SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

ECPP primality proving

av

Simon Vestberg

2022 - No K3

ECPP primality proving

Simon Vestberg

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Jonas Bergström

2022

ECPP primality proving

Simon V

October 2021

Abstract

I present and aim to describe the Goldwasser-Kilian algorithm for primality proving, which is an algorithm that make use of elliptic curves and produces a primality certificate on prime input. Subsequently, I describe Atkin-Morains algorithm on a surface level. This is an algorithm that uses roughly the same ideas but construct curves with the correct cardinality as to not use Schoof's algorithm of counting points on an elliptic curve.

Key words: elliptic curve, algorithm, primality proving, prime.

1 Introduction

Already at 500 to 300 BC mathematicians studied primes extensively for their special and numerological properties and by the time of 300 BC Euclid had proven both the fact that there are infinitely many primes and given a proof of the fundamental theorem of arithmetic. However after about 200 BC there was a huge break in the history of primes and not until the 17th century and Fermat was any particular progress made.

Definition 1.1. *An integer, p, is called prime if $p \geq 2$ and p only divisible by itself and* 1.

For example $2, 3, 5, 7, 11, ...$

Definition 1.2. *An integer, m, is called composite if it is divisible by another integer, $1 < n < m$, that is not* 1 *or itself.*

For example $14 = 7 \cdot 2, 596 = 2 \cdot 2 \cdot 149$.

According to the fundamental theorem of arithmetic any integer can be factorized into a product of prime numbers (if the integer is a prime its factorization is just itself). This is in fact a big part of why primes are of such importance to cryptography.

Now you might question why prime numbers? Why are they particularly important to cryptography? This is because in certain cryptosystems for example in RSA, the Rivest-Shamir-Adleman algorithm, the difficulty in "cracking" the

key stems from the idea that it take too long (too computer-heavy) to factorize large integers. And as we know, in RSA, the public key consists of two primes, p and q, which are unknown to the public. While the product $p \cdot q = N$ is known. Primes are also relevant for e.g. ECC, elliptic curve cryptography, where you want to utilize groups of prime order to make certain algorithms practically unusable meaning you get a more secure system. More precisely ECC is based on elliptic curves defined over a finite field, where the field usually has a prime amount of elements.

Definition 1.3. *An algorithm that takes as input an integer, N, and checks if it is possible to factorize is called a primality test. I.e. the test determines if N is a prime number or not.*

As the definition above states, primality test algorithms determine if a given input, n, is prime or not. These are for the most part probabilistic, meaning they have a possibility to fail, i.e. say that the input n is prime when it actually is not. It does however exist primality tests, which are deterministic, that give a definitive answer. On the other hand a primality proof, also called a primality certificate, is something you can use to prove that an integer is prime. The certificate is usually in a form of a list which you can check, to make sure said integer is prime, by using some algorithm.

So how do we determine if an integer, n, is prime or if it is composite? We are now talking about very large integers so obviously we are not expected to do it by brute-force. This is where primality tests come into the picture. There are quite a few primality tests but we are going to study one closer, the Goldwasser-Kilian algorithm using elliptic curves.

2 Elliptic curves

Definition 2.1. *A 'field' is a set F together with two binary operations, call these addition and multiplication. These operations are binary mappings $F \times F \to F$, and we denote the addition of two elements a, $b \in F$ as $a+b$ and multiplication as $a \cdot b$ or ab. If these operations satisfy the field axioms, namely:*

$$(i)\ \textit{commutativity}: \ a + b = b + a \ \textit{respective}\ ab = ba$$

$$(ii)\ \textit{associativity}: \ (a + b) + c = a + (b + c) \ \textit{respective}\ (ab)c = a(bc)$$

$$(iii)\ \textit{distributivity}: \ a(b + c) = ab + ac \ \textit{respective}\ (a + b)c = ac + bc$$

$$(iv)\ \textit{identity}: \ a + 0 = a = 0 + a \ \textit{respective}\ a1 = a = 1a$$

$$(v)\ \textit{inverses}: \ a + (-a) = 0 = (-a) + a \ \textit{respetive}\ aa^{-1} = 1 = a^{-1}a$$

we call this set F a 'Field', denoted as $\mathbb{F}$.

Generally a field is a (commutative) ring [7, page 83-84] where every non-zero element has a multiplicative inverse.

Lemma 2.2. *If p is prime then the set of integers $\mathbb{Z}$ mod p, usually denoted as $\mathbb{Z}/p\mathbb{Z}$, with its addition and multiplication rules is a field. [8, page 28]*

Some examples of fields are; all reals $\mathbb{R}$, the rationals $\mathbb{Q}$ or the complex numbers $\mathbb{C}$. Furthermore the field $\mathbb{Z}/p\mathbb{Z}$ has finitely many elements and is therefore known as a *finite* field, we denote this as $\mathbb{F}_p$. Furthermore we define the number of elements in a finite field as the *order* of the field.

Finite fields are of elementary importance for cryptology and this is mainly because of the property that any finite field has p^m elements, where p is a prime and m is an arbitrary positive integer. For example a field can have $2197 (= 13^3)$ elements but can not have $14 (= 2 \cdot 7)$ elements. A field where $m = 1$ is called a *prime field*.

Definition 2.3. *Let $\mathbb{F}$ be a field, an elliptic curve (in Weierstrass form) over $\mathbb{F}$ is the ordered pair (A,B), with $A, B \in \mathbb{F}$, and $4\,A^3 + 27\,B^2 \neq 0$. We define the points of the curve (A,B) as the set of ordered pairs (x, y), with $x, y \in \mathbb{F}$, which are solutions to the equation $y^2 = x^3 + Ax + B$ together with an extra point I. Furthermore we call this I the point at infinity. We denote the set of these points as $E_{A,B}(\mathbb{F})$.*

We introduce I, which can almost be seen as an artificial point, to our set of solutions to have a neutral element and later to be able to show that the points of the elliptic curve form a group. Since without this element, I, certain additions $P + Q$ would have no value (for some $P, Q \in E_{A,B}(\mathbb{F})$). We are using the standard algorithms for addition and doubling of points on the curve (A,B). These can be found in full in [6, page 456].

Also worth noting is that the condition $4\,A^3 + 27\,B^2 \neq 0$ is needed to make sure that our curve doesn't have any singular points which in turn is needed for the addition law [8, page 303] to work well. What $4\,A^3 + 27\,B^2 \neq 0$ actually ensures is that if we factor $x^3 + Ax + B$ completely as

$$x^3 + Ax + B = (x - e_1)(x - e_2)(x - e_3)$$

then $4\,A^3 + 27\,B^2 \neq 0$ if and only if e_1, e_1, e_3 are distinct which is equivalent to say that the curve doesn't intersect itself and doesn't have any cusps.

Let E be an elliptic curve on normal Weierstrass form $y^2 = x^3 + A\,x + B$ with $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ some points on E.

Standard addition algorithm$((x_1, y_1), (x_2, y_2), (A, B))$

1. If $x_1 = x_2$ and $y_1 = -y_2$ return(I).
2. If $x_1 = x_2$ and $y_1 = y_2$ then put $\lambda = \frac{3x_1^2 + A}{2\,y_1}$.
3. If $x_1 \neq x_2$ put $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$.

Let $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$
then return(x_3, y_3).

If the algorithm above is applied to P and Q we will denote the resulting point (x_3, y_3) as $P+Q$. Note that since the only operations used in the addition algorithm are addition, subtractions, multiplication and division with A and the coordinates of P and Q which are all in the field $\mathbb{F}$. The resulting point coordinates (x_3, y_3) will also be in $\mathbb{F}$.

Moreover we define qL, where q is an integer and L a point on an elliptic curve, by repeated addition and the value of qL may be calculated in the following way, using repeated doubling,

$$qL = \begin{cases} L & q = 1 \\ (L + L) \cdot q/2 & \textit{if } q \textit{ even} \\ L + (q-1)\, L & \textit{if } q \textit{ odd.} \end{cases}$$

Definition 2.4. *Let E be an elliptic curve over some field $\mathbb{F}$, we then define I as the neutral element such that for $P \in E_{A,B}(\mathbb{F})$*

$$P + I = I + P = P$$

and

$$P + (-P) = I$$

where $-P$ is defined to be $P = (x, y)$ reflected in the x-axis, i.e. $-P = (x, -y)$.

Theorem 2.5. *Let E be an elliptic curve over some field F, then the set of points $E_{A,B}(\mathbb{F})$ form an abelian group, i.e. commutative group, with the operation of addition defined as the standard point addition. That is, the points satisfy the commutativity, associativity, identity and inverse criteria as described in definition 2.1 with the operation of addition and also if P and Q are points in $E_{A,B}(\mathbb{F})$ then $P + Q$ will be a point in $E_{A,B}(\mathbb{F})$.*

Proof. The properties of identity and inverses are clear from definition 2.4 above while commutativity is easily seen by just switching the points in the addition algorithm will result in the same outcome. Associativity could be checked through the addition algorithm as well, however this is not as straightforward as for commutativity and requires a lot of laborious calculations, so instead I refer the reader to [11, page 61-62] where Silverman uses the Riemann-Roch theorem to prove associativity. Left to show is that $E_{A,B}(\mathbb{F})$ is closed under the operation i.e. if $P, Q \in E_{A,B}(\mathbb{F})$ then $P + Q \in E_{A,B}(\mathbb{F})$. This however comes directly from the addition algorithm and by noting that λ is the slope of the line between the two points if $P \neq Q$ and the slope of the tangent line if $P = Q$. Either way, substituting the equation of the line $y = \lambda x + d$, that intersect P and Q, into the equation of E and solving for x will directly give a solution to the equation defining E and as such a point on E.

□

Definition 2.6. *Furthermore, considering the size of this group we denote the number of points on (A,B) over $\mathbb{F}_p$ as $\#(A, B)$.*

It is worth noting here that it is possible to define this as an (abelian) group because we added I so that the group axioms of identity and inverse actually are true.

Furthermore the structure of this group will be cyclic or isomorphic to $\mathbb{Z}/\mathrm{mp}\mathbb{Z} \bigotimes \mathbb{Z}/\mathrm{np}\mathbb{Z}$ for some $n, m \in \mathbb{Z}$. This however is very much non-trivial and will not be proven here.

Definition 2.7. *Let E be an elliptic curve over $\mathbb{F}$, represented in the Weierstrass form*

$$E : y^2 = x^3 + Ax + B$$

the j-invariant of E is then given by

$$j(E) = 1728\frac{4A^3}{4A^3 + 27B^2}$$

and the discriminant is given by

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

It is worth noting that since any elliptic curve with Weierstrass equation will have a specific discriminant and j-invariant. In addition, if we have an isomorphism, this must respect the group structure and so must preserve the Weierstrass form of the equation. With this in mind, all isomorphisms are just changes of variables and actually very specific such changes.[11, page 45] Namely if E is an elliptic curve over $\mathbb{F}$ on Weierstrass form:

$$E : y^2 = x^3 + Ax + B$$

with its corresponding discriminant and j-invariant then the only change of variables preserving this form is

$$x = u^2x' \qquad y = u^3y' \qquad \text{for some } u \in \bar{\mathbb{F}} \setminus \{0\}.$$

Theorem 2.8. *Let $\mathbb{F}$ be a field and E, E' be two elliptic curves over $\mathbb{F}$ with j-invariants j and j'. Then there exists an isomorphism from E to E' over $\bar{\mathbb{F}}$ (the algebraic closure of $\mathbb{F}$ [5, page 543]) if and only if $j = j'$. [12, page 46]*

Let us consider the elliptic curve (A, B) over the ring $\mathbb{Z}_n$, defined (as in the definition of elliptic curve over a field) as the set of solutions (x, y) over $\mathbb{Z}_n$ to the equation:

$$y^2 = x^3 + Ax^2 + B.$$

This will form a set of points $E_{A,B}(\mathbb{Z}_n)$ and if $L, M \in E_{A,B}(\mathbb{Z}_n)$ is two points on this curve we can use the addition algorithm to calculate $L + M$. Note that this might not always be defined, we will discuss this more later.

Let p be a prime greater than 3 and n an integer and assume that $p|n$, then we can look at $4A^3 + 27B^2 \neq 0 \bmod p$ as well as mod n, since p divides n. We have that $\mathbb{Z}_n$ is the set of integers $\mathbb{Z}$ mod n and an elliptic curve defined over $\mathbb{Z}_n$ will give rise to groups $E(\mathbb{F}_p)$, for the various p that divides n. So given a point $L = (x, y) \in E_{A,B}(\mathbb{Z}_n)$ we define $L_p = (x_p, y_p)$. Where x_p is defined as the natural projection from $x \in \mathbb{Z}_n$ to $\mathbb{F}_p$.

Lemma 2.9. *Let $L, M \in E_{A,B}(\mathbb{Z}_n)$. If $L + M$ is defined, using the standard addition algorithm, then $(L + M)_p = L_p + M_p$.*

First note that $L + M$ is defined when the requisite inverse elements exist and if $x_1 = x_2$ then $y_1 = \pm y_2$. Where $(x_1, y_1), (x_2, y_2)$ are the coordinates for L and M respectively and the inverse elements mentioned are the inverses to $x_2 - x_1$ and $2y_1$ needed in the calculations of λ in the addition algorithm. The proof of the lemma considers the different cases (with which I mean the possible combinations of x_1, x_2, y_1, y_2 e.g. from the addition algorithm: 1. $x_1 = x_2$ and $y_1 = -y_2$) that can occur when adding two points, using the standard algorithm for addition. If $L + M$ and $L_p + M_p$ falls into the same case it is quite straight forward to show that the lemma holds. If $L + M$ and $L_p + M_p$ fall into different cases, for example if $x_1 \neq x_2$ while $(x_1)_p = (x_2)_p$, then we need to show that $L + M$ will be undefined. For the full proof see [6, page 457].

3 Goldwasser-Kilian algorithm

The property of being prime is called primality. The Goldwasser-Kilian algorithm checks if an integer, q, is prime or composite. If it is prime it will produce a certificate of primality consisting of a list of elliptic curves, a point on each curve and an "easily proven" prime p. With "easily proven" I mean that it can be rapidly proven to be prime using the algorithm of Cohen and Lenstra [4]. The certificate can then be deterministically checked using the $prove-prime(q)$ algorithm in $O((\lg q)^4)$ time. The big O notation is here used as a way to describe how fast, or slow, an algorithm runs based on the input size. If this 'prove-prime' algorithm accepts a certificate we say that we have proof of primality. And indeed we will later see that this algorithm will not accept a certificate unless q actually is prime.

Theorem 3.1 (Hasse's theorem)**.** *If N is the number of points of an elliptic curve E over $\mathbb{F}_p$, then $|N - (p + 1)| \leq 2\sqrt{p}$.*

Proof. Let

$$\phi_p : \bar{\mathbb{F}}_p \longrightarrow \bar{\mathbb{F}}_p$$
$$x \to x^p$$

be the p^{th}-power Frobenius map for $\mathbb{F}_p$ and let E be an elliptic curve over $\mathbb{F}_p$. Then ϕ_p acts on the points of $E(\bar{\mathbb{F}}_p)$ as follows

$$\phi_p(x, y) = (x^p, y^p), \qquad \phi_p(\infty) = \infty.$$

Then computing the number of points of E mod p will be the same as computing the number of solutions to $\phi(P) = P \iff$ #solutions for $(\phi - 1)P = 0$. The fact that the map fixes E pointwise, i.e. $\phi(P) = P$, we have from $x^p \equiv x$

mod p (Fermat's little theorem) for $x \in \mathbb{F}_p$ but not for $x \in \bar{\mathbb{F}}_p \setminus \mathbb{F}_p$. So P is in the kernel of $(\phi - 1)$ thus

$$E(\mathbb{F}_p) = ker(\phi - 1)$$

and using [11, III.5.5] and [11, III.4.10c] we have that

$$\#E(\mathbb{F}_p) = \#ker(\phi - 1) = deg(\phi - 1).$$

Since the degree map on End(E) (the set of homomorphisms from E to itself) is a positive definite quadratic form [11, III.6.3] and $deg(\phi) = p$ and $deg(\phi - 1) = \#E(\mathbb{F}_p) = N$, the following version of Cauchy-Scharwz inequality, Lemma 3.2, gives the desired result.[11, page 138] With $\psi = 1$ and $\phi = \phi$ we get $d(\phi - \psi) = d(\phi - 1) = deg(\phi - 1) = N$, $d(\phi) = deg(\phi) = p$ and $d(\psi) = 1$ which results in the bound given in the theorem. □

Lemma 3.2. *Let G be an abelian group and let*

$$d : G \longrightarrow \mathbb{Z}$$

be a positive definite quadratic form then

$$|d(\phi - \psi) - d(\phi) - d(\psi)| \leq 2\sqrt{d(\phi)d(\psi)}, \quad \forall\, \phi, \psi \in G.$$

Note that Hasse's theorem does only give us a bound for the number of points and not the exact number. We will use Schoof's algorithm, which utilizes theorem 3.1, to calculate the exact number of points of the curve. From 3.1, Hasse's theorem, we have that

$$E(\mathbb{F}_p) = p + 1 - N$$

where $|N| \leq 2\sqrt{p}$. Then let $S = 2, 3, 5, ..., L$ be the set of all primes such that

$$\prod_{l \in S} l > 4\sqrt{p}.$$

Usually L is chosen to be the least number so that the inequality holds. If we can determine N mod l for each $l \in S$ we can know N mod $\prod l$ and therefore uniquely determine N.

This is done by using the Chinese Remainder Theorem, by knowing N mod l for each $l \in S$ we can calculate N mod $\prod l$ and find N that satisfies this congruence and $|N| < 2\sqrt{p}$. In addition, the so called division polynomials ψ_l are used, see section 3.2 [12, page 81]. These polynomials have the property that they vanish precisely in the $l-$torsion points, so the roots of ψ_l are the x-coordinates of the points in $E(l)$. Here $E(l)$ is the set of l-torsion points on an elliptic curve $E(\bar{\mathbb{F}}_p)$ more concisely

$$E(l) = \{P \in E(\bar{\mathbb{F}}_p) : l \cdot P = I\}.$$

If we have a point (x, y) of order l then

$$(x^{p^2}, y^{p^2}) + p(x, y) = a(x^p, y^p)$$

now let $p_l \equiv p \bmod l, \quad |p_l| < \frac{l}{2}$ then

$$(x^{p^2}, y^{p^2}) + p_l(x, y) = a(x^p, y^p).$$

Now the idea is, since (x^p, y^p) are also of order l, that we can determine a mod l, by computing the other terms apart from a and find a value for a that makes the relation hold. For more details regarding the proof of Schoof's algorithm read section 4.5 in [12, page 123]. The time complexity of Schoof's algorithm is $O(\lg^8 p)$, where most of the computational time comes from calculating powers x^p, x^{p^2}... modulo the division polynomial and the curve, and the multiplication l times the point (x^p, y^p). So with $l = O(\lg p)$ and assuming we use the standard multiplication algorithm we arrive at the conclusion that the entire algorithm is calculated in $O(\lg^8 p)$ i.e. in polynomial time [10, page 234].

Recall that the set of points of the curve (A,B) form an abelian group with the operation of addition defined as the standard point addition. Also recall that when considering the size of this group we denote the number of points on (A,B) over $\mathbb{F}_p$ as $\#(A, B)$ (or $\#_p(A, B)$ where subscript p is to clarify the order of the field). We then have from Hasse's theorem that $p+1-2\sqrt{p} \leq \#(A, B) \leq p+1+2\sqrt{p}$. This procure the ground for the first important theorem, given by Lenstra, which states that the probability that the $\#(A, B)$ is in the interval $S \subseteq [p + 1 - \lfloor\sqrt{p}\rfloor, p + 1 + \lfloor\sqrt{p}\rfloor]$ is larger than a certain expression.

Theorem 3.3. *Let $p > 5$ be a prime and let*

$$S \subseteq [p + 1 - \lfloor\sqrt{p}\rfloor, p + 1 + \lfloor\sqrt{p}\rfloor].$$

If curve (A,B) over $\mathbb{F}_p$ is uniformly choosen then

$$prob(\#(A, B) \in S) > \frac{c}{ln\, p} \cdot \frac{|S| - 2}{2\lfloor\sqrt{p}\rfloor + 1},$$

where c is some fixed constant.[6, page 458] [9, page 667]

We will use this theorem later to bound the amount of curves we have to test to find one that has an order that is twice a prime.

3.1 Main primality proving algorithm

The second theorem which in reality is the main theorem of the Goldwasser-Kilian algorithm is a primality criterion and is stated as follows:

Theorem 3.4. *Let n be a integer and not divisible by 2 or 3. Let $A, B \in \mathbb{Z}_n$ with $gcd(4A^3 + 27B^2, n) = 1$ furthermore let $L \in E_{A,B}(\mathbb{Z}_n)$ with $L \neq I$. If $qL = I$ for some prime $q > n^{1/2} + 2n^{1/4} + 1$, then n is prime.*

Proof. The proof of this is given by a contradiction. Suppose n is composite, then there exists a divisor $p \neq 2, 3$ with $p \leq \sqrt{n}$ further $4A^3 + 27B^2 \neq 0 \ mod \ p$. Thus $L_p \in E_{A,B}(\mathbb{F}_p)$ and $qL_p = I$ (since $I = I_p$), for some prime $q > n^{1/2} + 2n^{1/4} + 1$, by application of lemma 2.9. Then since $L_p \neq I$ and the order of L_p must divide q (because $qL_p = I$ we have that the order of L_p is either $1\,, q$ or something dividing q) which is prime we have that the order of $L_p = q$. However, the order of L_p is at most $\#_p(A, B) \leq p+1+2\cdot\lfloor\sqrt{p}\rfloor \leq n^{1/2}+2n^{1/4}+1 < q$. $\square$

Given a prime $(n =)\, p$ the algorithm starts by uniformly generating a curve (A, B) over $\mathbb{F}_p$, with $(4A^3 + 27B^2, p) = 1$ and $\#_p(A, B) = 2q$, with q prime and $q \approx p/2$, and a point L on this curve of order q. This q will satisfy the inequality, and we can then use the primality criteria, given in theorem 3.4 to reduce the primality of p to the primality of q. It is worth noting that it is possible to allow $\#_p(A, B) = rq$, where r is some smooth number (or if not at least easy to factor out) and q large enough. However an analysis of this will not be presented here.

So the first step in the algorithm is to find A, B such that these criteria fit, this is done by uniformly picking A, B and checking if $(4A^3 + 27B^2, p) = 1$ and by Schoof's algorithm counting the points on the curve (A, B). If this is equal to $2q$, for some prime q, q is checked for primality using a standard primality testing algorithm, e.g. Miller-Rabin [8, page 131], with an extremely small, roughly $1/p$ where p is the prime we initially want to prove, probability of error (note that it takes relatively few calculations to complete this probabilistic test). This is repeated until we have found (A, B) that fits both of these requirements.

After (A, B) has been found, we uniformly pick $x \in \mathbb{F}_p$ and calculate $z = x^3 + xA + B$, if z is a quadratic residue we calculate $\sqrt{z} = \pm y$ and set $L = (x, \pm y)$, which of the y (the positive or negative root) we want to use is chosen uniformly. Since x is chosen independently and uniformly we know that z will be a quadratic residue with a constant probability and therefore only an expected number choices of x are needed, namely ≈ 2 [8, page 309] (since there are, for $p > 2$ prime, $\frac{p+1}{2}$ quadratic residues in $\mathbb{F}_p$ counting the zero). The algorithm then says to compute $qL = I$ to ensure that L is of order q, if $qL \neq I$ then we again look for x such that z is quadratic residue. There exist several algorithms to do the calculation and find the square roots of z mod p, Goldwasser and Kilian decide to use that of Adleman et al. [1], this runs in random polynomial time, which is a generalization of the algorithm given by Tonelli and Shanks.

Then we basically just iterate this reduction of primality until a certain bound, set to be such that the prime to be proven is small enough to be determined as prime in polynomial k time, where k is the number of bits of the initial prime p we wanted to certificate. In the full algorithm there is also a "fail-safe" to ensure that we can handle the rare cases when the probabilistic test, that we use to prove q prime, makes a mistake and the algorithm gets stuck trying to prove a composite number to be prime.

3.2 Procedure

Main-step(p)

Step 1. Compute $(A, B), q$ by generating curve and point L calculate, by repeated doubling, $qL = I$ (if $qL \neq I$ find new L).

Step 2. Return $((A, B), L, q)$

In 'Algorithm prove-prime(p)' we give the full algorithm which will return a certificate of primality, this certificate can be checked in the check-prime algorithm. The fail-safe I mentioned before is implemented as a check that if it has gone more than k^{lgk} steps since step 1. abort.

The check-prime algorithm will make use of the primality criterion from theorem 3.4 and if it accepts an input $p, (((A_0, B_0), L_0, p_1), ...((A_{i-1}, B_{i-1}), L_{i-1}, p_i))$ as prime. Then p_i must be prime, and by theorem 2, the check made throughout check-prime ensure if p_{j+1} prime then p_j is prime. Then p_i prime $\Rightarrow p_{i-1}$ prime ... $\Rightarrow p_0$ prime. Hence p must be prime.

Algorithm prove-prime(p)

Step 1. Let $i = 0$, $p_0 = p$ and lowerbound $= max(2^{k^{C/lglgk}}, 37)$

Step 2. While $p_i >$ lowerbound do
$(A_i, B_i), L_i, p_{i+1} \longleftarrow$Main-step$(p_i)$
set $i = i + 1$, if any p_i is divisible by 2 or 3 go back to 1.

Step 3. Use a deterministic test to check p_i prime. If it's not prime return to 1, otherwise return the certificate: $(p, ((A_0, B_0), L_0, p_1), ...((A_{i-1}, B_{i-1}), L_{i-1}, p_i))$

Check-prime($(p, ((A_0, B_0), L_0, p_1), ...((A_{i-1}, B_{i-1}), L_{i-1}, p_i))$**)**

Step 1. Abort if $p_i > max(2^{k^{C/lglgk}}, 37)$ otherwise test p_i for primality by a deterministic test.

Step 2. Define $p_0 = p$ For $j \in [0, i-1]$, check that
- p_i not divisible by 2 or 3
- $(4A_j^3 + 27B_j^2, p_j) = 1$
- $p_{j+1} > p_j^{1/2} + p_j^{1/4} + 1$
- $L_j \neq I_{p_j}$ and $p_{j+1}L_j = I_{p_j}$

Step 3. If these do not hold, abort. Otherwise accept p as prime.

4 Small example

I will illustrate the algorithm with a small, quite easy, example, I use Sagemath to do the necessary computations. We start with a number we believe to be prime and wish to create a certificate for, in this case I took $p = 1021$ which is a very small prime in the grand scheme of things. We then use the 'randint' function to uniformly choose A, B such that $(4A^3 + 27B^2,\ p) = 1$ and I do this til I got a curve with cardinality equal to $2q$, for some prime $q > p^{1/2} + 2p^{1/4} + 1$. To calculate the number of points of the curve I use Sagemaths function '.cardinality'. I then find a point $L = (x, y)$ according to the algorithm by again using 'randint' to uniformly choose x until I find a point such that $qL = I$. I then save $(A, B), L, q$ and repeat the procedure with $p = q$.

So starting with $p = 1021$ I found $(A_0, B_0) = (766,\ 924)$ and $q = 503$. These are, as mentioned above, found by 'randint' for A_0, B_0 and then calculating the points on the curve E_{A_0,B_0} through '.cardinality' to find q. This is done until A_0, B_0 and q satisfy the criteria. Then when looking for a point the first $x = 1008$ that made $z = x^3 + xA + B\ mod\ p = 0$ to be a quadratic residue, as desired, actually did not have $qL = I$ so I had to keep "drawing" numbers till I hit another $x = 859$ with $\sqrt{z} = y = \pm 17$. Which luckily did make $qL = I$, so I save these $(A_0, B_0),\ q_0 = p_1,\ L_0$ and move forward with $p = 503$. With the same technique as above I found $(A_1, B_1) = (432, 455),\ q_1 = p_2 = 241,\ L = (x, y) = (253,\ -17)$ however this time it actually took me seven tries to find a point, L, with order q_1. I continue this procedure of reducing p three more times and find $((A_2, B_2) = (173, 116),\ q_2 = p_3 = 127,\ L = (x, y) = (216,\ 13))$, $((A_3, B_3) = (15, 34),\ q_3 = p_4 = 73,\ L = (x, y) = (97,\ -4)$ and $((A_4, B_4) = (58, 0),\ q_4 = p_5 = 29,\ L = (x, y) = (4,\ 2)$. Now I deem the prime $p = 29$ small enough and we are done.

Our certificate then looks like this; $(1021,\quad ((766,\ 924), (859, -17), 503,$ $((432, 455), (253,\ -17), 241,\quad ((173, 116), (216,\ 13), 127,\quad ((15, 34), (97,\ -4), 73,$ $((58, 0), (4,\ 2), 29)$

$A = 766,\ B = 925,\ p = 1021$	$A = 432,\ B = 455,\ p = 503$
$E = EllipticCurve(GF(p,'a'), [A, B])$	$E = EllipticCurve(GF(p,'a'), [A, B])$
$R = IntegerModRing(p)$	$R = IntegerModRing(p)$
$gcd(4A^3 + 27B^2,\ p)(= 1)$	$gcd(4A^3 + 27B^2,\ p)(= 1)$
$q = int(E.cardinality()/2)$	$q = int(E.cardinality()/2)$
$x = 859$	$x = 253$
$z = x^3 + xA + B$	$z = x^3 + xA + B$
$R(z)$	$R(z)$
$y = -17$	$y = 17$
$L = E(x, y)$	$L = E(x, y)$
$L.order()$	$L.order()$
$q * L(= I)$	$q * L(= I)$
$(q = 503)$	$(q = 241)$

These calculations in Sagemath illustrates two steps of the algorithm, starting with $p = 1021$.

5 Time complexity

5.1 Complexity of check-prime

By the way we generate a curve and the fact that $\#_{p_j}(A_j, B_j) \geq p_j + 1 - 2\sqrt{p_j}$ we have that

$$p_{j+1} \geq \frac{p_j + 1 - 2\sqrt{p_j}}{2} > p_j^{1/2} + 2\,p_j^{1/4} + 1$$

for $p_j > 37$. If $p_j < 37$ then $p \leq 37$ by how we defined prove-prime and if this is the case it's easily verified that check will accept the output from prove-prime. And the way we choose a point on the curve gives us $L_j \neq I_{p_j}$ and $p_{j+1}L_j = I_{p_j}$ hence check will always accept a certificate on the form $(p, (((A_0, B_0), L_0, p_1), ...((A_{i-1}, B_{i-1}), L_{i-1}, p_i)))$ as given by prove-prime.

For a prime p, that is $k-$bits long, the steps required for the check to finish is $O(k^4)$. To see why this is the case, first observe that $p_{j+1} = p_j/2 + o(p_j)$ and therefore i, the number of primes p in the certificate, is equal to $O(\lg\, p) = O(k)$. And for each value of j the check algorithm must accomplish a set (constant) of standard arithmetic operations namely a single GCD computation and multiply a point L_j by an integer q_j. This can be done in $O(k^3)$ steps. Hence the computational time in entirety will be $O(k) \cdot O(k^3) = O(k^4)$.

5.2 Complexity of main-step

The time it takes to find a curve of order $2q$, which we are after, will be the expected number of curves to check multiplied by the expected time to generate and check one curve. This time is primarily used to calculate the number of points, i.e. by Schoofs algorithm which takes $O(\lg^8 p)$ steps to finish.

By utilizing Lenstra's theorem we can bound the number of curves that we need to test before we find one whose order is twice a prime.

Let $S(p)$ be defined as the set of primes in a given interval around $p/2$.

Lemma 5.1. *Let $p > 5$ and (A, B) be a uniformly chosen curve over $\mathbb{F}_p$ then*

$$prob(\#_p(A,B) is\, twice\, a\, prime) > \frac{c}{\lg\, p} \cdot \frac{|S(p)| - 2}{2\lfloor p \rfloor + 1}$$

for some fixed constant c.

For proof of this lemma and for a more descriptive definition of $S(p)$ I refer you to [6, page 462-463].

We have that generating a curve takes $O(k^{c+9})$ steps, what's left to check is how long selecting a point takes. This will in fact be a low-order term in comparison to the generation of the curve. Assume we have generated a curve which order is twice a prime, $E_{A,B}(\mathbb{F}_p)$ has order $2q$ for some prime q. Then $E_{A,B}(\mathbb{F}_p)$ will be isomorphic to $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ where $m_1 | m_2$, [12, page 97]. But since $E_{A,B}(\mathbb{F}_p)$ is of order $2q$ we have that $m_1 m_2 = 2q$ and therefore $m_1 = 1$ and $m_2 = 2q$, for $q > 2$. Thus $E_{A,B}(\mathbb{F}_p)$ will be isomorphic to $\mathbb{Z}_{2q}$ and therefore will have $q - 1$ points of order q. Furthermore, these points will be pairs since if (x, y) is a point so is $(x, -y)$ hence the expected time to select a point will be $2q/(q-1) = O(1)$ times the amount of time it takes to pick an x, compute y and check that (x, y) is of order q. It takes $O(k^3)$ to add two points and $O(k)$ to check if $qL = I$ using repeated doubling. Hence the naive running time will be $O(k^4)$, this can perhaps be improved but they are enough to prove that selecting a point is a low-order term.

6 Atkins & Morains algorithm

Another algorithm which is based on roughly the same idea as the Goldwasser-Kilians algorithm was constructed by Atkin and Morain, but using theory of elliptic curves over finite fields results, in particular properties associated with complex multiplication. From a practical standpoint it is said that this algorithm is faster and produces a list of numbers that may be easier to prove to have primality properties, i.e. easier to check if the computations done in the algorithm were correct. Since even though Schoofs algorithm is polynomial (of power 8) in complexity it has to be done many times which makes the Goldwasser-Kilian algorithm impractical.

The algorithm by Atkin and Morain works using properties of quadratic forms and in particular the theory of Hilbert class fields of imaginary quadratic

fields via modular forms, with the purpose of going from elliptic curve over $\mathbb{C}$ to elliptic curve over finite fields. They also develop an effective algorithm to construct Hilbert class fields of an imaginary quadratic field.

Here I will give an overview of what goes into the algorithm of Atkin and Morain without going through neither definitions nor details.

You start with finding a fundamental discriminant $-D_i$ which is 'good' for a given $N = N_i$ (N_0 is the probable prime you want to prove) and construct a quadratic field $K = \mathbb{Q}(\sqrt{-D})$ and compute a root j to $H_D(X) \equiv 0 \bmod N$ followed by computing the equation of the curve E with suitable cardinality. So in comparison to the G-K algorithm where we search for a curve and calculate its cardinality, the A-M algorithm goes about it kind of from the other direction where we start of by constructing a curve that has the cardinality we are after. This makes it possible to skip Schoofs algorithm which was a problem for the G-K algorithm.

When I say find a discriminant which is 'good' I mean that N should split in the quadratic order of discriminant D as a product of two elements $N = \pi\bar{\pi}$, or equivalently, there exist integers a, b such that $a^2 + b^2|D| = 4N$. Note that in G-K algorithm we search to find an a such that $N+1+a$ or $N+1-a$ has a prime factor q which is sufficiently large. In A-M we instead make sure this is the case by constructing the curve in such a way. If we can find such a discriminant we can construct an elliptic curve E over the complex numbers with complex multiplication by K. That E has complex multiplication by K means that its endomorphism ring (the morphisms from E to itself) when tensored with $\mathbb{Q}$ contains K.

Let $H_D(X)$ denote the Hilbert class polynomial which is defined as

$$H_D(X) = \prod_{k=1}^{h_d} (x - j(A_k)),$$

where $j(A_k)$ is the j-invariant of the elliptic curve corresponding to A_k, and the product is over all elliptic curves A_k with complex multiplication by K.

This is done by calculating the j-invariants of the $h(D)$ elliptic curves as complex numbers, which will form the roots of the class polynomial $H_D(X)$. Furthermore, because the j-invariants are calculated over $\mathbb{C}$ it is only possible to find approximations for these. However since $H_D(X)$ only have integer coefficients and we know roughly the size of these, from the theory of complex multiplication, we can approximate the j-invariants and round these to closest integer.

Now that we have found $H_D(X)$ with integer coefficients we can reduce this modulo N and find a root. This root will be the j-invariant for the elliptic curve E of the form

$$y^2 = x^3 - 3cg^{2k}x + 2cg^{3k}$$

where $c = j/(j - 1728)$, g is any non-quadratic residue and k is either 0 or 1. For any fixed j there are only two non-isomorphic curves E on this form, corresponding to the two different choices for k. The cardinality of $E(\mathbb{Z}/N_i\mathbb{Z})$

will be either $N+1-a$ or $N+1+a$, where a is the integer from $a^2+b^2|D| = 4N$ (we know this exists because of the requirement on the discriminant D).

When we have constructed a curve with correct cardinality we proceed by finding a point P of order q on this curve E. Then continuing with $N = N_{i+1}$.

References

[1] Adleman L., Manders K. and Miller G. 1977, *On taking roots in finite fields*, 18th Annual Symposium on Foundations of Computer Science, (october), pp. 175-178.

[2] Atkin A.O.L. and Morain F. 1993, *Elliptic curves and primality proving*, mathematics of computation volume 61, number 203, (july), pp. 29-68.

[3] Cohen H. and Frey G. et al. 2006, *Handbook of Elliptic and Hyperelliptic Curve Cryptography Scientific*, Chapman & Hall/CRC.

[4] Cohen H. and Lenstra Jr. H. W. 1984, *Primality Testing and Jacobi Sums*, Mathematics of Computation, Vol. 42, No. 165, (January), pp. 297-330.

[5] Dummit D.S. and Foote R. M. 2004, *Abstract algebra*, 3rd edition, John Wiley & Sons, inc.

[6] Goldwasser S. and Kilian J. 1999, *Primality Testing Using Elliptic Curves*, Journal of the ACM, Vol. 46, No. 4, (July), pp. 450-472.

[7] Herstein I.N. 1964, *Topics in Algebra*, Ginn and Company

[8] Hoffstein J., Pipher J. and Silverman J.H. 2014, *An introduction to mathematical cryptography*, 2nd edition, Springer-Verlag New York Inc.

[9] Lenstra Jr. H. W. 1987, *Factoring, integers with elliptic curves*, The Annals of Mathematics, Vol. 126, Issue 3, (november), pp. 649-673.

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

[11] Silverman J. H. 2009, *The Arithmetic of Elliptic Curves*, 2nd edition, Springer-Verlag New York Inc.

[12] Washington L. C. 2008, *Elliptic curves, Number theory and cryptography*, 2nd edition, Chapman & Hall/CRC.

[13] O'Connor J. J. and Robertson E. F. 2018, *https://mathshistory.st-andrews.ac.uk/HistTopics/Prime_numbers/* (accessed September 2021).