The CM Method

Travis Scholl

July 19, 2015

Abstract

Notes on [BSS99]

1 CM Method

(Choose D): The initial parameter for this method is some fixed negative fundamental discriminant $-D$, so in particular $D > 0$. We will construct a curve over a prime field with CM by an order in $K_D = \mathbb{Q}(\sqrt{-D})$.

(Choose p): Next we look for a prime p such that there exists a curve $E/\mathbb{F}_p$ with CM by the maximal order in $K = \mathbb{Q}(\sqrt{-D})$. Suppose there exists such a curve. Then the Frobenius endomorphism defines some element $\phi = \frac{x+y\sqrt{-D}}{2}$ (with $x, y \in \mathbb{Z}$) in $\mathbb{Z}_K$ with norm p (see [Sil09, Thm. V.2.3.1]). Hence

$$p = \left(\frac{x + y\sqrt{-D}}{2}\right)\left(\frac{x - y\sqrt{-D}}{2}\right) = \frac{x^2 + Dy^2}{4}$$

Note if $4 \mid D$ then both x, y need to be even because in this case $\mathbb{Z}_K = \mathbb{Z}[\sqrt{-D}]$.

In order to avoid supersingular curves, we will require $x \neq 0$. It turns out x will be the trace of the curve we want.

We will be interested in cases where p is large p does not ramify in K. So there are two cases: p is inert and p splits. If p is inert then we can not have such a solution to $4p = x^2 + Dy^2$. Therefore p must split, and moreover, p must split into principle ideals in $\mathbb{Q}(\sqrt{-D})$.

Remark 1.1. Primes will split quite often. It is a theorem that if L/K is a Galois extension, then the primes in K which split in L have density $1/[L : K]$ so we expect that about half of the primes in $\mathbb{Q}$ will split in $\mathbb{Q}(\sqrt{-D})$ (see [Mil13, Cor. 8.32]). Assuming that a "random" prime in $\mathbb{Q}$ factors into a "random" element of of the class group of K, then we expect to try $1/h_K$ primes where h_K is the class number of K, before finding a prime which factors into principle ideals in K. Therefore we expect to try $\frac{1}{2h_K}$ primes before finding one which splits into principle ideals in K.

Remark 1.2. The Brauer - Siegel Theorem implies that $h_K \sim \sqrt{D}$ asymptotically in D, meaning as D grows h_K will be similar to $\sqrt{D}$. Hence for large D this process could be quite slow. There is no known security vulnerability for curves with small class number, see [HVM04, Ch. 4.2.3, Pg. 179].

Checking whether a prime splits into principal ideals amounts to solving the Diophantine equation

$$4p = x^2 + Dy^2.$$

Note that if $D \equiv 0 \mod 4$ then this is equivalent to solving $p = u^2 + dv^2$ with d square free. Given p, d one can find u, v (if they exist) efficiently with Cornacchia's Algorithm (see [BSS99, Alg. VIII.1]). In the case when $-D \equiv 1 \mod 4$ it needs some slight modification, but it's more or less the same algorithm[1]. Hence determining whether p splits can be determined efficiently.

There is another condition p must satisfy. We will need the Hilbert class polynomial $H(x)$ to have a root mod p. This is because the j invariant of E (the curve we assumed to exist over $\mathbb{F}_p$ with CM given by the ring of integers of $\mathbb{Q}(\sqrt{-D})$) will be a solution to the Hilbert class polynomial mod p. This can be checked efficiently by precomputing the Hilbert class polynomial and then factoring it over $\mathbb{F}_p[x]$.

(Find j): The next step is to find the j-invariant of our curve.

Recall in the previous step we choose p so that the Hilbert class polynomial $H(x)$ has a root mod p. Since $H(x)$ is irreducible and separable, this gives a prime $\wp$ in the Hilbert class field H lying over p with ramification and inertia degree 1 (see [NS13, Prop. I.8.3]). Since $H(x)$ splits in H, we can choose j_0 to be a lift of the root mod p to a root in $\mathbb{Z}_H$.

First recall the following important and non-trivial theorem.

Theorem 1.3. *Let $K = \mathbb{Q}(\tau)$ be a quadratic imaginary field, H the Hilbert class field of K, and $H(x)$ be the Hilbert class polynomial (a certain polynomial generating H over K).*

Let $E/\mathbb{C}$ be an elliptic curve with complex multiplication by the ring of integers $\mathbb{Z}_K$ and $j(E)$ the j-invariant of E.

Then

(i) $H = K(j(E))$.

(ii) $H(x)$ *is the minimal polynomial for* $j(E)$ *over* $\mathbb{Z}$. *In particular,* $H_D \in \mathbb{Z}[x]$.

(iii) The roots $j_1, \ldots, j_h$ *of* $H(x)$ *are precisely the* j*-invariants of the elliptic curves (modulo isomorphisms) with complex multiplication by* $\mathbb{Z}_K$.

(iv) The orbit of $j(E)$ *under* $\mathrm{Gal}(H/K)$ *is a complete set of* j *invariants for elliptic curves (modulo isomorphisms) with CM by* $\mathbb{Z}_K$.

Proof. See [Sil94, Thm. II.4.1, Pg. 121]. □

In our setting, $K = \mathbb{Q}(\sqrt{-D})$. By the theorem, an elliptic curve over $\mathbb{C}$ with j-invariant equal to j_0 has CM by $\mathbb{Z}_K$. It is easy to write down an explicit formula for E given the j-invariant. For example the curve

$$E : y^2 = x^3 + 3c^2 \frac{j_0}{1728 - j_0} x + 2c^3 \frac{j_0}{1728 - j_0}$$

where c is any nonzero element in $\mathbb{F}_p$ (we need to assume $j \neq 0, 1728$), has j-invariant j_0. We may choose c so that the coefficients are all algebraic integers. Therefore this is a curve defined over $\mathbb{Z}_H$ with CM given by $\mathbb{Z}_K$.

[1] See `http://projecteuclid.org/download/pdf_1/euclid.pja/1116442240`

Remark 1.4. It is worth remarking this shows that a curve can always be naturally defined over the ring of integers a field containing the j-invariant, modulo the usual restrictions: this curve is isomorphic to the original over the algebraic closure (where the j-invariant parameterizes isomorphism classes), and this model does not work for $j = 0, 1728$ (however we can write down models for these separately[2]).

Now we can reduce this curve mod the prime $\wp$ to get an elliptic curve $\tilde{E}$ defined over $\mathbb{Z}_H/\wp \cong \mathbb{F}_p$ (since $\wp$ has inertia degree 1 over p).

The claim is that $\tilde{E}$ has complex multiplication by precisely $\mathbb{Z}_K$. This follows from assuming that $\tilde{E}$ is not supersingular (which is rare and only happens when the x from $4p = x^2 + Dy^2$ is 0, see [Sil09, Ex. 5.10b]) and the fact that the natural map $\mathrm{End}(E) \to \mathrm{End}(\tilde{E})$ is an injection. Then since $\mathrm{End}(\tilde{E})$ is an order in the ring of integers of some number field and contains the ring of integers of K, it must be equal to $\mathbb{Z}_K$.

Note that the equation of $\tilde{E}$ only requires to know the value of j_0 modulo $\wp$ which was just the root of $H(x)$ mod p. So we only need to know the root of $H(x)$ mod p.

(Find $\tilde{E}$) There is one last step. At this point we have some fundamental discriminant $-D$, a prime p, a solution of

$$4p = x^2 + Dy^2$$

and a root j_0 (we now use j_0 as the element in $\mathbb{F}_p$ as opposed to above when it lived in $\mathbb{C}$) of $H(x)$ mod p. We generated a curve $E/\mathbb{F}_p$ with j-invariant j_0 and CM by $\mathbb{Z}_K$.

By construction Frobenius endomorphism corresponds to an element of degree p which is $\frac{x \pm y\sqrt{-D}}{2}$. Hence the trace of Frobenius is $\pm x$. This means the number of points on E is given by

$$\#E = p + 1 \pm x$$

The reason there are two values is because there is the quadratic twist of E. This curve can be written down explicitly by changing the c we used in the previous definition by a quadratic non-residue mod p. Let E' be the quadratic twist of E, note that it has the same j-invariant and will have complex multiplication by $\mathbb{Z}_K$ as well. In this case the Frobenius endomorphism on the twist by $\frac{x \mp y\sqrt{-D}}{2}$.

So if $m = P + 1 + x$ or $m' = P + 1 - x$ is an acceptable number of points, then we need to figure out which curve is which. This can be done efficiently counting points using standard algorithms, or by choosing a random point P on E and computing $[p + 1 + x]P$. If this is 0 and $[p + 1 - x]P$ is not, then you know which curve is which.

2 Example

Fix $D = 532$. Note $-D$ is a fundamental discriminant because $-D/4 = -133 \equiv 3 \mod 4$.

To find a prime p, we randomly pick primes of about 100 bits until we find one that satisfies all the conditions. The equation $4p = x^2 + Dy^2$ reduces in this case to $p = u^2 + Dv^2$ where $x = 2u$. Let $m = p + 1 \pm 2u$ which will be the number of points on the curve or its quadratic twist. Thus we want a solution such that

- The Hilbert class polynomial $H(x)$ has a solution mod p.

[2]The curve $y^2 = x^3 - 1$ has j-invariant 0 and $y^2 = x^3 - x$ has j-invariant 1728

- m has a large prime factor to prevent small subgroup attack.
- $m \neq p + 1$ to avoid supersingular curves (which have smaller embedding degree).
- No small value of k such that $p^k \equiv 1 \mod m$ to avoid the MOV/Weil pairing attack, see [HVM04, Pg. 169].
- $m \neq p$ to avoid a trace 1 curve where ECDLP is trivial, see [Sma99].

Remark 2.1. We actually do not need to explicitly check if E is supersingular because of the fact that supersingular elliptic curves have embedding degree $k \leqslant 6$. Therefore as long as we check that the embedding degree is large (which is the typical case, see [Sil09, Rem. XI.6.3, Pg. 388]), we will always avoid supersingular curves. This also excludes curves of trace 0 and trace 2 (see [BSS99, Ch. V.7]).

Remark 2.2. In most cases it's enough to check if the embedding degree satisfies $k \geqslant 12$.

This is because with these k values the standard sub-exponential algorithms on the finite field p^k are still more expensive then the standard exponential algorithms on the prime order subgroup of our elliptic curve. A general number field sieve algorithm can solve the DLP in $\mathbb{F}_{p^k}^*$ in $L[1/3, c] = e^{c(\ln p^k)^{1/3}(\ln \ln p^k)^{2/3}}$ with $c = (64/9)^{1/3}$. On the other hand, the best attack on a general elliptic curve $E/\mathbb{F}_p$ with a large prime order subgroup is the Pollard-Rho algorithm or baby-step giant-step which has an exponential complexity of $L[1, 1/2]$ or about $\sqrt{p}$. Note that really the ECDLP algorithms time complexity is really based off the largest prime ℓ ordered subgroup. If we assume $E(\mathbb{F}_p)$ has a small cofactor (that is $\#E(\mathbb{F}_p)/\ell \approx O(1)$) then by the Hasse bound $\ell \approx \#E(\mathbb{F}_p) \approx p$.

Thus in order to be safe against the MOV attack, if p is about 150 bits it would only take an embedding degree of about 4 in order for the MOV attack to be slower than the general ECDLP. Again, this is forgetting about any constants and optimizations.

Remark 2.3. The L-notation above is used to measure time complexity of subexponential algorithms. In general, given an input size of n and ignoring constant factors,

$$L[\alpha, c] = e^{c(\ln n)^{\alpha}(\ln \ln n)^{1-\alpha}}$$

If $\alpha = 1$ then the algorithm is exponential in $\ln n$ (usually we measure algorithms by the number of bits in n). If $\alpha = 0$ it is polynomial.

We do this with the following Sage code.

First we implement Cornacchia's algorithm.

```
class NoSolutionError(Exception):
    pass
def Cornacchia(p,d):
    """
        solves (if possible) the equation
    p = x^2 + dy^2
        Assumes d is squarefree and p is prime

        EXAMPLES:

            sage: d = 21
            sage: p = 337
            sage: Cornacchia(p,d)
            (1, 4)
    """
    assert is_prime(p), "p must be prime"
    assert is_squarefree(d), "d must be square free"
    x0 = p
    try:
        x1 = Integer(mod(-d,p).sqrt())
    except:
        raise NoSolutionError('-d must have a sqrt mod p')
    x1 = x1 if x1 <= p/2 else p - x1
    while x1^2 >= p:
        x2 = x0%x1
        x0 = x1; x1 = x2
    s = (p-x1^2)/d
    if s.is_square():
        return (x1,sqrt(s))
    else:
        raise NoSolutionError('no solution')
```

Then we run the CM method. The timing information was collected from running the script on a Sage worksheet.

```
D = 532
H = QuadraticField(-D).hilbert_class_polynomial()
def CM():
    global E1,E2,p,j, m1
    while True:
        p = random_prime(2^150,2^151)
        # check H(x) has a solution mod p which is not 0 or 1728
        roots = [r[0] for r in H.change_ring(GF(p)).roots() if r[0] != 0 and r[0] != 1728]
        if len(roots) == 0:
            continue
        # check for solution 4p = x^2 + Dy^2
        try:
            u,v = Cornacchia(p,D/4)
        except NoSolutionError:
            continue
        trace = 2*u
        m1 = p + 1 + trace
        m2 = p + 1 - trace
        # check for trace 1
        if m1 == p or m2 == p:
            continue
        # check for large prime factor
        if max([len(l[0].bits()) for l in factor(m1)]) < 80 or \
            max([len(l[0].bits()) for l in factor(m2)]) < 80:
            continue
        # check for small embedding degree
        if mod(p,m1).multiplicative_order() < 12 or \
            mod(p,m2).multiplicative_order() < 12:
            continue
        # print acceptable paramaters
        j = roots[0]
        c = j / (1728 - j)
        E = EllipticCurve(GF(p),[0,0,0,3*c,2*c])
        if E.count_points() == m1:
            E1 = E
            E2 = E.quadratic_twist()
        else:
            E2 = E
            E1 = E.quadratic_twist()
        print E1
        print E2
        break
%time CM()
```

```
Elliptic Curve defined by y^2 = x^3 + 92250472838205334983608860081062755668496766 6*x + \
121709915454822140608555307140053224646085286 over Finite Field of size \
113694452136093203778608028424814750584405356 1
Elliptic Curve defined by y^2 = x^3 + \
896013149634954889359565923133800958773804357*x + \
597342099756636592906377282089200639182536238 over Finite Field of size \
113694452136093203778608028424814750584405356 1
CPU time: 6.19 s, Wall time: 6.82 s
```

Remark 2.4. The parameters used above are made up for exposition. To compare with actual parameters, the NIST curve P-192 (see [GK13, D.1.2.1, Pg. 90]) has a largest prime factor of 192 bits, and a 189 bit embedding degree. It should be noted that the embedding degree is often large (on the order of the size of the group). There is many optimizations that could be made to the above algorithm, yet it only takes less than a minute to find curves of similar size as P-192.

References

[BSS99] I.F. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography*. Lecture note series. Cambridge University Press, 1999.

[GK13] P Gallagher and C Kerry. Fips pub 186-4: Digital signature standard, dss, 2013.

[HVM04] D. Hankerson, S. Vanstone, and A.J. Menezes. *Guide to Elliptic Curve Cryptography*. Springer Professional Computing. Springer, 2004.

[Mil13] James S. Milne. Algebraic number theory (v3.05), 2013. Available at www.jmilne.org/math/.

[NS13] J. Neukirch and N. Schappacher. *Algebraic Number Theory*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2013.

[Sil94] J.H. Silverman. *Advanced Topics in the Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer New York, 1994.

[Sil09] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer, 2009.

[Sma99] N. P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12(3):193–196, 1999.

Travis Scholl
Department of Mathematics, University of Washington, Seattle WA 98195
email: tscholl2@uw.edu