Elliptic Curves over Finite Fields

Stefano Marseglia, Utrecht University

Utrecht Summer School 2019, August 28

1 Introduction

In this lecture notes we will introduce the *discrete logarithm problem* (DLP) for an abstract abelian group and discuss (some of) the fastest methods to solve it. Since these methods for a general abelian group have exponential running time, the DLP is a good candidate to be used as the underlying hard mathematical problem for public key cryptosystems. Hence we need to find a suitable candidate for our abelian group to set up our DLP. It turns out that the group of points of an elliptic curve over a finite field is a good candidate. We will compare the Elliptic Curve Discrete Logarithm Problem (ECDLP) with the "classical" DLP defined over $\mathbb{F}_p^*$ and discuss the advantages of the ECDLP.

There is a great amount of literature on these topics. Together with the references appearing at the end of the notes, we recommend the following books

- [HPS14] Hoffstein, Pipher and Silverman "An introduction to mathematical cryptography", and
- [Was08] Washington "Elliptic curves, Number Theory and Cryptography".

If you find typos or have comments don't hesitate to contact me at `s.marseglia@uu.nl`.

Acknowledgements

We express our gratitude to Harry Smit for proof-reading a preliminary version of the notes and for helpful comments.

v. August 25, 2019

2 DLP for abstract groups

Let G be a finite abelian group. For the group operations on G we will use the multiplicative notation, unless otherwise specified.

Definition 1. The *discrete logarithm problem* (DLP) in G consists of, given a pair of elements (g, h) of G, finding a positive integer n such that $g^n = h$, if such an n exists.

Observe that a DLP (g, h) has a solution if an only if h belongs to the *cyclic* subgroup $\langle g \rangle \trianglelefteq G$. We will often restrict ourselves to the case when G is cyclic.

If a solution n_0 exists then it is not uniquely defined. Indeed, for every integer k we have that $n_0 + kN$ is a solution, where N is the order of the base g, that is, $N = \#\langle g \rangle$. More precisely we have the following:

Lemma 2. *Let G be a finite cyclic group of order N with generator g. The map*

$$\begin{aligned} \log_g : G &\longrightarrow \mathbb{Z}/N\mathbb{Z} \\ h &\mapsto n \bmod N \end{aligned}$$

with $g^n = h$, is a group isomorphism.

2.1 Application: Diffie-Hellman key exchange

Alice and Bob want to agree on a key (to be used in some cryptosystem) but the only way to communicate is via a not-secure channel.

Alice	**Bob**
Public key creation	
Alice and Bob publicly agree on a cyclic group G of order N and a generator g	
Secret keys creation	
Chooses a secret $1 < a < N$	Chooses a secret $1 < b < N$
Computes $A = g^a$	Computes $B = g^b$
Public keys communication	
Sends A to Bob	Sends B to Alice
Shared key creation	
Computes $K = B^a$	Computes $K' = A^b$

Table 1: Diffie-Hellman key exchange

Exercise 3. (a) Prove that $K = K'$, so that in fact now Alice and Bob have a common shared key.

(b) Assume that Eve is monitoring the communication channel used by Alice and Bob. This means that she has knowledge of the public parameters G, g, N and of the public keys A and B. Assume that Eve has access to an *oracle* (i.e a "magic black-box") that solves arbitrary DLPs in cyclic groups. How can she recover the shared key K?

Remark 4. The *Diffie-Hellman Problem* (DHP) consists of finding $K = g^{ab}$ given as input the pair (A, B) where $A = g^a$ and $B = g^b$ without knowing the exponents a and b. In Exercise 3 you have seen that DLP is *stronger* than DHP, that is, being able to solve arbitrary DLP produces an algorithm to solve DHP. It is not known if the converse holds!

2.2 Application: ElGamal Public key cryptosystem

Alice wants to send Bob a message over a not-secure channel and she has access to an *encoding scheme* that translates the message into an element m of a cyclic group $G = \langle g \rangle$ of order N, say $m = g^n$. Note that encoding does not add any security. It is just a translation into mathematical data that can be fed into a cryptosystem and/or communicated over a digital line.

Alice	**Bob**
Key creation	
	Chooses a secret $0 < b < N$ Computes $B = g^b$ Publishes B
Encryption	
Chooses a random secret $1 < k < N$	
Computes $c_1 = g^k$	
Computes $c_2 = mB^k$	
Sends the cipher-text (c_1, c_2) to Bob	
Decryption	
	Computes $m' = c_2(c_1^b)^{-1}$

Table 2: ElGamal cryptosystem

Exercise 5. (a) Prove that $m = m'$, so that in fact Bob can read the message m.

(b) Assume that Eve is eavesdropping on the communication. Assume that Eve has access to an oracle that solves arbitrary DLPs for arbitrary cyclic groups. How can she decrypt the message m?

(c) Assume that Eve has access to an oracle that decrypts arbitrary ElGamal cipher-texts. How can she use it to solve a specific DHP? Vice versa, assume that she has access to an oracle that solves arbitrary DHPs. Can she use such oracle to decrypt an ElGamal cipher-text?

3 Solving the DLP

3.1 $\mathcal{O}$-notation and running times

Definition 6. Let $f(x)$ and $g(x)$ be real functions with values in $\mathbb{R}_{\geq 0}$. We say that f is "big-$\mathcal{O}$ of g", in symbols,

$$f(x) = \mathcal{O}(g(x))$$

if there are positive constants C, c such that

$$f(x) \leq Cg(x) \text{ for all } x \geq c.$$

Proposition 7. *If the limit*

$$\lim_{x\to\infty} \frac{f(x)}{g(x)}$$

exists and is finite then $f(x) = \mathcal{O}(g(x))$.

Definition 8. Consider an algorithm $\mathcal{A}$ that takes as input a number with k digits. We say that the algorithm $\mathcal{A}$ runs in

- *polynomial time* if there is a constant $A \geq 0$ such that $\mathcal{A}$ terminates in $\mathcal{O}(k^A)$-steps.
- *exponential time* if there is a constant $A \geq 0$ such that $\mathcal{A}$ terminates in $\mathcal{O}(e^{AK})$-steps.
- *sub-exponential time* if for every constant $a \geq 0$ the algorithm $\mathcal{A}$ terminates in $\mathcal{O}(e^{aK})$-steps.

In the next exercise, you will be asked to think why we do not specify the base of the input. In particular, if the input is N then one can take $k = \log N$. Also, we are intentionally being not precise with the definition of "step". Usually one considers (modular) multiplication and (modular) addition as basic steps, but depending on the hardware where the algorithm is run, one might want to separate them or not. For this course, we will not worry about such technicalities, as well as as the embedded constants in the definition of the $\mathcal{O}$-notation. Indeed it might happen that in practice for some range of inputs a sub-exponential algorithm is faster than a polynomial one.

We will consider a (mathematical) problem to be *easy* if it can be solved in polynomial time and *hard* if it requires exponential time, leaving the sub-exponential time somewhere in the middle, say *quite hard*.

Exercise 9. (a) Write 83 in binary (base 2).

(b) Write 1010111 in decimal (base 10) expansion.

(c) Consider an algorithm that given an input with k digits (in some base N) terminates in polynomial, exponential or sub-exponential time. Explain why it is not necessary to specify the value of N in order to discuss the complexity of the algorithm (as long as it is understood that the base is a constant).

(d) Consider an algorithm that given an input with k digits takes $\mathcal{O}(f(k))$ steps to terminate. In the following cases, determine whether the algorithm is running in polynomial, subexponential or exponential time?

- $f(k) = e^{\sqrt[3]{\log k}}$;
- $f(k) = e^{\sqrt{k}}$;
- $f(k) = k^k$.

3.2 Shanks's Baby-Step-Giant-Step Algorithm - BSGS

Here we describe a *collision* method or *meet-in-the-middle* method to solve a DLP.

Proposition 10 (Shanks's Babystep–Giantstep Algorithm - BSGS)**.** *Let G be an abelian group and let $g \in G$ be an element of order $N \geq 2$. The following algorithm solves the discrete logarithm problem $g^x = h$ in $\mathcal{O}(\sqrt{N}\log N)$ steps using $\mathcal{O}(\sqrt{N})$ of storage.*

1. *Put $n = 1 + \lfloor \sqrt{N} \rfloor$.*
2. *Create two lists*

$$\begin{aligned} &\textit{List } 1 : 1_G, g, g^2, \ldots, g^n \\ &\textit{List } 2 : h, hg^{-n}, hg^{-2n}, \ldots, hg^{-n^2} \end{aligned}$$

3. *If there is a match between the two lists, say $g^i = hg^{-jn}$, then return $x = i + jn$, ...*
4. *... otherwise return "there is no solution".*

Exercise 11. Prove that the algorithm is correct, that is, if there is a solution to the DLP then it will be found.

Remark 12. In Proposition 10 building List 1 requires $n \approx \sqrt{N}$ operations, while in each step in building List 2 one needs to perform two multiplications (eg. $h(g^{-n})(g^{-n})$). In the statement of Proposition 10 we are assuming that the group operation can be performed in polynomial time and that the Lists can be sorted. Indeed under this assumption looking for a match can be performed one of the many well known searching algorithms in $\mathcal{O}(n \log n)$ steps. It is a good exercise to try to guess how such an algorithm works and its complexity.

Exercise 13. Consider the finite group $G = \mathrm{GL}_2(\mathbb{F}_5)$ of 2×2 invertible matrices with entries in $\mathbb{F}_5$. Consider the matrices

$$g = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \text{ and } h = \begin{pmatrix} 2 & 2 \\ 2 & 4 \end{pmatrix}$$

Observe that the order of g in G is 20. Use Shank's baby step giant step to solve the following Discrete Logarithm Problem $g^x = h$.

3.3 Pohlig-Hellman Algorithm

Exercise 14. (a) Let $m_1, \ldots, m_t$ be positive pairwise coprime integers. Put $m = m_1 \cdot \ldots \cdot m_t$. Prove the Chinese reminder theorem that states that given integers $n_1, \ldots, n_t$ there exists a unique solution $\mathrm{mod} m$ to the system of modular equations:

$$\begin{cases} x \equiv n_1 \bmod m_1 \\ x \equiv n_2 \bmod m_2 \\ \ldots \\ x \equiv n_t \bmod m_t \end{cases}$$

(b) Prove that such a solution can be found in $\mathcal{O}(\log m)$ steps. Hint: you can use without proof that given coprime positive integers m_1 and m_2 one can find in polynomial time (in $\log(m_1 m_2)$) integers a and b such that $1 = am_1 + bm_2$.

Theorem 15. *Assume that we have an algorithm that can solve a DLP $g_0^x = h_0$ where g_0 has order a prime power q^e in S_{q^e} steps. Now let g have order N with prime factorization*

$$N = q_1^{e_1} q_2^{e_2} \cdots q_t^{e_t}.$$

Then the DLP $g^x = h$ can be solved in

$$\mathcal{O}\left(\sum_{i=1}^{t} S_{q_i^{e_i}} + \log N \right)$$

steps using the following procedure:

1. *For each $1 \leq i \leq t$, let*

$$g_i = g^{N/q_i^{e_i}} \quad \text{and} \quad h_i = h^{N/q_i^{e_i}}.$$

Since the order of g_i is q_i so we can find a solution n_i to

$$g_i^x = h_i$$

in $S_{q_i^{e_i}}$ steps.

2. *Use the Chinese reminder theorem to find a solution n to the system*

$$\begin{cases} n \equiv n_1 \bmod q_1^{e_1} \\ n \equiv n_2 \bmod q_2^{e_2} \\ \dots \\ n \equiv n_t \bmod q_t^{e_t} \end{cases}$$

in $\mathcal{O}(\log N)$ steps.

3. *return n.*

Proof. Use the Chinese Reminder Theorem: there are group isomorphisms

$$\langle g \rangle \simeq {}^{\mathbb{Z}}\!/_{N\mathbb{Z}} \simeq \prod_{i=1}^{t} {}^{\mathbb{Z}}\!/_{q_i^{e_i}\mathbb{Z}} \simeq \prod_{i=1}^{t} \langle g_i \rangle .$$

□

Theorem 16. *Assume that we have an algorithm that can solve a DLP $g_0^x = h_0$ where g_0 has prime order q in S_q steps. Now let g have order q^e. Then the DLP $g^x = h$ can be solved in $\mathcal{O}(eS_q)$ steps.*

Proof. Write

$$x = x_0 + x_1 q + x_2 q^2 + ... + x_{e-1} q^{e-1} \text{ with } 0 \leq x_i < q,$$

that is, write the q-basis expansion of x. Now we can find x_0 in S_q steps by solving

$$h^{q^{e-1}} = (g^x)^{q^{e-1}} = \left(g^{q^{e-1}}\right)^{x_0},$$

and proceed recursively. □

Exercise 17. Fill in the details of the proof of Theorem 16.

3.4 An (historically) important example: $G = \mathbb{F}_p^*$

When the ElGamal public key cryptosystem was introduced in 1985 [ElG85] he proposed to work with the group $G = \mathbb{F}_p^*$.

Exercise 18. Explain why a DLP over $\mathbb{F}_p^*$ where $p-1$ factors into a product of powers of small primes is not safe against an attack based on Pohlig-Hellman theorems.

Exercise 19. Alice and Bob agree to use the prime $p = 679$ and the base $g = 2$ for a Diffie–Hellman key exchange. Alice sends Bob the value $A = 188$. Bob uses the secret exponent $b = 625$.

(a) What value B will Bob send to Alice, and what is their secret shared value?

(b) Can you figure out Alice's secret exponent? (using directly the BSGS without Pohlig-Hellman, might require a bit of computation).

Exercise 20. Alice wants to send two messages m_1 and m_2 to Bob using the ElGamal PKC using the group $\mathbb{F}_p^*$ with $p = 659$ and generator $g = 2$. Bob's public key is $B = 374$ which is used by Alice to produce the cipher-texts $(c_{1,1}, c_{1,2}) = (24, 309)$ and $(c_{2,1}, c_{2,2}) = (24, 242)$, which are the encryptions of m_1 and m_2 respectively. Assume that Eve is monitoring the conversation so she knows all public parameters and the cipher-texts. Moreover, she discovers that the first secret message is "Hello" which in the encoding scheme corresponds to $m_1 = 562$. Use this information to help Eve decrypting m_2 (without solving any DLP).
Hint: $c_{1,1} = c_{2,1}$! Which mistake has Alice made?
This kind of attack is known as *known-plaintext attack* .

The choice of $G = \mathbb{F}_p^*$ seems quite natural and well-suited for applications based on the DLP:

- G is a cyclic group and it is easy to find a generator.
- the elements of G are easy to represent by using $\{0, 1, 2, \ldots, p-1\}$.
- it is easy to guarantee that a DLP in G is safe against a Pohlig-Hellman attack (Theorems 15 and 16), by choosing a prime p of the form $p = 2q + 1$ where q is a large prime (see Exercise 18).

On the other hand, time proved that G is not a great choice for a DLP-based cryptosystem or signature scheme. The underlying reason is that $\mathbb{F}_p^*$ is not just an abelian group but the multiplicative group of a finite field. This means that its algebra is much more "rigid" than the one of a "random" abelian group and hence it should not be a surprise that there are methods to solve DLPs in $\mathbb{F}_p^*$ which are much faster than the combination of Shank's BSGS and Pohlig-Hellman.

Theorem 21 (Index-Calculus)**.** *There is an algorithm to solve a DLP*

$$g^x = h \bmod p$$

running in sub-exponential time (more precisely it requires approximatively $L(p)^{\sqrt{2}} = e^{\sqrt{2 \ln p \ln \ln p}}$*-steps).*

The idea of the index-calculus goes as follows. Instead of solving directly

$$g^x = h \bmod p$$

we first fix a positive integer B and find the solution $\log_g(l)$ of

$$g^x = l \bmod p$$

for all primes $l < B$. Now we look at quantities

$$h \cdot g^{-k} \bmod p \text{ for } k = 1, 2, \ldots$$

until we find one which is B-*smooth*, that is, its prime factorization involves only primes $l < B$:

$$h \cdot g^{-k} = \prod_{l<B} l^{e_l} \bmod p.$$

This is equivalent to

$$\log_g(h) = k + \sum_{l<B} e_l \log_g(l) \bmod (p-1).$$

Since we are assuming that we have already computed the values of $\log_g(l)$ and that we know k, this returns the desired value of $\log_g(h)$.

As you can see the idea underlying Theorem 21 is quite simple. On the other hand, estimating the "best" value of B to make the method succeed, which directly implies the sub-exponential running-time of the algorithm, requires a lot of work and some advanced tools from analytic number theory.

Moreover, a very recent pre-print by Kleinjung and Wesolowki proposes a probabilistic method to solve DLP in $\mathbb{F}_q^*$ (with $q = p^d$) in expected quasi-polynomial time! `https://eprint.iacr.org/2019/751.pdf`

This consideration should convince you that one should find a different group G, where the only known algorithms to break a DLP are fully exponential (as the combination Shank's BSGS + Pohlig-Hellman).

This is the moment when Elliptic Curves get on then stage.

4 Elliptic Curves

Definition 22. An *elliptic curve* over a field k is a non-singular projective curve over k of genus 1 equipped with a marked k-rational point.

One can prove that an elliptic curve can always be represented by (the projective closure) of a "(long) Weierstrass equation"

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

with $a_i \in k$. The marked point in this model is the (unique) *point at infinity*. If the characteristic of the field is not 2 or 3 there is a change of variable that sends the previous equation to the "(short) Weierstrass equation"

$$Y^2 = X^3 + AX + B,$$

with A, B in k. Again, the marked point is the unique point at infinity of this model. Moreover, since the elliptic curve is non-singular, the discriminant $\Delta = -16(4A^3 + 27B^2)$ is non-zero.

Exercise 23. Let C be an algebraic plane curve defined by a polynomial equation $f(X, Y) = 0$ over a field k. Recall that C is *smooth* or *non-singular* if the system of equations

$$\begin{cases} f(X,Y) = 0, \\ \frac{\partial f}{\partial X}(X,Y) = 0, \\ \frac{\partial f}{\partial Y}(X,Y) = 0 \end{cases}$$

has no solutions over the algebraic closure $\bar{k}$ of k.

Assume now that the characteristic of the field k is not 2. Let E be the elliptic curve defined by the short Weierstrass equation

$$Y^2 = X^3 + AX + B$$

over k. Recall that we defined the discriminant of E as

$$\Delta_E = -16(4A^3 + 27B^2).$$

Prove that the following statements are equivalent:

(a) E is smooth.

(b) $\Delta_E \neq 0$.

(c) the polynomial $g(X) = X^3 + AX + B$ has distinct roots over $\bar{k}$, that is there are distinct elements e_1, e_2 and e_3 in $\bar{k}$ such that

$$g(X) = (X - e_1)(X - e_2)(X - e_3).$$

Remark 24. Note that a short Weierstrass equation is always singular over a field k of characteristic 2. In characteristic 3 it might not be possible to apply the required change of variable to go from a long Weierstrass equation to the shot one, so one might need more coefficients. Fro sake of simplicity, we will always assume that k has characteristic different from 2, and that we can identify an elliptic curve E over k with its short Weierstrass equation.

The set of *k-rational points* $E(k)$ of an elliptic curve E can be described as the set of solutions of the equation $Y^2 = X^3 + AX + B$, the *affine* points, together with the point at infinity $\mathcal{O}$.

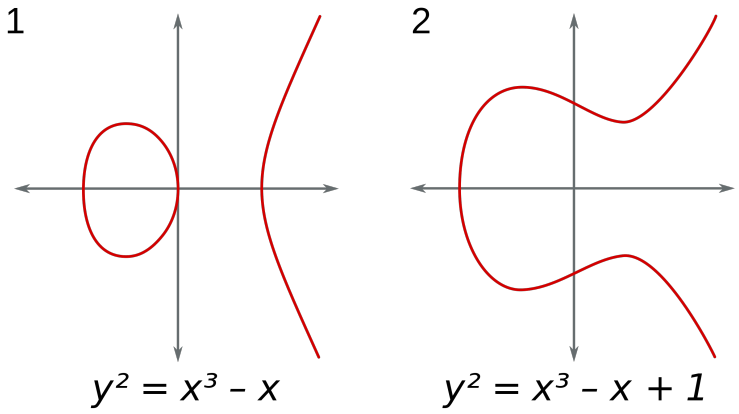


Figure 1: two examples of elliptic curves over $\mathbb{R}$

source: `https://commons.wikimedia.org/wiki/File:ECClines-3.svg`

licence: `https://creativecommons.org/licenses/by-sa/3.0/legalcode`, no changes were made.

Remark 25. More precisely, the set of rational points is described by the projective solutions (that is, in $\mathbb{P}^2(k)$) of the homogeneous equation

$$Y^2Z = X^3 + AXZ^2 + BZ^3.$$

In particular, one sees that the affine points are in bijection with the projective solutions with $Z \neq 0$ and the point at infinity $\mathcal{O}$ is the unique projective solution with $Z = 0$ (recall that the point with coordinates $X = Y = Z = 0$ is not on part of the projective plane).

Our interest in elliptic curves is motivated by the following Theorem.

Theorem 26. *Let E be an elliptic curve over k. Then $E(k)$ is an abelian group with unit element $\mathcal{O}$. The inverse of $P = (x_P, y_P)$ is given by $-P = (x_P, -y_P)$. Given two points P and Q (distinct from $\mathcal{O}$), the point $R = P + Q$ is given by the following algorithm. If $Q = -P$ then return $R = \mathcal{O}$. Otherwise put*

$$\lambda = \begin{cases} \dfrac{y_P - y_Q}{x_P - x_Q} & \text{if } P \neq Q \\ \dfrac{3x_P^2 + A}{2y_P} & \text{if } P = Q \end{cases}$$

and return $R = (x_R, y_R)$, where $x_R = \lambda^2 - x_P - x_Q$ and $y_R = \lambda(x_P - x_R) - y_P$. (Note : the fraction symbol here means division in the field k.)

Proof. The underlying "geometric" process is described in Figure 4, using the short Weierstrass equation as model of the curve. Given two points P and Q on E (over k) let L be the line through P and Q, or the tangent line at P if $P = Q$. By Bezout theorem, since E is non-singular, L intersects E

on a third point R, counting with multiplicities and with the understanding that $\mathcal{O}$ lies on each vertical line. Now we define the sum $P + Q$ as the reflection of R with respect to the x-axis.

In formulas, write $L : Y = \lambda X + \nu$, with $\nu = y_P - \lambda x_P$, that is, L is a line through P. Note that λ is as in the statement of the theorem. Substituting the equation of L in E we get

$$(\lambda X + \nu)^2 = X^3 + AX + B$$

so

$$X^3 - \lambda^2 X^2 + (A - 2\lambda\nu)X + (B - \nu^2) = 0.$$

We know that x_P and x_Q are roots of this equation. Denote by x_R the third root. Hence

$$X^3 - \lambda^2 X^2 + (A - 2\lambda\nu)X + (B - \nu^2) = (X - x_P)(X - x_Q)(X - x_R).$$

So, the coefficients of X^2 must be the same on both sides, that is

$$-\lambda^2 = -(x_P + x_Q + x_R),$$

which gives us the formula for x_R (and proves that $x_R \in k$). Now substituting in the equation of the curve, and changing the sign (due to the reflection), will give us the formula for y_R.

Therefore the addition is well defined. Proving the other group axioms is a long calculation, which we leave to the reader. □

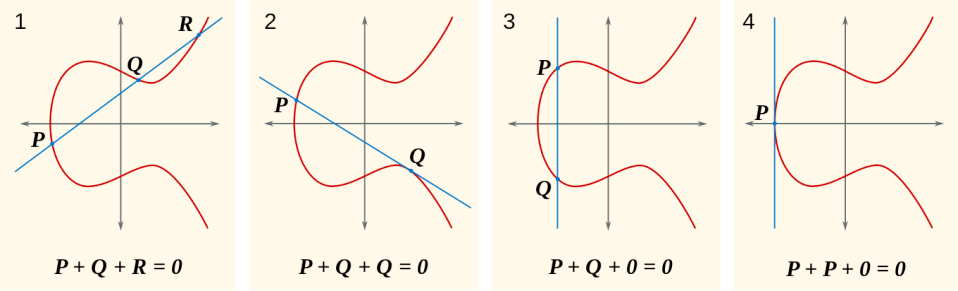


Figure 2: Addition law (over $\mathbb{R}$)

source: `https://commons.wikimedia.org/wiki/File:ECClines.svg`

licence: `https://creativecommons.org/licenses/by-sa/3.0/legalcode`, no changes were made.

Exercise 27. Let $E : Y^2 = X^3 + AX + B$ be an elliptic curve over a field k. Let $P = (x_P, y_P)$ be a point of E over the algebraic closure $\bar{k}$ of k. Prove that P has order exactly 3 if and only if x_P is a root of the following polynomial

$$3x^4 + 6Ax^2 + 12Bx - A^2.$$

This polynomial is called the 3-division polynomial of E.

5 Elliptic Curves over finite fields

We will now focus on finite fields. For simplicity we will focus on primes fields $\mathbb{F}_p$, where p is a prime number. Observe that the group $E(\mathbb{F}_p)$ is finite. So a natural question is: "What is the size of $E(\mathbb{F}_p)$?"

Assuming that $p \geq 3$ and $E/\mathbb{F}_p$ is given by

$$E : Y^2 = X^3 + AX + B,$$

for each $x \in \mathbb{F}_p$ denote by $Q(x)$ the quantity (x^3+Ax+B). If $Q(x) = 0$ then there is only one point P on E with $x_P = x$, namely $P = (x, 0)$. If $Q(x)$ is a square in $\mathbb{F}_p$, say $Q(x) = y^2$ then there are two points with X-coordinate equal to x, which are $P = (x, y)$ and $-P = (x, -y)$. If $Q(x)$ is not a square, then there is no such point. Since half of the elements of $\mathbb{F}_p^*$ are squares, we get a rough estimate:

$$\#E(\mathbb{F}_p) \sim p + 1.$$

A much more precise, and meaningful, statement is the following Theorem. The proof requires a bit of algebraic geometry so we will just refer to [Sil09, Theorem V.1.1].

Theorem 28 (Hasse's Theorem)**.** *Given an elliptic curve E over $\mathbb{F}_p$ there exists and integer t, known as the* trace of Frobenius*, such that*

$$\#E(\mathbb{F}_p) = p + 1 - t.$$

Moreover, we have $|t| \leq 2\sqrt{p}$.

Exercise 29. (a) For which primes p does the equation

$$Y^2 = X^3 + 2X + 3$$

define an elliptic curve over $\mathbb{F}_p$?

(b) Consider the elliptic curve

$$E : \ Y^2 = X^3 + X + 3$$

over $\mathbb{F}_5$. Compute the number of points on E. Is the group $E(\mathbb{F}_5)$ cyclic?

(c) Can you find an elliptic curve over $\mathbb{F}_7$ with 15 points?

From the previous theorem we deduce that it is enough to know the value of t in order to know the value of $\#E(\mathbb{F}_p)$.

Theorem 30. *The value of t can be computed in polynomial time.*

There quite a few algorithms to perform this task that use various methods. In Section 5.1 we give an overview the first one to be proposed.

Knowing the size of $E(\mathbb{F}_p)$ is almost sufficient to determine the group structure of $E(\mathbb{F}_p)$.

Theorem 31 ([Was08, Theorem 4.1])**.** *Let E be an elliptic curve over $\mathbb{F}_p$ then either*

$$E(\mathbb{F}_p) \simeq \mathbb{Z}/n\mathbb{Z},$$

for some integer $n > 0$, or,

$$E(\mathbb{F}_p) \simeq \mathbb{Z}/n_1\mathbb{Z} \times \mathbb{Z}/n_2\mathbb{Z},$$

for positive integers $n_1 | n_2$.

5.1 Schoof's algorithm

The first polynomial-time algorithm to count points on an elliptic curve was proposed in 1985 by Schoof. Here is an outline of the algorithm, which has a running time of $\mathcal{O}(\log^6 p)$. This section requires some knowledge of algebraic geometry to be fully understood.

The underlying idea is to compute the value of

$$t_l = t \bmod l$$

for primes $l \leq L$ such that $\prod_{l \leq L} l > 4\sqrt{p}$ (for simplicity we exclude $l = p$). Then we can use the Chinese Reminder Theorem to compute the value

$$t \bmod \left(\prod_{l \leq L} l \right),$$

which by Theorem 28 equals the value of t.

The Frobenius endomorphism π of E acts on the l-torsion points $E[l]$ of E (defined over the algebraic closure $\bar{\mathbb{F}}_p$). An important fact is that

$$E[l] \simeq \mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z},$$

that is, it is a two-dimensional vector space over $\mathbb{F}_l$. If we denote by π_l the restriction of π to $E[l]$ then it satisfies a relation in $\mathrm{End}(E[l])$ of the form

$$\pi_l^2 - t_l \pi_l + q_l \cdot \mathrm{id}_l = 0,$$

where $q_l = q \bmod l$ and id_l is the restriction of the identity. Hence it suffices to compute π_l and id_l in $\mathrm{End}(E[l])$ and formulas to perform addition and multiplication and then solve the previous equation to get t_l.

The key point is that there is a polynomial $h \in \mathbb{F}_p[x]$, called the *l-division polynomial* of E, satisfying

$$P = (x_P, y_P) \in E[l] \iff h(x) = 0.$$

Moreover h can be computed in $\mathcal{O}(l^2 \log p)$ time. See also Exercise 27.

We can then represent elements of $\mathrm{End}(E[l])$ (as rational functions) by elements of the polynomial ring:

$$R_l := \frac{\mathbb{F}_p[x, y]}{(h(x), y^2 - Ax^3 - B).}$$

In particular,

$$\begin{aligned}\pi_l &= \left(x^p \bmod h(x), y^p \bmod (h(x), y^2 - Ax^3 - B)\right) \\ &= \left(x^p \bmod h(x), \left((Ax^3 + B)^{(p-1)/2}\right) y\right)\end{aligned}$$

and

$$\begin{aligned}\mathrm{id}_l &= \left(x \bmod h(x), y \bmod (h(x), y^2 - Ax^3 - B)\right) \\ &= \left(x \bmod h(x), (1 \bmod h(x))\, y\right).\end{aligned}$$

For non-zero elements $\alpha_1 = (a_1(x), b_1(x)y)$ and $\alpha_2 = (a_2(x), b_2(x)y)$ in $\mathrm{End}(E[l])$ we have

$$\text{Multiplication: } \alpha_1 \circ \alpha_2 = (a_1(a_2(x)), b_1(a_2(X))b_2(x)y)$$

and

$$\text{Addition: } \alpha_3 = \alpha_1 + \alpha_2 = (a_3(x), b_3(x)y)$$

with

$$\begin{aligned}a_3 &= r^2(Ax^3 + B) - a_1 - a_2 \\ b_3 &= r(a_1 - a_3) - b_1\end{aligned}$$

where $r = (b_1 - b_2)/(a_1 - a_2)$ when $\alpha_1 \neq \alpha_2$ and $r = (3a_1 + A)/2b_1(Ax^3 + B)$ when $\alpha_1 = \alpha_2$.

Remark 32. Not all elements of $\mathrm{End}(E[l])$ have the form $(a(x), b(x)y)$, but we are only interested in π_l^2, π_l and id_l.

Remark 33. In computing r we might find a non-zero divisor, when computing the inverses. If this is the case we can use it to find a non-trivial divisor g of h and replace h with g or h/g and work in a smaller ring!

Later Schoof's algorithm has been improved by Elkies and Atkin, leading to what is usually known as *SEA* algorithm which has estimated running time $\mathcal{O}(\log^4 p)$. The improvement consists of choosing the primes l carefully in such a way that one can replace the division polynomial h with one of smaller degree.

5.2 ECDLP

Since we now know how to compute the size of $E(\mathbb{F}_p)$, we can easily produce a point P generating (a subgroup of) $E(\mathbb{F}_p)$ (and deduce the group structure of $E(\mathbb{F}_p)$). This means that we have all the ingredients to "translate" Diffie-Hellman and ElGamal to elliptic curves.

Exercise 34. Write down tables for the Elliptic Curves Diffie-Hellman key exchange and the Elliptic Curves ElGamal Public Key Cryptosystem analogous to Tables 1 and 2.

Exercise 35. Put $p = 89$ and consider the elliptic curve $E : Y^2 = X^3 + 2X + 3$ over $\mathbb{F}_p$. Denote by P the point $(75, 41)$ on E, which has order 42. Alice and Bob want to communicate using the Elliptic Curve ElGamal Public Key Cryptosystem. Alice chooses as a secret key the integer $n_A = 33$.

(a) What is Alice's public key Q_A?

Bob wants to send to Alice the message $M = (51, 71) \in E(\mathbb{F}_p)$ and chooses as secret ephemeral key the integer $k = 8$.

(b) What is the cipher-text (C_1, C_2) that Bob will send to Alice.

You will probably find the following table useful.

n	nP
4	(19,74)
6	(35,36)
9	(64,80)
16	(71,86)
25	(51,18)
32	(17,12)

The security of these two protocols is then based on the hardness of the *elliptic curve discrete logarithm problem*, in short ECDLP.

Theorem 36. *The fastest known algorithm to solve the ECDLP on an elliptic curve $E/\mathbb{F}_p$ is fully exponential, with running time $\mathcal{O}(\sqrt{p})$.*

In particular we can use Shank's Baby-Step-Giant-Step and Pohlig-Hellmann described in Section 3.

Exercise 37. Consider the following elliptic curve defined over $\mathbb{F}_p$, with $p = 17$:

$$E : Y^2 = X^3 + 16X - 4$$

Put $P = (0, 8)$ and $Q = (7, 14)$. What is the order of P? Solve the ECDLP

$$xP = Q.$$

You will probably find the following table useful.

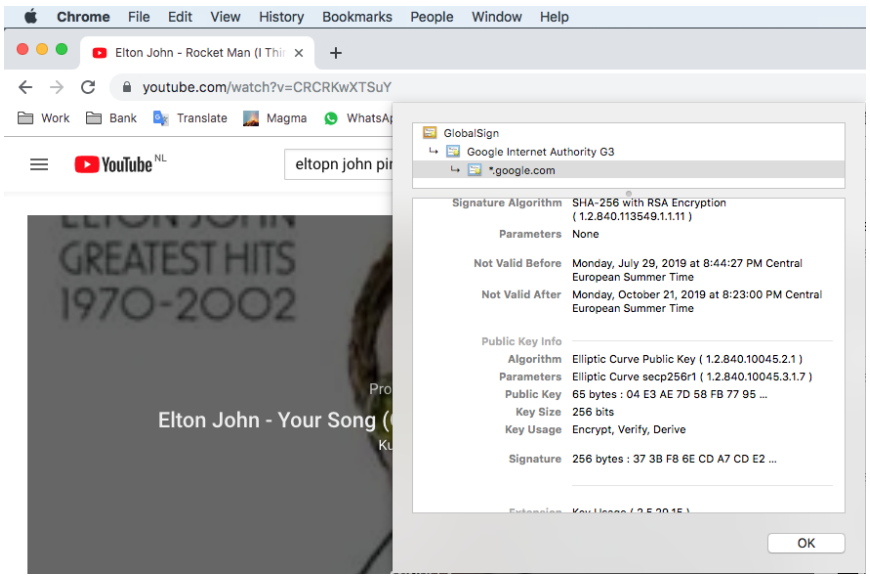
n	nP
5	$(13, 15)$
6	$(6, 11)$
9	$(10, 0)$
14	$(2, 6)$
15	$(16, 8)$

5.3 Brief history of ECC

The hardness of the ECDLP should be compared with cryptosystems based on the integer factorization, like the RSA, for which there are sub-exponential time algorithms (eg. the Number Field Sieve or Lenstra's Factorization method), or the DLP on $\mathbb{F}_p^*$ which can be solved in time which is less or equal than sub-exponential (eg. Index Calculus).

This implies that in order to achieve the same security level, one can use smaller parameters (in this case, the coefficients of the curve). This makes the use of Elliptic Curves Cryptography (ECC) preferable since it requires less resources to encrypt and decrypt.

Nowadays ECC is widely used:



But this was not always the case. In the last 40 years ECC has seen quite a few antagonists.

- first decade (1985-1995): the RSA foundation strongly opposed ECC because, at the time, Elliptic curves were considered "esoteric mathematics" by the crypto-community. This was not entirely false: RSA is based on integer factorization which is a problem that has been studies

for centuries (if not millennia), while EC have been under the investigative eyes of mathematicians for much less time! Nevertheless in December 1995 the The National Security Agency (NSA) of the U.S. decisively gave support to ECC. Maybe it is a good point to remind you that NSA and the equivalent agencies of the other countries are leading their research secretly and it is possible that more is known to them than what is known to the public. Nevertheless one can try to guess what do they known by looking at their recommendations and the protocol used by government agencies. So it is not entirely clear why NSA went against the suggestions of the RSA foundation (which is private!) but perhaps it was because of the high patenting fees asked by the RSA foundation.

- second decade (1995-2005): a lot of more research is done on ECC. Some families of "weak" elliptic curves were found. No big deal for ECC, since these are easy to identify and avoid. The NSA supports for ECC increased, as their recommendations to move to ECC. Nevertheless the transition was slow. Even if one can factor integers in subexponential time and ECDLP requires full exponential time algorithms to be broken, it was easier to just increase the size of the keys and keep using the same algorithms, rather than changing completely the protocols.

- third decade (2005-2015): not much happens in the first half. But in 2013 Edward Snowden leaks a lot of confidential documents about US government agencies. On September 5, 2013, *The New York Time* reported than these documents showed that the NSA has put a *backdoor* in of the standardized version of one algorithm based on the ECDLP (the Dual EC Deterministic Random Bit Generator, to be precise). Moreover, on the 20 December of 2013, *Reuters* reported also that the NSA has paid the RSA foundation a secret 10 million dollar payment to include such version of the Dual EC DRBG in one of their toolkits. Roughly speaking, in this algorithm one needs to pick two points P and Q which are supposed to be "independent" one from each other in order to guarantee security. But the points recommended in the standardized version of Dual EC DRPG where satisfying a relation $P = kQ$. Knowing the solution k of this ECDLP, the NSA was effectively able to break-in the security. It is important to point out that the algorithm in question was very anomalous and already in 2007 the possibility of such a back-door was brought-up. Maybe as a consequence of this fact, in 2015 the NSA released a report claiming the "need to move to Post Quantum Cryptography".

- It has been known since 1994 that both integer factorization and DLP can be broken in quantum-polynomial-time by Shor's algorithm

[Sho94]. On the other hand the current (2018) record of integer-factorization on a quantum device is 4088459, which was achieved using IBM's 5-qubit processor. This number has 22 bits, so we are very far from the current standards for RSA, which recommends to use integers of (at least) 1024 bits for commercial communications.

- Even assuming that quantum computers will become a reality soon (which is not believed to happen at least until 2030) we will need to move to different algorithms that are quantum-secure. So is ECC condemned? No! Indeed one of the proposed quantum-secure algorithms is based on the so-called "isogeny problem" between supersingular elliptic curves over $\mathbb{F}_{p^2}$.

An interesting reading, containing speculations about the motivation and choice of the timing for the above mentioned announcement by the NSA, is [KM16] "Koblitz, Menezes - A riddle wrapped in an enigma", which can be found online at `https://eprint.iacr.org/2015/1018.pdf`.

References

[ElG85] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Inform. Theory **31** (1985), no. 4, 469–472. MR 798552

[HPS14] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman, *An introduction to mathematical cryptography*, second ed., Undergraduate Texts in Mathematics, Springer, New York, 2014. MR 3289167

[KM16] Neal Koblitz and Alfred Menezes, *A riddle wrapped in an enigma*, IEEE Security and Privacy **14** (2016), no. 6, 34–42.

[Sho94] Peter W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, 35th Annual Symposium on Foundations of Computer Science (Santa Fe, NM, 1994), IEEE Comput. Soc. Press, Los Alamitos, CA, 1994, pp. 124–134. MR 1489242

[Sil09] Joseph H. Silverman, *The arithmetic of elliptic curves*, second ed., Graduate Texts in Mathematics, vol. 106, Springer, Dordrecht, 2009. MR 2514094

[Was08] Lawrence C. Washington, *Elliptic curves*, second ed., Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2008, Number theory and cryptography. MR 2404461

S. Marseglia, Mathematics Department, Utrecht University, P.O. Box 80010, 3508 TA Utrecht, The Netherlands

E-mail: s.marseglia@uu.nl