École Polytechnique Fédérale de Lausanne

Semester project

Schoof-Elkies-Atkin point counting algorithm for cryptographic purposes - Python Sage implementation

Student:
Aleksa Stankovic

Mentors:
Prof. Arjen Lenstra
Benjamin Wesolowski

Lausanne, January 12, 2017

Abstract

In this paper an implementation of Schoof-Elkies-Atkin algorithm for counting points on elliptic curve over finite field of characteristic larger than 3 is presented. An overview of elliptic curve cryptography and mathematical background leading to the algorithm is presented. Schoof's algorithm was discussed, as well as major ideas behind Schoof-Elkies-Atkin algorithm. Required modifications allowing usage of Atkin modular polynomials instead of classical ones are explained. Results of Sage implementation are commented as well.

Acknowledgements

I would like to thank professor A. Lenstra for finding very interesting and challenging topic which introduced me to modern research in cryptography.

I am also very thankful to Benjamin Wesolowski, for supervising my work and helping me understand more involved topics. His support was of invaluable significance for completing this project.

Contents

1 Introduction

In this paper we investigate implementation of the fastest known algorithm for counting the number of rational points on elliptic curves. The code developed in this project will be used for checking cryptographic safety of generated curves, and therefore we restrict ourselves to analyzing the curves defined over finite field $\mathbb{F}_p$, with p being large.

To the best knowledge of the author, this is the first open source SAGE implementation of the SEA point counting algorithm. The algorithm was developed for double-checking the correctness of existing implementations, as well as in hope that it will be useful for studying interesting aspects of the algorithm and trying out different upgrades. Author especially believes that due to the fact that the code is easy to read and maintain, it will provide good basis for further research.

This paper is organized as following. In section 2 we give basic definitions and theorems, and show how elliptic curves can be used in cryptography on an example of ElGamal cryptosystem. Then in section 3, we proceed with outlining some more theory behind elliptic curves, and explain first polynomial time algorithm for point counting. After that, in section 4, we look at the theory of elliptic curves from another perspective, using tools of complex analysis. Finally, in section 5, we introduce the reader to the ideas behind Schoof-Elkies-Atkin algorithm, show how it can be coded, and discuss its implementation. Furthermore, we discuss the special case of using Atkin modular curves instead of originally proposed classical modular curves, which give us significant improvement in performance. Finally, in section 6 we discuss the scope of the work done, and propose directions for further research.

2 Basic definitions and properties

We start this chapter by giving a definition of an algebraic curve which can be realized in a two-dimensional space. We use a symbol K to denote a field, and $\overline{K}$ to denote an algebraic closure of K.

Definition 2.1. A **plane algebraic curve** over a field K is a set of points

$P = (x, y) \in \overline{K} \times \overline{K}$ satisfying

$$p(x, y) = 0, \tag{1}$$

where p is a polynomial in variables x and y ($p \in K[x, y]$).

The definition of an algebraic curve can be also given in higher dimensions. These objects are known as skew-curves. For a curve that is lying in n-dimensional affine space we need at least $n - 1$ polynomials in n variables, and the polynomials must generate a prime ideal of Krull dimension 1. But for cryptographic purposes study of plane algebraic curves is sufficient. In fact, we restrict ourselves to a more specific setting.

Definition 2.2. On a plane algebraic curve over a field K, defined by the polynomial $p \in K[x, y]$, a point $P = (x, y) \in \overline{K} \times \overline{K}$ is called **singular** if and only if its coordinates x and y satisfy

$$p(x, y) = p_x(x, y) = p_y(x, y) = 0, \tag{2}$$

where p_x and p_y are formal partial derivatives of polynomial p over variables x and y. Curve that has no singular points is called **non-singular**.

Furthermore, we are interested only in polynomials p that are quadratic in y and cubic in x. Major objects we will be working with are given by the following definition:

Definition 2.3. An **elliptic curve** E over a field K, written as E/K, is a non-singular, plane algebraic curve whose points $P = (x, y) \in \overline{K} \times \overline{K}$ satisfy the following equation

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{3}$$

where $a_1, a_2, a_3, a_4, a_6 \in K$. We usually equate elliptic curve with the equation (3), which is called Weierstrass equation.

From definition 2.3 we se that we use notation E/K to denote that elliptic curve has implicit equation over $K[x, y]$, but we consider points to be in algebraic closure $\overline{K} \times \overline{K}$. Sometimes we want just to consider points $P = (x, y)$ with $x, y \in L$, where L is a field extension $K \subseteq L \subseteq \overline{K}$. Hence, we give a following definition:

Definition 2.4. Let E/K be an elliptic curve over a field K, and let L be a field such that $K \subseteq L \subseteq \overline{K}$. Then, an point $P = (x, y)$ on the curve is called **L-rational point** if and only if $x, y \in L$. Furthermore, we formally define another point O_E to be L-rational, which we call point at the infinity. The set of all **L-rational points** is denoted as $E(L)$.

The logic behind defining the point at infinity can be seen if we look at elliptic curves as a projective varieties. For this, we see the curve as a set of points in the projective n-dimensional space $(x, y, z) \in P^n(\overline{K})$ that satisfy following equation

$$zy^2 + a_1zxy + a_3z^2y = x^3 + a_2x^2z + a_4xz^2 + a_6z^3. \tag{4}$$

In case $z \neq 0$, by changing the variables as $(x, y, z) = (\frac{x}{z}, \frac{y}{z}, 1)$ we get the same equation (3). The only different point in $P^n(\overline{K})$ we get is when $z = 0$, in which case the only solution is a point $P = (0, 1, 0)$, which we denote with 0_E.

Let us now try to simplify equation (4). We begin by defining isomorphisms between the elliptic curves:

Definition 2.5. Let L be a field extension $K \subseteq L \subseteq \overline{K}$. Two elliptic curves E_1/K and E_2/K are **isomorphic** over L(or **L-isomorphic**) if and only if there exists a mapping between $(x_1, y_1) \in E_1/K$ and $(x_2, y_2) \in E_2/K$ defined with:

$$x_1 \to u^2x_2 + r, \quad y_1 \to u^3y_2 + u^2sx_2 + t, \quad 0_{E_1} \to 0_{E_2}, \tag{5}$$

where $(r, s, t, u) \in L^3 \times L^*$.

From now on, we assume that characteristic p of the underlying field K is prime to 6. Let us look at isomorphism

$$L(x, y) \to (x, y + \frac{1}{2}(a_1x + \frac{a_3}{2})), \tag{6}$$

Then, a point (x, y) belongs to E if and only if $(x_1, y_1) = L(x, y)$ belongs to E_1 defined by :

$$y_1^2 = x_1^3 + \frac{b_2}{4}x_1^2 + \frac{b_4}{2}x_1 + \frac{b_6}{4}, \tag{7}$$

where $b_2 = a_1^2+4a_2$, $b_4 = 2a_4+a_1a_3$ and $b_6 = a_3^2+4a_6$. Next, the isomorphism

$$L'(x_1, y_1) \to (x_1 + \frac{b_2}{12}, y_1), \tag{8}$$

maps points $(x_1, y_1) \in E_1$ to $(x_2, y_2) = L'(x_1, y_1)$ in a curve E_2 given by

$$y_2^2 = x_2^3 - \frac{c_4}{48}x_2 - \frac{c_6}{864}, \tag{9}$$

where $c_4 = b_2^2 - 24b_4$ and $c_6 = -b_2^3 + 36b_2b_4 - 216b_6$.

Thus, every elliptic curve over K is isomorphic to a curve with implicit equation:

$$y^2 = x^3 + ax + b, \qquad \text{with } a, b \in K. \tag{10}$$

Unfortunately, the representative of a class of isomorphic curves given in (10) is not unique. It is easy to see that for any $u \in K^*$ elliptic curve in (10) is isomorphic to a curve:

$$y^2 = x^3 + au^4x + bu^6. \tag{11}$$

Anyway, there is a way to assign a value to equation (10) that will by shared by all isomorphic curves and thus introduce some form of classification. Before we state how the curves can be classified, we introduce a curve that is closely related to the one defined as in (10).

Definition 2.6. Let E/K be an elliptic curve defined over a field K by Weierstrass equation (10) and let the characteristic of K be relatively prime to 6. Then, an elliptic curve E^d/K defined by Weierstrass equation

$$dy^2 = x^3 + ax + b, \tag{12}$$

where $d \in K$ is a non-square element of K is called quadratic twist of curve E/K.

Equation (13) can be equivalently written as

$$y^2 = x^3 + ad^2x + bd^3. \tag{13}$$

Curves E/K and E^d/K are not isomorphic over K, but they are isomorphic over field extension $K(\sqrt{d})$, and thus also over algebraic close $\overline{K}$. Now we define a value that all isomorphic curves or their quadratic twists have in common.

Definition 2.7. Let E/K be an elliptic curve defined over field K by Weierstrass equation (10) and let K be a field of characteristic that is relatively prime to 6. Then we assign to E/K a value

$$j_E = 1728\frac{4a^3}{4a^3 + 27b^2}, \tag{14}$$

which is referred to as j-invariant of a curve.

Importance of j-invariant can be seen from the following theorem:

Theorem 1. Let K be a field with characteristic relatively prime to 6. Then, all isomorphism classes of elliptic curves E/K are, up to twists, uniquely determined by the absolute invariants j_E, and for every $j \in K$ there exists elliptic curve with absolute invariant j.

If K is algebraically closed then the isomorphism classes of elliptic curves over K correspond one-to-one to the elements of K via the map $E \to j_E$.

Now that we simplified implicit curve equation, let us show how we can assign a group structure to the set $E(L), K \subseteq L \subseteq \overline{K}$. First, we define the point at infinity O_E to be neutral element of the group. Next, for $P_1, P_2 \neq O_E, P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ we define $P_1 \oplus P_2 = P_3(x_3, y_3)$ as:

$$\begin{aligned}
\lambda &= \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2, \\ \frac{3x_1^2 + a}{2y_1} & \text{if } x_1 = x_2, y_1 = y_2 \end{cases} \\
x_3 &= \lambda^2 - x_2 - x_1, \\
y_3 &= \lambda(x_1 - x_3) - y_1.
\end{aligned} \tag{15}$$

Finally, in case $x_1 = x_2$ and $y_1 \neq y_2$, we must have $y_2 = -y_1$, and in this case we define $P_3 = P_1 \oplus P_2$ to be neutral element 0_E. Therefore, the inverse element of $P_1 = (x_1, y_1)$ can be written as :

$$-P_1 = (x_1, -y_1), \tag{16}$$

and we can thus define subtraction operation in elliptic curve group as

$$P_1 \ominus P_2 = P_1 \oplus (-P_2). \tag{17}$$

Elliptic curve assigned with thusly defined group structure forms commutative group. The structure $E(L)$ can also be seen as a module over $\mathbb{Z}$. To show this, we define for some $n \in \mathbb{N}$ and some $P \in E(L)$, the product nP to be

$$nP = P \oplus P \oplus ... \oplus P,$$

where we have $n-1$ additions in the previous equation. In case n is negative, we define nP as $nP = (-n)P$. This gives us the mapping

$$[n] : E(L) \to E(L), \qquad [n](P) = nP, \tag{18}$$

which will be extensively used in later chapers.

2.1 Elliptic curves in cryptography - ElGamal encryption

Group structure makes elliptic curves interesting for cryptographic purposes. In abstract sense, the encryption procedure can be seen as a mapping between space of possible messages[1] $A = \{0, 1\}^*$, $f : A \to A$. It is desirable that for a message $m \in A$ encryption procedure is easy, so calculation of $b = f(m)$ is fast, but finding m given only knowledge of b is very hard (computationally expensive). The setting and properties of such functions can be formalized, and a function satisfying these is known as one-way function [14, ch. 2.1-2.3].

Existence of such a mapping is still an open question. Actually, finding at least one function would prove that the complexity classes NP and P are not equal. While existence of such functions is still not confirmed or refuted, there have been numerous proposals for functions f that have been extensively studied and are considered to be good candidates for use in cryptographic systems.

The most common cryptosystems base complexity of finding an inverse of their one-way functions on either a problem of factoring a large integer or on a problem of computing discrete logarithm in finite cyclic groups. Factoring large numbers presents a basis of celebrated RSA cryptosystem [20]. But for us of interest will be cryptosystems that base their security on difficulty

[1] Set A is here defined as a set containing all finite messages encoded as zeros and ones.

of finding discrete logarithm, such as ElGamal cryptosystem[10] and Diffie-Hellman protocol [7, 9].

Discrete logarithm problem can be described as following. Let us assume that we are given finite group G and elements $P, Q \in G$ such that $Q \in \langle P \rangle$, where $\langle P \rangle \subseteq G$ is a cyclic finite group generated by P. Then, the problem of finding a discrete logarithm consists in determining the integer n such that $nP = Q$.

The difficulty of solving discrete logarithm problem depends significantly on the representation of the group G or equivalently $\langle P \rangle$, as noted by Avanzi [1]. For example, in case G is additive group of $\mathbb{Z}_l$, the problem of discrete logarithm is equivalent to finding a multiplicative inverse of $P \in \mathbb{Z}_l^*$, which can be done very fast. Better choice for underlying structure G would be a multiplicative group $\mathbb{Z}_l^*$. Actually, the research and experience has found that carefully chosen elliptic curves provide very good level of security. The fastest known approach for solving the discrete logarithm problem when $G = E(\mathbb{F}_p)$ is Pollard-Rho algorithm [18], which is exponential in the size of input, as discussed in [16, 17]. In comparison, big integers can be factored in sub-exponential time by general number field sieve algorithm (see for example [19]). This means that cryptosystems based on elliptic curves can in fact use smaller key sizes than the ones basing their security on integer factorization schemes.

To illustrate this, we provide a comparison of key sizes between:

- FFC - finite field cryptography, such as Diffie-Hellman key exchange and DSA,
- IFC - integer factorization cryptography, such as RSA, and
- ECC - elliptic curve cryptography, such as ElGamal scheme and elliptic curve digital signature algorithm (ECDSA).

In table 1, taken from [3], we can spot the difference between the schemes. From here, we can see that elliptic curve cryptography has significant advantage over other systems and should be preferred at least for implementation in machines that have limited hardware resources.

Now, let us describe one popular method of choice for secure communication between two peers based on elliptic curve cryptography, that is known

Bits of security	FFC	IFC	ECC
80	1024 for public, 160 for private	1024	160-223
112	2048 for public, 224 for private	2048	224-255
128	3072 for public, 256 for private	3072	256-383
192	7680 for public, 384 for private	7680	384-511
256	15360 for public, 512 for private	15360	512+

Table 1: Comparison of key sizes for FFC - finite field cryptography, IFC - integer factorization cryptography, and ECC - elliptic curve cryptography.

as ElGamal cryptosystem. As it is customary in the literature, we assume that the communication is happening between two peers named Alice and Bob. The algorithm can be explained for any cyclic group G, so we will develop our discussion in this abstract setting, keeping in mind that usual choice for G is $G = E(\mathbb{F}_p)$, for the reasons explained before.

First step in the algorithm is generation of the keys. We assume that Alice wants to initiate secure communication, so that any information being sent to her is encrypted in such a way that only she can decrypt it. For this, she needs to generate keys and explain the encryption procedure to the outer world. She also needs to have private key that will be crucial for decryption procedure. This phase is completed in following steps:

- Alice choses cyclic group G of order q and a generator $g \in G$.
- Alice choses $x \in \{1, 2, \ldots, q-1\}$. Chosen x is saved as a private key and not revealed to the public.
- Alice computes $h = g^x$.
- Alice publishes G, g, q and h to the public.

The published information is enough for encryption, that can be done by Bob in the following steps:

- Bob choses random $y \in \{1, 2, \ldots, q-1\}$, and calculates $c_1 = g^y$.
- Bob calculates shared secret $s \in G$ as $s = h^y = (g^x)^y = g^{xy}$.

- Bob maps data m he wants to send by injective function to group G. The mapped information is denoted with m'.
- Bob calculates $c_2 = m' \cdot s$.
- Bob sends the ciphertext (c_1, c_2) to Alice over potentially insecure channel.

It is also important to remark that new random y needs to be generated every time this procedure is called, since malicious user can easily find h^y in case he knows m'. The decryption procedure works as follows:

- Alice calculates shared secret $s = c_1^x = (g^y)^x = g^{xy}$.
- Then Alice can calculate $m' = c_2 \cdot s^{-1}$. Inverse of s in G can be calculated efficiently in at most $O(\log(q))$ operations. From m', Alice can then reconstruct sent information and get m.

We finish the section by a remark that elliptic curve must be chosen very carefully for achieving wanted level of security. In case this is not accomplished, various subexponential time algorithms for solving discrete logarithm problem can be applied that could compromise security of the scheme. Some of the interesting ways to exploit vulnerability can be found in [2] and [11]. One of the concerns is that the order of elliptic curve group will be divisible by small primes. Ideally, we want to pick a curve which group will be cyclic, i.e. its cardinality is a very big prime. In particular, given a curve with its coefficients, we might ask ourselves how can we efficiently compute its order. The answer will be given in the following chapters.

3 Schoof's algorithm

Schoof's algorithm is the first polynomial time algorithm for counting the number of points on an elliptic curve $E/\mathbb{F}_p$. In the following two sections we give theoretical introduction to the necessary concepts, and then proceed and explain the algorithm in the final part of this chapter.

3.1 Torsion group

Definition 3.1. Let E/K be an elliptic curve and let $n \in \mathbb{N}$. Kernel of the mapping $[n]$

$$E[n] = \{P \in E(\overline{K}) \mid [n]P = O_E\} \tag{19}$$

is called n-torsion group. Elements $P \in E[n]$ are called n-torsion points.

Following theorem shows that torsion groups have very simple structure, which makes them suitable for use in algorithms.

Theorem 2. Let $E/\mathbb{F}_p$ be an elliptic curve over a finite field $\mathbb{F}_p$, and let n be an integer relatively prime to p. Then, the n-torsion group $E[n]$ satisfies

$$E[n] \cong \mathbb{Z}_n \times \mathbb{Z}_n. \tag{20}$$

For latter study we will also need following theorem:

Theorem 3. Let $E/\mathbb{F}_p$ be an elliptic curve over a finite field $\mathbb{F}_p$. Then, for a prime $\ell \neq p$, ℓ-torsion group $E[\ell] = \mathbb{Z}_\ell \times \mathbb{Z}_\ell$ has exactly $\ell + 1$ cyclic groups $C_i, i = 0, ..., \ell$, and for every i we have $|C_i| = \ell$.

Proof. From theorem 2 we know that $E[\ell]$ is generated by two points $P_0, P_1 \in E[\ell]$ of order ℓ. Now, let us define $P_i = [i-1]P_0 \oplus P_1, i = 2, 3, ..., \ell$ and show that $\langle P_i \rangle$ are all cyclic subgroups of $E[\ell]$.

First, by using the fact that elliptic curve group is commutative, we have that

$$\ell P_i = \ell((i-1)P_0 \oplus P_1) = \ell(i-1)P_0 \oplus \ell P_1 = 0_E, \tag{21}$$

and thus $|\langle P_i \rangle| = \ell$. Let us now show $\langle P_i \rangle \cap \langle P_j \rangle = \{0_E\}$ for $i \neq j$. We argue by reduction to absurd. If there are two $i, j \in \{0, 1, \ldots, \ell\}, i \neq j$ such that $\langle P_i \rangle \cap \langle P_j \rangle = P \neq 0_E$, then

$$\begin{aligned} P = aP_i + bP_j &= a((i-1)P_0 \oplus P_1) \oplus b((j-1)P_0 \oplus P_1) \\ &= (a(i-1) + b(j-1)P_0) \oplus (a+b)P_1. \end{aligned} \tag{22}$$

Now, because P_0 and P_1 are linearly independent, we have that

$$\begin{aligned} a(i-1) + b(j-1) &\equiv 0 \pmod{\ell}, \\ a + b &\equiv 0 \pmod{\ell}, \end{aligned} \tag{23}$$

from where directly follows $i \equiv j \pmod{\ell}$. Finally, in case there exists any other cyclic group C different from C_i, it will be generated by a point $P \in E[\ell]$. But since it is easy to see $\bigcup_{i=0}^{\ell} C_i = E[\ell]$, we know that $P \in C_i$ for some $i = 0, ..., \ell$, and thus $\langle P \rangle = C_i$. □

3.2 Division polynomials

Definition 3.2. For $m \in \mathbb{N}_0$, we define m**-th division polynomial** $\psi_m \in \mathbb{Z}[x, y, a, b]$ by a recurrent formula:

$$\begin{aligned}
\psi_0 &=0,\\
\psi_1 &=1,\\
\psi_2 &=2y,\\
\psi_3 &=3x^4 + 6ax^2 + 12bx - a^2,\\
\psi_4 &=4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3), \qquad (24)\\
&\vdots\\
\psi_{2m} &= \left(\frac{\psi_m}{2y}\right)(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2), \qquad m \geq 3,\\
\psi_{2m+1} &=\psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3, \qquad m \geq 2.
\end{aligned}$$

Let us comment that we also need to make sure that ψ_{2m} is well defined by our definition. For this, the numerator in equation for ψ_{2m} needs to be divisible by $2y$. This is indeed true, and can be easily proven by induction.

Division polynomials have close connection with the elliptic curve $E/\mathbb{F}_p$ given by implicit equation $y^2 = x^3 + ax + b$, as can be seen from the following theorem:

Theorem 4. Let m be a nonnegative integer, and let $E/\mathbb{F}_p$ be an elliptic curve given by Wierstrass equation (10). Also, let us denote with $f(x) = x^3 + ax + b$ the right hand side of Wierstrass equation. Now, we define with f_m a polynomial in the ring $\mathbb{F}_p[x, y]/\langle y^2 - f(x)\rangle$ by

$$f_m(x, y) = \begin{cases} \psi_m(x, y) & \text{if } m \text{ odd},\\ \psi_m(x, y)/(2y) & \text{if } m \text{ even}. \end{cases} \qquad (25)$$

Then, the polynomials $f_m(x, y)$ are well defined, and have following properties:

- f_m depends only on x.
- The degree of f_m in x is smaller or equal than $(m^2 - 1)/2$ for m odd, and smaller or equal than $(m^2 - 4)/2$ for m even. In case p is relatively prime to m for m odd, or relatively prime to $m/2$ for m even, the bounds are exact.
- Let $P \neq 0_E$ be a point on elliptic curve $E/\mathbb{F}_p$. Then $P = (x, y) \in E[m]$ if and only if $f_m(x) = 0$.

Proof. The fact that polynomials are well defined can be proven by induction. For this, we assert that ψ_m belongs to $\mathbb{F}_p[x]/\langle y^2 - f(x)\rangle$ for $m \leq 4$. Then, using recurrence relations (24) and fact that $y^2 = f(x) \pmod{y^2 - f(x)}$ we have that for $m > 4$:

$$f_{2m+1} = \begin{cases} f_{m+2}f_m^3 - 16f(x)^2 f_{m-1}f_{m+1}^3 & \text{if } m \text{ odd,} \\ 16f(x)^2 f_{m+2}f_m^3 - f_{m-1}f_{m+1}^3 & \text{if } m \text{ even,} \end{cases} \tag{26}$$

and

$$f_{2m} = f_m(f_{m+2}f_{m-1}^2 - f_{m-2}f_{m+1}^2). \tag{27}$$

Thus, by induction we see that f_m is well defined and also by induction we can see that it depends only on x. The second and third point are resulting from the construction of division polynomials. The proof of these points can be found in [15, pp. 33-34]. □

We see that all points in $E[n]$ can be characterized by division polynomials, since they can be expressed as elements of the set $E[n] = 0_E \bigcup \{(x, y) \in \overline{\mathbb{F}}_p \times \overline{\mathbb{F}}_p \mid y^2 - f(x) = 0, f_n(x) = 0\}$. Division polynomials have another nice property, as it is shown in the following theorem:

Theorem 5. Let $P = (x, y)$ be a point on the elliptic curve E/K given by the implicit equation $E : y^2 = x^3 + ax + b$. Then, for $n \in \mathbb{N}$ and $P = (x, y) \notin E[n]$ we have

$$[n]P = \left(x - \frac{\psi_{n-1}\psi_{n+1}}{\psi_n^2}, \frac{\psi_{n+2}\psi_{n-1}^2 - \psi_{n-2}\psi_{n+1}^2}{4y\psi_n^3}\right) \tag{28}$$

Proof. We refer to [15] for details. □

Now, with this knowledge we can proceed to the next section, in which main algorithm of this chapter is explained.

3.3 Schoof's algorithm

In this section we provide a quick description of the Schoof's algorithm. For more detailed discussion, we refer to [5] and [21].

Definition 3.3. Let $E/\mathbb{F}_p$ be an elliptic curve defined over finite field $\mathbb{F}_p$. Then the mapping:

$$\phi_p : E/\mathbb{F}_p \to E/\mathbb{F}_p, \quad \phi_p(x,y) = (x^p, y^p), \quad \phi_p(0_E) = 0_E, \tag{29}$$

is called Frobenius endomorphism.

It is trivial to prove that Frobenius mapping acts as an automorphism on $E/\mathbb{F}_p$. The relationship between Frobenius endomorphism and $|E(K)|$ can be seen from the following theorem:

Theorem 6. Let $E/\mathbb{F}_p$ be an elliptic curve with p being a prime and $p > 3$. Then, Frobenius endomorphism ϕ_p defined on $E/\mathbb{F}_p$ satisfies following equation:

$$\phi_p^2 \ominus [t]\phi_p \oplus [p] = 0_E. \tag{30}$$

The number $t \in \mathbb{Z}$ is called the trace of Frobenius endomorphism, and it satisfies

$$|E(\mathbb{F}_p)| = p + 1 - t. \tag{31}$$

From this theorem we see that determining $|E(K)|$ is equivalent to finding the trace of Frobenius endomorphism. The value of t is bounded with:

$$|t| \leq 2\sqrt{p}, \tag{32}$$

which is known as Hasse bound. But even under this bound, value $|t|$ of elliptic curves used in cryptography is too large for direct computation. The strategy of Schoof's and SEA algorithm is to compute value of t modulo small primes ℓ, and then to reconstruct t using Hasse bound and Chinese remainder theorem. From now on, value $t \pmod{\ell}$ will be denoted as t_ℓ.

We first explain how we can find value of t_ℓ for $\ell = 2$. In this case, since $|E(\mathbb{F}_p)| = p + 1 - t$, and because we assume $p > 3$, we know that t will be even if and only if $|E(\mathbb{F}_p)|$ is even. And $|E(\mathbb{F}_p)|$ is even if and only if it has a point of order two. This happens only if there is a point $P = (x, 0), x \in \mathbb{F}_p$ which satisfies Wierstrass equation. This is then equivalent to polynomial $f(x) = x^3 + ax + b$ having a root in $\mathbb{F}_p$, which be quickly checked with $\gcd(x^3 + ax + b, x^p - x) \neq 1$.

Now we assume that $\ell > 2$ and explain how we treat this case.

Theorem 7. Let $E/\mathbb{F}_p$ be an elliptic curve over finite field $\mathbb{F}_p$, with $p > 3$. Then, the reduction of Frobenius endomorphism to ℓ-torsion group $E[\ell]$ with $\ell \neq p$ and ℓ being a prime satisfies:

$$\phi_p^2 \ominus [t_\ell]\phi_p \oplus [p_\ell] = 0_E, \tag{33}$$

where $t_\ell = t \pmod{\ell}$ and $p_\ell = p \pmod{\ell}$.

From here, the strategy is to find t_ℓ by checking for which t_ℓ equation (33) holds for all points in $E[\ell]$. The idea of Schoof is to write point $P \in E[\ell]$ as $P = (x, y)$, where x and y are not known beforehand, and use the fact that $f_\ell(x) = 0$ and $y^2 - x^3 - ax - b = 0$ for such chosen (x, y). Then, we can use formulas for additions in group, leave x and y as variables, and work with polynomials in $\mathbb{F}_p[x, y]/\langle y^2 - f(x), f_\ell(x)\rangle$. Finally, for some unique $t_\ell \in \mathbb{F}_\ell$, equation (33) will reduce to trivial one in this ring.

Let us give a bit more details on this procedure. First, we rewrite equation (33) as

$$\phi_p^2 \oplus [p_\ell] = [t_\ell]\phi_p. \tag{34}$$

As we have said, we start by assuming that we are working with some generic non-trivial point $P \in E[\ell], P = (x, y)$. Since the left hand side of equation (34) does not depend on t_ℓ, we can calculate it before we start a search for values of t_ℓ. First, we need to compute $\phi_p^2(x, y) = (x^{p^2}, y^{p^2})$ in $\mathbb{F}_p[x, y]/\langle y^2 - f(x), f_\ell(x)\rangle$. Since all polynomials occuring in this discussion are members of $\mathbb{F}_p[x, y]/\langle y^2 - f(x), f_\ell(x)\rangle$, we will always assume that the results of our calculations will be of form $(r_1(x), yr_2(x))$, where $r_1(x), r_2(x)$ are rational functions, because every time we encounter y^2 we can change it with $f(x)$. Thus, we find $\phi_p^2(x, y) = (x^{p^2}, y^{p^2}) = (p_1(x), yp_2(x))$ by calculating:

$$\begin{aligned} p_1(x) &\equiv x^{p^2} && \pmod{f(x)}, \\ p_2(x) &\equiv x^{(p^2-1)/2} && \pmod{f(x)}. \end{aligned} \tag{35}$$

These calculations can be done in $O(\log(p))$ operations by using square-and-multiply algorithm. Next, we calculate the point $[p_\ell](x, y)$ by using division polynomials :

$$[p_\ell](x, y) = \left(x - \frac{\psi_{p_\ell - 1}\psi_{p_\ell + 1}}{\psi_\ell^2}, \frac{\psi_{p_\ell + 2}\psi_{p_\ell - 1}^2 - \psi_{p_\ell - 2}\psi_{p_\ell + 1}^2}{4y\psi_{p_\ell}^3} \right). \tag{36}$$

We can see that for $p_\ell = 1$, equation (36) is not well defined. But in this case we have that $[1](x, y) = (x, y)$. So, we can assume we have calculated:

$$\phi_p^2(P) = (p_1(x), yp_2(x)), \qquad [p_\ell]P = \left(\frac{g_1(x)}{h_1(x)}, y\frac{g_2(x)}{h_2(x)} \right), \tag{37}$$

with $p_1, p_2, g_1, g_2, h_1, h_2 \in \mathbb{F}_p[x]/\langle f_\ell \rangle$. Now, we differ between three possible cases:

- $\phi_p^2(P) \oplus [p_\ell](P) = 0_E$. By using definition of inverse in an elliptic curve group, we see that this happens if

  $$p_1(x)h_1(x) = g_1(x), \quad p_2(x)h_2(x) + g_2(x) = 0. \tag{38}$$

  In this case, we obviously set $t_\ell = 0$.

- $\phi_p^2(P) \ominus [p_\ell](P) = 0_E$. This case can be recognized by

  $$p_1(x)h_1(x) = g_1(x), \quad p_2(x)h_2(x) = g_2(x). \tag{39}$$

  In this case, we have that $\phi_p^2(P) \oplus [p_\ell](P) = [p_\ell](P) \oplus [p_\ell](P) = [2p_\ell](P)$, and thus from (33) we have $[t_\ell]\phi_p(P) = [2p_\ell](P)$, or equivalently $\phi_p(P) = \left[\frac{2p_\ell}{t_\ell}\right](P)$. Then, finding the value t_ℓ is equivalent to finding an eigenvalue of Frobenius endomorphism $\lambda = \frac{2p_\ell}{t_\ell} \in \mathbb{F}_\ell$. Since we have already computed (x^p, y^p), all we need to check now is which

λ satisfies $\phi_p(P) = (x^p, y^p) = [\lambda](P) = [\lambda](x, y)$, by checking for which λ following equation is satisfied:

$$[\lambda](x,y) = \left(x - \frac{\psi_{\lambda-1}\psi_{\lambda+1}}{\psi_\lambda^2}, \frac{\psi_{\lambda+2}\psi_{\lambda-1}^2 - \psi_{\lambda-2}\psi_{\lambda+1}^2}{4y\psi_\lambda^3} \right) = (x^p, y^p). \quad (40)$$

or equivalently:

$$\begin{aligned} &(x - x^p)\psi_\lambda^2 - \psi_{\lambda-1}\psi_{\lambda+1} = 0, \\ &\psi_{\lambda+2}\psi_{\lambda-1}^2 - \psi_{\lambda-2}\psi_{\lambda+1}^2 - 4y^{p+1}\psi_\lambda^3 = 0, \end{aligned} \quad (41)$$

in $\mathbb{F}_p[x]/\langle f_\ell, y^2 - f(x)\rangle$.

- $\phi_p^2(P) \oplus [p_\ell](P) \neq 0_E$ and $\phi_p^2(P) \ominus [p_\ell](P) \neq 0_E$. Then, by using group law, we can calculate expression $\phi_p^2(P) \oplus [p_\ell](P) = \left(\frac{g_4(x)}{h_4(x)}, y\frac{g_3(x)}{h_3(x)}\right)$, and check if it equates $[t_\ell](P) = \left(x - \frac{\psi_{n-1}\psi_{n+1}}{\psi_n^2}, \frac{\psi_{n+2}\psi_{n-1}^2 - \psi_{n-2}\psi_{n+1}^2}{4y\psi_n^3}\right)$, by equating first and second coordinates, multiplying equality by denominators and checking it in $\mathbb{F}_p[x]/\langle f_\ell(x)\rangle$. Note again that we do not need to care for formal variable y, since second coordinate will be always expressed as $y \cdot r(x)$, with r being a rational function.

We remark here that for better performance all calculations can be done just for the first coordinate. When the match is found, we know that we have found a candidate τ for which it is either $\tau = t_\ell$ or $\tau = \ell - t_\ell$. Then we can quickly check whether equality holds for second coordinate as well, in which case we accept τ as a value of t_ℓ, and otherwise we take $t_\ell = \ell - \tau$.

Described algorithm is particularly important because it is the first polynomial time algorithm for counting points on elliptic curve. But, as noted by Schoof in [22] it is actually too slow for practical purposes. The bottleneck of the presented algorithm comes from the fact that degrees of the division polynomials ψ_ℓ grow very fast, and thus all computations in $\mathbb{F}_p[x]/\langle f_\ell(x)\rangle$ are very slow.

4 Complex analysis look at elliptic curves

Elliptic curves can be also analyzed by using techniques of complex analysis. This comes from the fact that elliptic curves defined over complex field $\mathbb{C}$ are in bijective correspondence with lattices. Here we briefly state results needed for developing SEA algorithm, while more thorough study of these relations is given in [24].

4.1 Lattices

First, we begin with a definition of lattice.

Definition 4.1. Let $w_1, w_2 \in \mathbb{C}$ be $\mathbb{R}-$linearly independent. Then the set $\Lambda = w_1\mathbb{Z} + w_2\mathbb{Z}$ is called a lattice.

Of fundamental importance for connecting theory of lattices with elliptic curves is Weierstrass $\wp$ function.

Definition 4.2. Let $\Lambda \subset \mathbb{C}$ be a lattice. Weierstrass elliptic function $\wp$ is defined by the series:

$$\wp(z, \Lambda) = \frac{1}{z^2} + \sum_{w \in \Lambda \setminus \{0\}} \frac{1}{(z-w)^2} - \frac{1}{w^2} \tag{42}$$

The connection of Weierstrass function with elliptic curves can be seen from the following theorem.

Theorem 8. Let $E/\mathbb{C}$ be an elliptic curve given by the Weierstrass equation (3). Then there exists a lattice $\Lambda \subset \mathbb{C}$ such that the map:

$$\mathbb{C}/\Lambda \to E/\mathbb{C}$$

$$z + \Lambda \to \begin{cases} (x_\Lambda, (\wp'(z) - a_1 x_\Lambda - a_3)/2) & \text{if } z \notin \Lambda, \\ 0_E \text{ if }, z \in \Lambda \end{cases} \tag{43}$$

where $x_\Lambda = \wp(z) - \frac{a_1^2 + 4a_2}{12}$ is a bijection. Conversely, for every lattice Λ there is an unique curve $E/\mathbb{C}$ such that the above map exists.

Proof. We refer to [24, page 161]. □

In special case a curve is given in form (10) map from the previous theorem can be expressed as

$$z + \Lambda \to (\wp(z), \wp'(z)/2) \tag{44}$$

From the previous theorem we can see that to the every lattice Λ we can associate an elliptic curve $E/\mathbb{C}$ and vice-versa. Also, by investigating action of isomorphisms on the group of $E/\mathbb{C}$ we can prove the following result:

Theorem 9. Let Λ, Λ' be two lattices, and $E_\Lambda, E_{\Lambda'}$ elliptic curves corresponding to them. Then, the elliptic curves $E_\Lambda, E_{\Lambda'}$ are isomorphic if and only if $\Lambda = c\Lambda'$ for some nonzero $c \in \mathbb{C}$.

Proof. Proof can be found in [24, page 161]. □

Since it is enough to look at the class of isomorphic curves for counting the points rather than at one curve specifically, we might be interested to find a canonical representation of a lattice corresponding to the class of isomorphic curves. It is not hard to prove that using homogeneous transformation $z \to cz$ class of isomorphic curves can be uniquely represented by a lattice $\Lambda = \mathbb{Z} + \tau\mathbb{Z}$, where $\tau \in \mathcal{H} = \{z \in \mathbb{C} \mid \operatorname{Im}(z) > 0\}$.

In (14) we showed how to define j-invariant in function of coefficients from elliptic curve equation. Here we see that the class of isomorphic curves can be also defined with a lattice $\Lambda_\tau = \mathbb{Z} + \tau\mathbb{Z}$. We will now give a definition of j-invariant by using lattice representation of elliptic curve. For this, we define $\mathbb{Z}[[q]]$ to be a ring of formal Laurent series over $\mathbb{Z}$, and define in this ring following series:

$$\begin{aligned} &E_2(q) = 1 - 24\sum_{n=1}^{\infty}\frac{nq^n}{1-q^n}, E_4(q) = 1 + 240\sum_{n=1}^{\infty}\frac{n^3q^n}{1-q^n}, \\ &E_6(q) = 1 - 504\sum_{n=1}^{\infty}\frac{n^5q^n}{1-q^n}. \end{aligned} \tag{45}$$

Then, j-invariant can be defined equivalently in given complex lattice $\Lambda = \mathbb{Z} + \tau\mathbb{Z}$ by the formula:

$$j(q) = 1728\left(\frac{E_4^3(q)}{E_4^3(q) - E_6^2(q)}\right) = \frac{1}{q} + 744 + 196884q + 21493760q^2 + ..., \tag{46}$$

where $q = e^{2i\pi\tau}$.

4.2 Isogenies

In this chapter we define mappings between elliptic curves that respect group operations. We give only basic definitions and theorems. For more detailed study we refer to Silverman's textbook [24].

Elliptic curves can be seen as algebraic varieties, meaning that they are solutions to a set of polynomial equations. On the set over which algebraic varieties are defined we can introduce Zariski topology, in which all sets that are solutions to a system of a polynomial equations are closed, and open sets are defined as complements to these sets.

Definition 4.3. A map between two algebraic varieties $f : X \to Y$ is **regular** at a point $x \in X$ if there is a neighborhood $U \ni x$ and neighborhood $V \ni f(x)$, such that $f(U) \subseteq V$ and $f : U \to V$ is polynomial mapping. The mapping f is called **regular** if and only if it is regular in every point $x \in X$.

In other words, if we look at the elliptic curves as manifolds we say that a mapping between two elliptic curves is regular if and only if it can be represented as polynomial in local coordinates. Now, we can define a class of regular functions of particular importance:

Definition 4.4. An **isogeny** between elliptic curves E/K and E'/K is a regular map $\psi : E/K \to E'/K$ that is either non-constant and maps $0_E \to 0_{E'}$ or zero-morphism $\psi(P) \to 0_{E'}, P \in E/K$. Two elliptic curves are called *isogenous* if an isogeny between them exists. *Degree* of isogeny ψ can be defined as the size of its kernel.

Some of the isogenies we already encountered are curve isomorphisms. Other isogenies are maps $[n], n \in \mathbb{N}$. Following theorem gives a connection between isogenies and group structure of elliptic curve:

Theorem 10. Let E/K and E'/K be isogenous elliptic curves over K under some isogeny ψ. Then ψ is also a group homomorphism between $E(K)$ to $E'(K)$.

Proof. Proof can be found in [24, pages 75-76]. □

Isogeny between elliptic curves defines an equivalence relation. Also, in case an isogeny $\psi : E/K \to E'/K$ is of degree n, then there exists dual isogeny $\tilde{\psi} : E'/K \to E/K$ such that $\tilde{\psi} \circ \psi = [n]$ on E/K and $\psi \circ \tilde{\psi} = [n]$ on E'/K. In this case we say that E/K and E'/K are n-isogenous, and this defines equivalence relation between elliptic curves as well.

Class of n-isogenous elliptic curves can be studied with so called classical modular curves. One of the equivalent ways to define classical modular curve is the following:

Definition 4.5. Let $n \in \mathbb{N}$. Then, n-th **classical modular curve** is an irreducible plane algebraic curve given by equation

$$\Phi_n(x, y) = 0, \tag{47}$$

such that $(x, y) = (j(q^n), j(q))$, and $j(q)$ is j-invariant of some elliptic curve.

Following theorem gives relation between elliptic curves defined over $\mathbb{C}$, and n-isogenies:

Theorem 11. Let $E/\mathbb{C}$ and $E'/\mathbb{C}$ be two elliptic curves with j-invariants j_E and $j_{E'}$. Then $\Phi_n(j_E, j_{E'}) = 0$ if and only if there exists isogeny from $E/\mathbb{C}$ to $E'/\mathbb{C}$ whose kernel is cyclic of degree n.

In case an elliptic curve is $E/\mathbb{F}_p$, and given ℓ prime and $\ell \neq p$, we have that $E[\ell] = \mathbb{F}_\ell \times \mathbb{F}_\ell$. Then, the zeros of polynomials $\Phi_\ell(x, j_E)$ will be exactly j-invariants of $\ell + 1$ cyclic subgroups of $E[\ell]/C_i, i = 0, 1, ..., \ell$.

5 Schoof-Elkies-Atkin algorithm

Schoof-Elkies-Atkin (SEA) algorithm is an improvement of the Schoof's algorithm. The algorithm was developed by carefully analyzing the characteristic equation of Frobenius endomorphism $\chi_\ell(\phi_p) = \phi_p^2 \ominus [t_\ell]\phi_p \oplus [p_\ell] = 0_E$. First observation is that $E[\ell]$ has module structure $\mathbb{Z}_\ell \times \mathbb{Z}_\ell$, and that $\phi_p : E[\ell] \to E[\ell]$ can be seen as a linear mapping.

Let us denote with λ and μ the eigenvalues of characteristic polynomial $\chi_\ell(\phi_p)$. Then, we have that $t_\ell = \mu + \lambda$ and $p_\ell = \mu\lambda$. Idea of SEA algorithm is to analyze the characteristic equation to see if polynomial $\chi_\ell(\phi_p)$ can be factored, and use this information to calculate t_ℓ. To begin, we give the following definition:

Definition 5.1. Let $\Delta = t_\ell^2 - 4p_\ell \in \mathbb{F}_\ell$ be a discriminant of the characteristic polynommial of Frobenius endomorphism $\chi_\ell(\phi_p)$, where Frobenius endomorphism is defined over some curve $E/\mathbb{F}_p$ and ℓ is a prime and $\ell \neq p$. If Δ is a square in $\mathbb{F}_\ell$, we call ℓ Elkies prime. Otherwise, we call ℓ Atkin prime.

We see that whether a prime ℓ can be considered Elkies or Atkin depends on the choice of the curve $E/\mathbb{F}_p$. Unfortunately, characteristic equation $\chi_\ell(\phi_p)$ can not be directly used for determining whether ℓ is Elkies or Atkin since we don't know the value of t_ℓ beforehand. For this we need to use modular curves, which are given in section 4.2, in a theorem that is one of the crucial parts of the Schoof-Elkies-Atkin algorithm:

Theorem 12 (Atkin)**.** Let E be an ordinary elliptic curve defined over $\mathbb{F}_p$ with j-invariant $j \neq 0$ and $j \neq 1728$. Let $\Phi_\ell(x, j) = h_1 h_2 \cdots h_s$ be the factorization of $\Phi_\ell(x, j) \in \mathbb{F}_p[x]$. Then there are following possibilities for degrees of polynomials $h_1, h_2, ..., h_s$:

1. $(1, 1, \ldots, 1)$ or $(1, l)$. In either case $t^2 - 4p \equiv 0 \pmod{l}$. In the first case we set $r = 1$, and in the second $r = l$.

2. $(1, 1, r, r, r, \ldots, r)$. In this case $t^2 - 4p$ is a square modulo l, r divides $l - 1$, and ϕ_p acts as a diagonal matrix $\begin{bmatrix} \lambda & 0 \\ 0 & \mu \end{bmatrix}$

3. $(r, r, \ldots, r)$ for some $r > 1$. In this case $t^2 - 4p$ is a non-square in $\mathbb{F}_l$, r divides $l + 1$.

In any case, r is the order of ϕ_p in the projective general linear group $PGL_2(\mathbb{F}_l)$, and the trace of Frobenius satisfies

$$t^2 \equiv p(\zeta + 2 + \zeta^{-1}) \pmod{l}, \tag{48}$$

where ζ is primitive r-th root of unity in $\overline{\mathbb{F}}_l$. The number of irreducible factors s satisfies

$$(-1)^s = \left(\frac{p}{l}\right). \tag{49}$$

In case we encounter first or second case, the prime ℓ is Elkies, and otherwise it is Atkin. For checking which of these options is true it is enough to calculate $\gcd(\Phi_\ell(x, j), x^p - x)$ to find whether $\Phi_\ell(x, j)$ has any roots in $\mathbb{F}_p$. The procedure for treating Atkin and Elkies primes is entirely different, so we describe them independently in the next two sections.

5.1 Treatment of Atkin primes

In case ℓ is Atkin prime, from the theorem 12 we know that

$$t^2 \equiv p(\zeta + 2 + \zeta^{-1}) \pmod{l}, \tag{50}$$

where ζ is r-th root of unity in $\overline{\mathbb{F}}_\ell$. Let us prove now that actually $\zeta \in \mathbb{F}_{\ell^2}$. From Atkin's classification theorem we know that $\Phi_\ell(x, j)$ is of degree $\ell+1$ in x, and because the Atkin case is the third from the theorem we have $r|\ell+1$, or differently $rs = \ell + 1$ for some $s \in \mathbb{N}$. But then $\zeta^{\ell^2-1} = \zeta^{(\ell-1)(\ell+1)} = \zeta^{rs(\ell-1)} = 1$, and therefore $\zeta \in \mathbb{F}_{\ell^2}$.

All possible r-th roots of unity in $\mathbb{F}_{\ell^2}$ can be calculated in the following way. Let g be generator of the multiplicative group $\mathbb{F}^*_{\ell^2}$. Then $\zeta_0 = g^{(l^2-1)/r}$ is r-th primitive root of unity. Other primitive roots can be found as $\zeta_i = \zeta_0^i$, where i is relatively prime to r and $1 \leq i < r$. Finite field $\mathbb{F}_{\ell^2}$ is isomorphic to $\mathbb{F}_\ell(\sqrt{d})$, where $\sqrt{d}$ is non-square in $\mathbb{F}_\ell$. Since for SEA algorithm we usually use values of $\ell < 200$, we can precompute values of d and g. Finally, from equation (48) we can generate all possible candidates for t_ℓ^2, by calculating $p(\zeta + 2 + \zeta^{-1})$. Note that we need only first half of the values ζ_i, since all others will be exactly inverses of the first half. Then, for those candidates that are square[2] in $\mathbb{F}_\ell$ (there must be at least one such candidate), we solve the quadratic equation (50) to get solution using Tonelli-Shanks algorithm [23, 26]. Then, from this we form the list T_ℓ containing all the possible values of t_ℓ.

Unfortunately, there exists no procedure that can reduce number of the candidates even further. Anyway, the information gathered in such a way is still useful, since it decreases number of possibilities for t_ℓ, and thus for t as well. Also, the procedure in case ℓ is Atkin prime is very fast compared to the Elkies case, and thus gathering information in this way does not hurt the performance very much.

To conclude this section, let us discuss how we can find r efficiently. Since we are in Atkin case, classification theorem gave us $\gcd(\Phi_\ell(x, j), x^p - x) = 1$. From that theorem we also know that $\Phi_\ell(x, j)$ splits to s polynomials of order r. Thus, all the roots of polynomial $\Phi_\ell(x, j)$ belong to $\mathbb{F}_{p^r}$, and r is

[2] Checking whether the candidates are square can be done very fast using Legendre symbol.

smallest number having this property. Therefore, to find r we can compute $\gcd(\Phi_\ell(x,j), x^{p^i} - x)$ until we find i such that $\gcd(\Phi_\ell(x,j), x^{p^i} - x) = x^{p^i} - x$. Since calculating $\gcd(\Phi_\ell(x,j), x^{p^i} - x))$ is very expensive, we can use intermediate results $\gcd(\Phi_\ell(x,j), x^{p^i})$ to compute $\gcd(\Phi_\ell(x,j), x^{p^{i+1}})$ as :

$$\gcd(\Phi_\ell(x,j), x^{p^{i+1}}) = \gcd(\Phi_\ell(x,j), (x^{p^i})^p). \tag{51}$$

Another way number r can be found is with a help of Berlekamp matrices. Using them we can count the number of irreducible factors of polynomial $\Phi_\ell(x,j)$, which we have denoted s, and then r can be calculated with $r = (l+1)/s$.

In order to calculate s, we first compute the matrix $Q = (q_{i,j})_{i,j=0}^{\ell}$ in which the coefficients $q_{i,j} \in \mathbb{F}_p$ are found as

$$x^{ip} \equiv q_{i,0} + q_{i,1}x + \ldots + q_{i,\ell}x^\ell \pmod{\Phi_\ell(x,j)}. \tag{52}$$

Note that once the $x^p \pmod{\Phi_\ell(x,j)}$ is known, the matrix Q can be constructed in $O(\ell)$ polynomial multiplications and reductions $(\bmod\ \Phi_\ell(x,j))$, by iteratively computing $x^p, x^{2p} \equiv x^p \cdot x^p, x^{3p} = x^{2p} \cdot x^p$ and so on. Now, the number of irreducible factors s can be found as $s = \ell + 2 - m$, where m is rank of the matrix $(Q - I)$, and I is a diagonal matrix.

Assuming that $\gcd(\Phi_\ell(x,j), x^p)$ is already calculated, the first approach (using fast arithmetic) has complexity of $O(r \log(p)^2 \ell \log(\ell))$, while the second can be executed in $O(\ell^3 \log(p))$. Actually, the second method is much faster because typically $r \sim \ell/2$, and fast arithmetic is interesting only for large values of $\log(p)$.

5.2 Treatment of Elkies primes

In case ℓ is Elkies prime, we saw that in $E[\ell]$ Frobenius endomorphism reduces as:

$$(\phi_p - [\lambda])(\phi_p - [\mu]) = 0,$$

for some $\lambda, \mu \in \mathbb{F}_\ell$. Since λ, μ are eigenvalues of Frobenius endomorphism, there exists some point $P \in E[\ell]$ such that $\phi_p(P) \in \langle P \rangle$, and thus $\phi_p(\langle P \rangle) = \langle P \rangle$. In Elkies case we aim to find one of the eigenvalues λ or μ, and then to calculate the other by using $p_\ell = \lambda\mu$. Finally, we will find t_ℓ by exploiting $t_\ell = \lambda + \mu$.

Now, the question is how we can find any of the eigenvalues. Since they are symmetric, from now on we will assume we are searching for eigenvalue μ. We know that in some subgroup $C = \langle P \rangle$ equation $\phi_p - [\mu] = 0$ holds. Then, idea of Elkies step is to check this equation for a subgroup C in a similar way as in Schoof's step, but instead of $\ell^2 - 1$ relevant points of $E[\ell]$ we can check this equation only for $\ell - 1$ points of C. Let us denote with $g_\ell(x)$ the polynomial whose roots are exactly x-coordinates of non-trivial points in C. Then, since division polynomial ψ_ℓ has roots that are x-coordinates of all non-trivial points in $E[\ell]$, we necessarily have that $g_\ell | \psi_\ell$. Furthermore, because first $(\ell - 1)/2$ points of C will have the same x-coordinates as the second $(\ell - 1)/2$, the polynomial g_ℓ will actually be of degree $(\ell - 1)/2$.

Now, remaining question we need to answer is how to compute polynomial g_ℓ in an efficient way. For this, we first note that C is a subgroup of $E[\ell]$ and remark that C is actually a kernel of some ℓ-isogeny φ between $E/\mathbb{F}_p$ and some $\tilde{E}/\mathbb{F}_p$. From theorem 11 we know that j-invariant of $\tilde{E}/\mathbb{F}_p$ is actually a root of polynomial $\Phi_\ell(x, j)$. Let us denote j-invariant of curve $\tilde{E}/\mathbb{F}_p : y^2 = x^3 + \tilde{a}x + \tilde{b}$ with $\tilde{\jmath}$. Knowing the value of $\tilde{\jmath}$, let us show how to find coefficients $\tilde{a}, \tilde{b}$.

For this, we first define formal derivative of Laurent series $f(q) = \sum\limits_n a_n q^n$ to be $f'(q) = q\frac{\partial f}{\partial q} = \sum\limits_n n a_n q^n$. Then, we have following result:

Theorem 13. The following identities hold in $\mathbb{Z}[[q]]$:

$$\frac{j'}{j} = -\frac{E_6}{E_4}, \quad \frac{j'}{j - 1728} = -\frac{E_4^2}{E_6}, \quad \frac{j''}{j'} = \frac{1}{6}E_2 - \frac{1}{2}\frac{E_4^2}{E_6} - \frac{2}{3}\frac{E_6}{E_4} \tag{53}$$

Proof. Proof can be found in [22, page 244]. □

As noted in [22], values of $E_2(q), E_4(q), E_6(q)$ are actually elements in ring of integers $\mathcal{O}_K$, where K is some field. Then, there exist a prime ideal $\mathcal{B} \subseteq \mathcal{O}_K$ such that $\mathcal{O}_K/\mathcal{B} \cong \mathbb{F}_p$. From [22, page 245] we can state that:

$$E_4(q) \equiv -48a_4 \pmod{\mathcal{B}}, \qquad E_6(q) \equiv 864a_6 \pmod{\mathcal{B}}, \tag{54}$$

which gives us a connection between the value of $q = e^{2i\pi\tau}$ of some lattice Λ_τ, and coefficients for a Weierstrass equation of a curve. Previous relation works in case $j \neq 0, 1728$ as usual. Thus, with knowing the value of j and j' we can easily calculate coefficients E_4, E_6 from equation (53) and then by (54) we can proceed and calculate a_4 and a_6 as:

$$a_4 = -\frac{1}{48}\frac{j'^2}{j(j-1728)}, \qquad a_6 = -\frac{1}{864}\frac{j'^3}{j^2(j-1728)} \tag{55}$$

For ℓ-isogenous curve $\tilde{E}/\mathbb{F}_p$ we already have $\tilde{\jmath}$, and to get coefficients $\tilde{a}_4, \tilde{a}_6$ it is enough to find value of $\tilde{\jmath}'$. From now on, we shorten the notation by using Φ for Φ_ℓ, and denote with Φ_x and Φ_y partial derivatives of Φ with respect to the first and the second variable, respectively. Value $\tilde{\jmath}'$ can be found by differentiating equation $0 = \Phi(j, \tilde{\jmath}) = \Phi(j(q), j(q^\ell))$ to get :

$$j'\Phi_x(j, \tilde{\jmath}) + \ell\tilde{\jmath}'\Phi_y(j, \tilde{\jmath}) = 0, \tag{56}$$

from where we obtain $\tilde{\jmath}' = -j'\Phi_x(j, \tilde{\jmath})/(\ell\Phi_y(j, \tilde{\jmath}))$.

Now that we have constructed isogenous curve $\tilde{E}/\mathbb{F}_p$, we aim to find a polynomial g_ℓ that will describe a kernel of isogeny φ. Schoof in [22] found it easier to work with curve isomorphic to $\tilde{E}/\mathbb{F}_p$ given by $\hat{E}/\mathbb{F}_p : y^2 = x^3 + \hat{a}x + \hat{b}$, where $\hat{a} = \tilde{a}l^4, \hat{b} = \tilde{b}l^6$. Method for computation of polynomial g_ℓ was found by studying special relation between g_ℓ and Weierstrass elliptic function $\wp$. As before, let Λ_τ be a lattice associated to a curve E, and let reduced Weierstrass equation be given with

$$\wp(z) = \frac{1}{z^2}\sum_{n=1}^{\infty} c_n z^{2n}, \tag{57}$$

where coefficients c_n can by computed with:

$$c_1 = -\frac{a}{5}, \;\; c_2 = -\frac{b}{7}, \;\; c_n = \frac{3}{(n-2)(2n+3)}\sum_{j=1}^{n-2} c_j c_{n-1-j}, \quad \text{for } n \geq 3. \tag{58}$$

In case p is not very big, calculations of coefficients c_n might fail, in which case we need to turn to some other methods. But for SEA algorithm we are interested for p very large, and thus we can assume that these coefficients can be calculated.

The following theorem given by Elkies shows us how the coefficients of polynomial g_ℓ can be calculated:

Theorem 14. Let ψ be an ℓ-isogeny between curves E and $\hat{E}$ which has C as its kernel, and and let g_ℓ be a polynomial which roots are exactly x-coordinates of cyclic group C. Then:

$$z^{\ell-1} g_\ell(\wp(z)) = \exp\left(-\frac{1}{2}p_1 z^2 - \sum_{n=1}^{\infty} \frac{\hat{c}_n - \ell c_k}{(2n+1)(2n+2)} z^{2n+2}\right), \tag{59}$$

where Weierstrass elliptic function and coefficients c_n are as defined in (57) and (58), respectively, and p_1 is sum of all distinct x-coordinates points in C and can be calculated with

$$p_1 = \ell\left(\frac{1}{2}\left(\frac{j''}{j'} - \ell\frac{\tilde{\jmath}''}{\tilde{\jmath}'}\right) + \frac{1}{4}\left(\frac{E_4^2}{E_6} - \ell\frac{\tilde{E}_4^2}{\tilde{E}_6}\right) + \frac{1}{3}\left(\frac{E_6}{E_4} - \ell\frac{\tilde{E}_6}{\tilde{E}_4}\right)\right). \tag{60}$$

Value $\frac{j''}{j'} - \ell\frac{\hat{\jmath}''}{\hat{\jmath}'}$ can be evaluated by differentiating equation $\Phi(j, \hat{\jmath}) = 0$ two times. Doing this gives us:

$$\frac{j''}{j'} - \ell\frac{\tilde{\jmath}''}{\tilde{\jmath}'} = \frac{j'^2\Phi_{xx}(j, \tilde{\jmath}) + \ell j'\tilde{\jmath}'\Phi_{xy}(j, \tilde{\jmath}) + \ell^2\tilde{\jmath}'^2\Phi_{yy}(j, \tilde{\jmath})}{j'\Phi_x(j, \tilde{\jmath})}. \tag{61}$$

We also give pseudocode in which all steps necessary for generating polynomial g_ℓ are shown.

After calculating polynomial g_ℓ, we can proceed in fairly similar way as in Schoof's algorithm, only that now instead of division polynomial ψ_ℓ of degree $(\ell^2-1)/2$ we use much smaller polynomial g_ℓ of degree $(\ell-1)/2$. Also, instead of checking characteristic equation of Frobenius endomorphism, we check equation:

$$\phi_p(x, y) - [\mu](x, y) = 0$$

since it has simpler form and as we have assumed it is true for all elements $P = (x, y) \in C$.

Algorithm 1 Generation of Elkies modular polynomial(Source code provided by Andrew Sutherland)

```
Input: p; $a, b \in \mathbb{F}_p$; $\ell > 2, \ell$ prime, $j, \tilde{j} \in \mathbb{F}_p$.
Output: $\tilde{a}, \tilde{b}, g_\ell(x)$
Compute $\Phi_x, \Phi_x x, \Phi_y, \Phi_{yy}, \Phi_{xy}$.
Let $m = 18b/a$, $j' = mj$, and $k = j'/(1827 - j)$.
Let $\tilde{j}' = -j'\Phi_x/(\ell\Phi_y), \tilde{m} = \tilde{j}'/\tilde{j}$, and $\tilde{k} = \tilde{j}'/(1728 - \tilde{j})$
Define $\tilde{a} = \ell^4\tilde{m}\tilde{k}/48$ and $b = \ell^6\tilde{m}^2\tilde{k}/864$
Let $r = -(j'^2\Phi_{xx} + 2\ell j'\tilde{j}'\Phi_{xy} + \ell^2\tilde{j}'^2\Phi_{yy})/(j'\Phi_x)$
Define $p_1 = \ell(r/2 + (k - \ell\tilde{k})/4 + (\ell\tilde{m} - m)/3)$
Let $d = (\ell - 1)/2$
Let $t_0 = d, t_1 = p_1/2, t_2 = ((1 - 10d)a - \tilde{a})/30$, and $t_3 = ((1 - 28d)b - 42t_1a - \tilde{b})/70$
Let $c_0 = 0, c_1 = 6t_2 + 2at_0, c_2 = 10t_3 + 6at_1 + 4bt_0$
for $n = 2$ to $d - 1$ do
    Define $s = \sum_{i=1}^{n-1} c_i c_{n-i}$
    Let
        $$c_{n+1} = \frac{3s - (2n-1)(n-1)ac_{n-1} - (2n-2)(n-2)bc_{n-2}}{(n-1)(2n+5)}$$
end for
for $n = 3$ to $d - 1$ do
    Let
        $$t_{n+1} = \frac{c_n - (4n-2)at_{n-1} - (4n-4)bt_{n-2}}{4n+2}$$
end for
Let $s_0 = 1$
for $n = 1$ to $d$ do
    Let $s_n = \frac{-1}{n}\sum_{i=1}^{n}(-1)^i t_i s_{n-i}$
end for
return $g_\ell = \sum_{i=0}^{d}(-1)^i s_i x^{d-i}$
```

5.3 Putting the information together

After we have gathered enough information from Atkin and Elkies steps (which happens when $\prod \ell > 4\sqrt{p}$), we can proceed and calculate the trace of Frobenius endomorphism t. We do this by using Chinese remainder theorem, but we need to make small adjustments because from Atkin steps we have actually multiple candidates for value of t_ℓ.

Let us denote with m_E product of all processed Elkies primes. Using Chinese remainder theorem we can find $t_E \equiv t \pmod{m_E}$. For treating Atkin primes we take inspiration from Shanks baby-step/giant-step(BSGS) algorithm. First, let us denote with T_ℓ the list of all possible candidates for t_ℓ calculated for some Atkin prime ℓ. Then, we split all calculated sets T_ℓ into two collections of similar sizes, A_1 and A_2. When we say we want these collections to be of similar sizes, we actually ask for values:

$$s_1 = \prod_{T_\ell \in A_1} |T_\ell|, \qquad s_2 = \prod_{T_\ell \in A_2} |T_\ell|,$$

where $|T_\ell|$ is cardinality of T_ℓ, to be as close to each other as possible.

Furthermore, let us write with $m_1 = \prod\limits_{T_\ell \in A_1} \ell$ and $m_2 = \prod\limits_{T_\ell \in A_2} \ell$. Then we know that the trace of Frobenius endomorphism satisfies:

$$t \equiv t_1 \pmod{m_1}, \qquad t \equiv t_2 \pmod{m_2}, \tag{62}$$

for some t_1, t_2 that can be reconstructed from sets T_ℓ by using Chinese remainder theorem.

Because m_1 and m_2 are relatively prime we know that t can be represented as

$$t = t_E + m_E(m_1 r_2 + m_2 r_1), \tag{63}$$

for some $r_1, r_2 \in \mathbb{Z}$. Taking this equation modulo m_1 and m_2, we get

$$r_1 \equiv \frac{t_1 - t_E}{m_E m_2} \pmod{m_1}, \qquad r_2 \equiv \frac{t_2 - t_E}{m_E m_1} \pmod{m_2}. \tag{64}$$

Using t_1 and t_2 we can calculate candidates for r_1 and r_2. But this is not yet very useful for reconstructing the value t, because these are given only modulo m_1, m_2, and thus there are infinitely many possibilities for them. Following theorem will give us suitable bounds:

Theorem 15. Let us assume that t_E is such that $0 \leq t_E < m_E$, and $|r_1| \leq \lfloor \frac{m_1}{2} \rfloor$. Then we have that $|r_2| \leq m_2$.

Proof. From equation (63) we have that $r_2 = \frac{t - t_E - m_E m_2 r_1}{m_E m_1}$. Then, because $m_1 m_2 m_E > 4\sqrt{p}$ and $|t| \leq 2\sqrt{p}$ we have

$$\begin{aligned} |r_2| &\leq \frac{|t| + |t_E| + m_E m_2 |r_1|}{m_E m_1} \leq \frac{2\sqrt{p}}{m_E m_1} + \frac{1}{m_1} + \frac{m_2}{2} \\ &\leq \frac{m_2}{2} + \frac{1}{m_1} + \frac{m_2}{2} = m_2 + \frac{1}{m_1}. \end{aligned} \tag{65}$$

From here we have directly $|r_2| \leq m_2$, which concludes the proof. □

We know that the relation between order of a group $E(\mathbb{F}_p)$ and trace is given by $|E(\mathbb{F}_p)| = p + 1 - t$. Therefore, for any point $P \in E(\mathbb{F}_p)$ we have that

$$[p+1]P = [t]P = [t_E + m_E(m_1 r_2 + m_2 r_1)]P, \tag{66}$$

or, equivalently

$$[p + 1 - t_E - m_2 r_1]P = [m_E m_1 r_2]P. \tag{67}$$

Now, to see if given pair of candidates (r_1, r_2) is good we can check this equation for some random $P \in E(\mathbb{F}_p)$. We do this using BSGS strategy. First, after picking random point P we generate $[p+1-t_E]P$, from which we subtract all $[m_2 r_1]P$, where r_1 is chosen from equation (64) and $|r_1| < \lfloor \frac{m_1}{2} \rfloor$. The result of this step is a set of all possible values for left hand side of equation (67), which we store in data structure and keep the values sorted in order to allow for quick searching. Then, for every candidate r_2, we calculate the value of the right hand side $[m_E m_1 r_2]P$ and check if it matches any of the values in our sorted structure representing left hand side. In case of match, we can easily recover t using values of r_1 and r_2.

5.4 Using Atkin modular polynomials

Classical modular polynomials used for Atkin's classification theorem and for calculating isogeny in Elkies step have huge coefficients, which considerably impacts the performance of the algorithm. While they are originally proposed for SEA algorithm, research that followed introduced other more suitable alternatives. Popular choices are canonical modular polynomials, Atkin modular polynomials, or Webber modular polynomials. Fastest point counting was done with Webber polynomials [25]. These polynomials can be used only for special types of fields, and computation shows that curves defined over these fields have order divisible by two. As discussed before, we are interested in checking whether order of a curve is a large prime, and thus Webber polynomials are not applicable in our setting. Hence, we decide to use Atkin polynomials as common choice for various softwares implementing SEA algorithm (one of them being PARI/GP).

Atkin modular polynomials have same splitting properties as the classical modular polynomials, and thus they can be used in the same fashion for Atkin classification theorem as well as in the Atkin steps. Only difference we need to take care of is the calculation of invariant $\tilde{j}$ and derivatives related to it. For Atkin modular polynomial $\Psi(f, j)$, we have that:

$$\Psi(f, j) = 0, \tag{68}$$

where f is a solution of a polynomial equation and can be seen as a modular function. For such f, every ℓ-isomorphic curve will have invariant $\tilde{j}$ satisfying:

$$\Psi(f, \tilde{j}) = 0. \tag{69}$$

So, to find value of $\tilde{j}$, we need to first calculate value of f_E by solving equation $\Psi(f_E, j) = 0$ in $\mathbb{F}_p$. Since we are in Elkies case this equation will be at most quadratic, and thus it will be easy to find at least one solution f_E. Calculating $\tilde{j}$ will be done by searching for all zeros of $\Psi(f_E, \tilde{j})$ in $\mathbb{F}_p$ different than j. Also, since we are interested in finding specific isogeny which kernel is invariant to Frobenius endomorphism, after calculation of g_ℓ we need to prove that we have found good $\tilde{j}$ by checking whether g_ℓ divides division polynomial ψ_ℓ. Also note that here we can construct division polynomial in $\mathbb{F}_p[x]/\langle g_\ell \rangle$. Furthermore, we can save all necessary terms used in construction of division polynomials that can be reused later for finding rational expression for $[n](x, y), n \in \mathbb{N}$.

Now, given j, j' and $\tilde{j}$, let us show how to compute $\tilde{j}'$ and also $\left(\frac{j''}{j'} - \ell\frac{\tilde{j}''}{\tilde{j}'}\right)$, which is only other place in explained SEA algorithm we need to adapt. We take modular function f_E to be $f_E = f(q)$, and use notation f and j to mean both modular forms $j = j(q), f = f(q)$ and values in $\mathbb{F}_p$. Differentiating equations (68) and (69) we get

$$f'\Psi_f(f,j) + j'\Psi_j(f,j) = 0, \qquad f'\Psi_f(f,\tilde{j}) + \ell\tilde{j}'\Psi_j(f,\tilde{j}), \tag{70}$$

from where we conclude $f' = -j'\Phi_j(f,j)/\Phi_f(f,j)$, and thus

$$\tilde{j}' = -f'\frac{\Phi_f(f,\tilde{j})}{\ell\Phi_j(f,\tilde{j})} = \frac{j'\Phi_j(f,j)}{\Phi_f(f,j)}\frac{\Phi_f(f,\tilde{j})}{\ell\Phi_j(f,\tilde{j})}. \tag{71}$$

Differentiating equations in (70) one more time, we get:

$$\begin{aligned} f''\Psi_f(f,j) + f'^2\Psi_{ff}(f,j) + 2f'j'\Psi_{fj}(f,j) + j''\Psi_j(f,j) + j'^2\Psi_{jj}(f,j) = 0,\\ f''\Psi_f(f,\tilde{j}) + f'^2\Psi_{ff}(f,\tilde{j}) + 2\ell f'\tilde{j}'\Psi_{fj}(f,\tilde{j}) + \ell^2\tilde{j}''\Psi_j(f,\tilde{j}) + \ell^2\tilde{j}'^2\Psi_{jj}(f,\tilde{j}) = 0. \end{aligned} \tag{72}$$

Then, by using these two equations we can compute:

$$\begin{aligned} \left(\frac{j''}{j'} - \ell\frac{\tilde{j}''}{\tilde{j}'}\right) = &-\frac{1}{f'\Phi_f(f,\tilde{j})}\left(f'^2\Phi_{ff}(f,\tilde{j}) + 2\ell f'\tilde{j}'\Phi_{fj}(f,\tilde{j}) + \ell^2\tilde{j}'^2\Phi_{jj}(f,\tilde{j})\right)\\ &-\frac{1}{\Phi_j(f,j)}\left(j'\Phi_{jj}(f,j) + 2f'\Phi_{fj}(f,j) + \frac{f'^2}{j'}\Phi_{ff}(f,j)\right). \end{aligned} \tag{73}$$

The changes mentioned in this section will affect lines 3 and 5 of pseudocode for generating g_ℓ. For line 3, value of $\tilde{j}'$ should be calculated as in (71), while in line 5 we should set r as in the right hand side of equation (73).

5.5 Implementation and comments

The SEA algorithm variant described in this paper was implemented in SAGE mathematical software. The SAGE was chosen due to its simplicity, in hope that the code will be easy to upgrade and maintain. Also, author believes that it is the first open source SAGE implementation of SEA algorithm, and thus hopefully it will be useful for researchers wanting to quickly check impact of their ideas on the SEA algorithm.

Unfortunately, there have been spotted downsides of using SAGE software as well. Because it is based on Python, most of the code is not optimized in the best way which results in considerably slower performances compared to the implementations in compiled programming languages such as C. In comparison to PARI/GP code, the algorithm implemented in this project performs on average 10 to 20 times slower. The performance was slightly improved by cythonizing the code, but only by about 20%.

We have also implemented early-abort strategy for the algorithm. The idea is that when we want to check whether a given curve is safe for cryptographic use, we are interested to check whether it has prime number of points for some large prime. Thus, in case we detect during execution of our algorithm that a curve is divisible by a small prime, we can immediately reject the curve as unsafe. Early-abort strategy is implemented for Elkies stage of algorithm, since at every Elkies step we have calculated

$$t = p + 1 - |E(\mathbb{F}_p)| \pmod{\ell},$$

and thus we can escape each time we calculate for some Elkies prime ℓ that $p + 1 - t \equiv 0 \pmod{\ell}$. Another safety requirement is that a twist of elliptic curve is also prime number, and we can easily include this requirement into early-abort stragey for Elkies primes knowing that quadratic twist has $p+1+t$ rational points.

The code is available for download from github repository: `https://github.com/StankovicAleksa/SEA-SAGE`.

6 Conclusion

In this paper we have discussed implementation of Schoof-Elkies-Atkin algorithm. The algorithm is developed for special purpose of checking whether given elliptic curve or its twist has prime number of rational points, which find its importance in cryptography. While most of the steps respected standard procedure of SEA algorithm, for the sake of performance algorithm was implemented with early detection of curves whose order is divisible by small primes, for which we have aborted the full calculation.

While the algorithm is slow compared to variant implemented in C, it is still useful for checking the number of points for reasonably large curves, as well as for quickly rejecting curves whose order is divisible by small primes.

Furthermore, for ensuring the end users that curve which they use for encryption/decryption is indeed safe, it is useful to provide a code that is easier to understand and which they can check. And indeed Python based code is quite suitable for this as well, since it tends to be more readable compared to codes written in most other languages.

Finally, we note that there are numerous possible upgrades for this algorithm. First of all, we might want to check underlying SAGE implementation of routines to see where the performance losses come from. On more theoretical side, for Atkin values of ℓ (usually $\ell < 13$) it can be interesting to see if using Schoof's algorithm will be more suitable then using Atkin's approach. Finally, we might want to consider implementing the strategy revolving around isogeny cycles [8] for calculating value of trace t modulo powers of ℓ^n, again for ℓ chosen reasonably small and with $n \leq 4$.

References

[1] R.M. Avanzi, *Generic algorithms for computing discrete logarithms*, Handbook of Elliptic and Hyperelliptic Curve Cryptography (Henri Cohen and Gerhard Frey, eds.), Chapman & Hall/CRC, 2006, pp. 477–494.

[2] R. M. Avanzi, N. Thériault, *Index calculus*, Handbook of Elliptic and Hyperelliptic Curve Cryptography, Chapman & Hall/CRC, 2006, pp. 495-509.

[3] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, *Recommendation for key management - part 1: General (revised)*, Tech. Report NIST Special Publication 800-57, National Institute of Standards and Technology, May 2006.

[4] E.R. Berlekamp, *Factoring Polynomials Over Finite Fields.* Bell System Technical Journal. 46: 1853–1859. doi:10.1002/j.1538-7305.1967.tb03174.x. MR 0219231. BSTJ Later republished in: Berlekamp, Elwyn R. (1968). Algebraic Coding Theory. McGraw Hill. ISBN 0-89412-063-8.

[5] I. Blake, G. Seroussi, N. Smart, *Elliptic curves in cryptography,* Undergraduate Texts in Mathematics, Springer-Verlag, 2006.

[6] R. Bröker, K. Lauter, and Andrew V. Sutherland. *Modular polynomials via isogeny volcanoes.* Math. Comput., 81(278), 2012.

[7] J.A. Buchmann, *Introduction to Cryptography (2nd ed.)*, Springer Science & Business Media, pp. 190–191, (2013)

[8] J.-M. Couveignes, L. Dewaghe, and F. Morain, *Isogeny cycles and the schoof-elkies-atkin algorithm.* In Research Report LIX/RR/96/03, LIX, page 96, 1996.

[9] W. Diffie, M. Hellman, M. *New directions in cryptography*, IEEE Transactions on Information Theory. 22 (6) (1976).

[10] T. ElGamal (1985), *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory. 31 (4): 469–472.

[11] G. Frey, T. Lange, *Arithmetic of elliptic curves*, Handbook of Elliptic and Hyperelliptic Curve Cryptography, Chapman & Hall/CRC, 2006, pp. 529-543.

[12] S.D. Galbraith, *Mathematics of Public Key Cryptography*, Cambridge University Press, 2012.

[13] P. Gaudry and F. Morain, *Fast algorithms for computing the eigenvalue in the schoof-elkies-atkin algorithm*, In ISSAC'06, pages 109-115. ACM Press, 2006.

[14] O. Goldreich, *Foundations of Cryptography: Volume 1*, Cambridge University Press (2001).

[15] S. Lang, *Elliptic curves: Diophantine analysis*, A Series of Comprehensive Studies in Mathematics 231, Springer-Verlag, 1978.

[16] J. López, R. Dahab, *An overview of ellitic curve cryptography*, Institute of Computing, State University of California, May 2000.

[17] A.J. Menzes, P.C. van Oorschot, S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1996.

[18] J. M. Pollard, *Monte Carlo methods for index computation (mod p)*, Mathematics of Computation. 32 (143)

[19] C. Pomerance, *A Tale of Two Sieves*, Notices of the AMS. 43 (12). pp. 1473–1485.

[20] R. Rivest, A. Shamir, L. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM. 21 (2): 120–126.

[21] R. Schoof, *Elliptic curve over finite fields and the computation of square roots mod p*, Mathematics of Computation 44 (1985), no. 170, 483-495.

[22] R. Schoof, *Counting points on elliptic curves over finite fields*, Journal de Théorie des Nombres 7(1995), 219-254.

[23] D. Shanks, *Five Number Theoretic Algorithms*, Proceedings of the Second Manitoba Conference on Numerical Mathematics. Pp. 51–70. 1973.

[24] J.H. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics 106. Springer-Verlag, 2007.

[25] A.V. Sutherland, *On the evaluation of modular polynomials*, CoRR, abs/1202.3985, 2012.

[26] A. Tonelli, *Bemerkung über die Auflösung quadratischer Congruenzen*, Nachrichten von der Königlichen Gesellschaft der Wissenschaften und der Georg-Augusts-Universität zu Göttingen. Pp. 344–346. 1891.