes an integer λ as input, and that outputs (with the previous notations), p_1, p_2, n, P and Q. Precise how to define the pairing e. What must be the sizes of n, p_1 and p_2 in order to have a secure scheme ?

D. Robert's Part

$\boxed{3}$ Let P be a point on an elliptic curve E and n an integer. Denote $(b_k, \dots, b_1, b_0)$ the binary decomposition of n: $n = \sum_{i=0}^{k} b_i 2^i$. For instance the binary decomposition of 229 is (1,1,1,0,0,1,0,1).

(a) We recall that the Sage function 229`.digits(2)=[1, 0, 1, 0, 0, 1, 1, 1]` gives the binary decomposition of 229 from right to left: $[b_0, b_1, \dots, b_k]$.

Write a Sage function base2 (or an algorithm in pseudo code) which computes the left to right binary decomposition of n: `base2(n)=`$[b_k, \dots, b_1, b_0]$. (Here the bits have to be computed in the order $b_k, \dots, b_0$.) Hint: k `=floor(n.log(2))`.

(b) Write a Sage function `scalar` (or an algorithm in pseudo code) taking as input $[b_k, \dots, b_1, b_0]$, P and E and returning n.P. Compute the number of doubling and additions (depending on the bits b_i).

Explain the doubling and additions this function would do when calling it with $n = 229$.

(c) If we precompute 2P, 3P, explain how to improve the algorithm `scalar` using a window of size 2 and write the corresponding Sage function.

Apply this algorithm to $n = 229$, and count the number of doubling and additions (don't forget the precomputations). Explain the computations (including the precomputations) we would do with $n = 229$ and a window of size 3.

(d) Same questions for a sliding window of size 2 and then 3 applied to $n = 229$. We recall that for a sliding window of size 2, we only precompute 2P, 3P and that for a sliding window of size 3, we only precompute 4P, 5P, 6P, 7P.

(e) Rather than trying to write n as a sum $n = \sum_{i=0}^{k} b_i 2^i$ where $b_i \in \{0,1\}$, we try to write it as a sum where $b_i \in \{-1,0,1\}$. Such a decomposition is not unique, but show that it is unique if we impose it to be in non adjacent form (naf): $n = \sum_{i=0}^{k} b_i 2^i$ where $b_i \in \{-1,0,1\}$, and if $b_i \neq 0$ with $i > 0$ then $b_{i-1} = 0$.

(f) Let `naf` be the following function:

```
def naf(n):
  r=[]
  while(n!=0):
    if n%2==0:
      r.insert(0,0) #this prepends 0 to the list
    else:
      z=2-n%4; r.insert(0,z) #this prepends z to the list
      n=n-z
    n=n // 2
  return(r)
```

Show that this function computes a non adjacent form of n, in particular the naf form always exists and is unique by the preceding question.

(g) Non adjacent forms are particularly useful for elliptic curves because computing $-$P is not costly: give the expression of $-$P in terms of $P = (x_P, y_P)$.

We compute naf(229) $= [1,0,0,-1,0,0,1,0,1]$. Explain how to use this result to compute 229.P and count the number of doubling and additions.

(h) Let Q be another point in E and m an integer. Give an algorithm computing nP + mQ that is faster than the naive algorithm which compute separately nP, mQ and then the sum nP+mQ. (Hint: precompute P + Q and do a double and add algorithm where the addition step can involve P, Q or P + Q according to the current bits of n and m.)

Compute the average number of doubling and additions according to the size of n and m and compare to the naive algorithm above.

$\boxed{4}$

(a) Let $E : y^2 = x^3 + x$ be a curve over a finite field $\mathbf{F}_p$. Show that E is an elliptic curve when $p \neq 2$. **From now on we assume this is the case.**

(b) Show that -1 is a square in $\mathbf{F}_p$ if and only if $p \equiv 1 \pmod 4$.

(c) If $p \equiv 1 \pmod 4$ show that all the points of 2-torsion of the elliptic curve (meaning points $P \neq 0_E$ such that $2P = 0_E$) are defined over $\mathbf{F}_p$. If $p \equiv 3 \pmod 4$ show that there is one point of 2-torsion over $\mathbf{F}_p$ and two over $\mathbf{F}_{p^2}$.

(d) **Assume from now on that** $p \equiv 1 \pmod 4$ **and let** ξ **be a square root of** -1 **in** $\mathbf{F}_p$. Show that $[\xi]$ defined by $[\xi](x, y) = (-x, \xi y)$ sends a point $P \in E$ to a point $[\xi]P$ in E.

(e) Show that $[\xi]^4 = 1$. Here by $[\xi]^4$ we mean $[\xi]$ iterated four times.

(f) Show that $[\xi](P + Q) = ([\xi]P) + ([\xi]Q)$, and deduce that $[\xi]$ is an endomorphism of E.

(g) If r is a prime divisor of $\#E(\mathbf{F}_p)$, recall the definition of the embedding degree d and of the Weil and Tate pairing for $E[r]$.

(h) Assume that the embedding degree d of E for r is not equal to 1, show that $\#E[r](\mathbf{F}_p) = r$.

(i) If $d > 1$, let $P \in E(\mathbf{F}_p)$ be a point of r-torsion. Show that $[\xi]P$ is still a point of r-torsion, and deduce that there exists $\lambda \in \mathbf{Z}/r\mathbf{Z}$ such that $[\xi]P = \lambda P$.

(j) Show that $\lambda^2 = -1$ and deduce that $r \equiv 1 \pmod 4$.

(k) Let $m \in \mathbf{Z}/r\mathbf{Z}$. We admit that we can write $m = m_0 + \lambda m_1$ with $m_0, m_1 \approx \sqrt{r}$. Explain with the help of question 3 (h) how we can use this decomposition to speed up the computation of mP.

(l) Still if $d > 1$, define $G_1 = \{P \in E[r] \mid \pi P = P\}$ and $G_2 = \{P \in E[r] \mid \pi P = pP\}$ where π is the Frobenius of $\mathbf{F}_p$ acting on E. Show that $[\xi]$ commutes with π and deduce that $[\xi]$ stabilizes G_1 and G_2.

(m) If $P \in G_1$ and $Q \in G_2$ show that $e_r(P, [\xi]Q) = e_r(P, Q)^\lambda$ where $\lambda^2 \equiv -1 \pmod r$.

(n) Show that if r is a prime divisor of $\#E(\mathbf{F}_p)$ with $r \equiv 3 \pmod 4$, and $P \neq 0_E$ a point of r-torsion, then $(P, [\xi]P)$ is a basis of $E[r]$. What is the embedding degree d for this r?

(o) When $r \equiv 3 \pmod 4$ show that we can construct a type I pairing on the subgroup $< P >$ generated by a point of r-torsion P.

(p) Let $p = 13$. Compute the two possible values for ξ.

(q) We compute that $\#E(\mathbf{F}_{13}) = 20$. Show that $E(\mathbf{F}_{13}) \simeq \mathbf{Z}/2\mathbf{Z} \times \mathbf{Z}/10\mathbf{Z}$.

(r) We check with Sage that if $P = (4, 4)$, then $5.P = 0_E$. Deduce that $[\xi]P = [\pm 2]P$.

(s) What is the embedding degree d for $r = 5$?