Generating Elliptic Curves of Prime Order[⋆] [⋆⋆] [⋆⋆⋆]

Erkay Savaş[1], Thomas A. Schmidt[2], and Çetin K. Koç[1]

[1] Department of Electrical & Computer Engineering
Oregon State University, Corvallis, Oregon 97331, USA
{savas,koc}@ece.orst.edu

[2] Department of Mathematics
Oregon State University, Corvallis, Oregon 97331, USA
toms@math.orst.edu

Abstract. A variation of the Complex Multiplication (CM) method for generating elliptic curves of known order over finite fields is proposed. We give heuristics and timing statistics in the mildly restricted setting of prime curve order. These may be seen to corroborate earlier work of Koblitz in the class number one setting. Our heuristics are based upon a recent conjecture by R. Gross and J. Smith on numbers of twin primes in algebraic number fields.

Our variation precalculates class polynomials as a separate off-line process. Unlike the standard approach, which begins with a prime p and searches for an appropriate discriminant D, we choose a discriminant and then search for appropriate primes. Our on-line process is quick and can be compactly coded.

In practice, elliptic curves with near prime order are used. Thus, our timing estimates and data can be regarded as upper estimates for practical purposes.

1 Introduction

An important category of cryptographic algorithms is that of the elliptic curve cryptosystems defined over a finite field $\mathbb{F}_p$, see [9] for a recent overview. While there are many methods proposed for performing fast elliptic curve arithmetic, there is a paucity of efficient means for generating suitable elliptic curves. The methods proposed to date for curve generation mainly necessitate implementing complex and floating point arithmetic with high precision. However, this hinders the implementation of the proposed algorithms on simple processors with limited amounts of memory. In [13], Miyaji proposed a practical approach to construct

⋆ This research was supported by rTrust.

⋆⋆ The reader should note that Oregon State University has filed US and International patent applications for inventions described in this paper.

⋆⋆⋆ *Cryptographic Hardware and Embedded Systems - CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, editors, Lecture Notes in Computer Science No. 2162, pages 145-161, Springer Verlag, Berlin, Germany, May 13-16, 2001.

"anomolous" elliptic curves; these elliptic curves, of order p over fields of characteristic p, have since been shown to be insecure, [14], [16], [19]. However, the idea of the construction can be applied to quickly find non-anomolous curves as well. We present such a variant of the method to construct elliptic curves of known prime orders. Our variant has less computational complexity in its online implementation than that proposed in the IEEE standards [7]. Heuristics and calculations show that our method is practical.

Timing estimates for the Complex Multiplication (CM) method of generating elliptic curves seem difficult to find in the public literature. The above mentioned survey article [9] mentions that in practice the method is fast, but cites a timing result for a single curve. Our timing statistics are averaged over 1000 curves per discriminant. As to previous theoretical bounds on running times, it seems that Koblitz's [8] conjectures and statistics for reduction of class number one CM curves defined over the rationals are taken to indicate that the CM method is in general speedy. We concur in our 6.3.

We thank the referees for helpful comments and for pointing us to important entries in the literature. The second-named author thanks Professor G. Frey and his team at the Institut für Experimentelle Mathematik for clarifying some of the basics of the theory of elliptic curves.

The paper is organized as follows. Section 2 summarizes the complex multiplication curve generation method. In section 3, we explain our variant which requires less data size and computation, while avoiding the weakness of Miyaji's method. Section 4 summarizes the method to construct the class polynomials, the most computationally intensive part of the CM method. In our approach, we pre-calculate a set of these and store the coefficients. Also in section 4, we give some experimental results which indicate the efficiency of our approach. In section 5, we provide more detailed implementation results. In section 6, we give heuristics for the number of trials necessary to find prime order elliptic curves. Section 7 is a brief conclusion.

2 Complex Multiplication Curve Generation Algorithm

For the ease of the reader, we summarize some basics of the theory of elliptic curves.

An elliptic curve $\mathcal{E}$ defined over a finite field $\mathbb{F}_p$, where $p > 3$, can be given as

$$\mathcal{E}(\mathbb{F}_p) : y^2 = x^3 + ax + b \qquad a, b \in F_p \tag{1}$$

Associated with $\mathcal{E}$, there are two important quantities:
the discriminant

$$\Delta = -16(4a^3 + 27b^2) \tag{2}$$

and the j-invariant

$$j = 1728(4a)^3/\Delta \tag{3}$$

where $\Delta \neq 0$.

Lemma 1. *Given $j_0 \in \mathbb{F}_p$ there is an elliptic curve, $\mathcal{E}$, defined over $\mathbb{F}_p$ such that $j(\mathcal{E}) = j_0$.*

An elliptic curve with a given j-invariant j_0 is constructed easily. We consider $j_0 \notin \{0, 1728\}$; these special cases are also easily handled. Let $k = j_0/(1728 - j_0)$, $j_0 \in \mathbb{F}_p$ then the equation

$$\mathcal{E}\colon y^2 = x^3 + 3kx + 2k \tag{4}$$

gives an elliptic curve with j-invariant $j(\mathcal{E}) = j_0$.

Theorem 1. *Isomorphic elliptic curves have the same j-invariant.*

Theorem 2. *(Hasse) Let $\#\mathcal{E}(\mathbb{F}_p)$ denote the number of points on the elliptic curve $\mathcal{E}(\mathbb{F}_p)$. If $\#\mathcal{E}(\mathbb{F}_p) = p + 1 - t$, then $|t| \leq 2\sqrt{p}$.*

Definition 1. *(Twist) Given $\mathcal{E}\colon y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_p$ the* twist *of $\mathcal{E}$ by c is the elliptic curve given by*

$$\mathcal{E}_c : y^2 = x^3 + ac^2x + bc^3 \tag{5}$$

where $c \in \mathbb{F}_p$.

Theorem 3. *Let $\mathcal{E}$ be defined over $\mathbb{F}_p$ and its order be $\#\mathcal{E}(\mathbb{F}_p) = p + 1 - t$. Then the order of its twist is given as*

$$\#\mathcal{E}_c(\mathbb{F}_p^*) = \begin{cases} p + 1 - t \text{ if } c \text{ is square in } \mathbb{F}_p \\ p + 1 + t \text{ if } c \text{ is non-square in } \mathbb{F}_p \end{cases} \tag{6}$$

For the above basics of elliptic curves, we refer to [18]. The following result is based upon work of M. Deuring in the 1940s. See [1] and [10].

Theorem 4. *(Atkin-Morain) Let p be an odd prime such that*

$$4p = t^2 + Ds^2 \tag{7}$$

for some $t, s \in \mathbb{Z}$. Then there is an elliptic curve $\mathcal{E}$ defined over $\mathbb{F}_p$ such that $\#\mathcal{E}(\mathbb{F}_p) = p + 1 - t$.

An integer D which satisfies (7) for a given p is called a *CM discriminant* of p. Indeed, the curve $\mathcal{E}$ has complex multiplication by the integers of $\mathbb{Q}(\sqrt{-D})$. Given such a D for a prime p, the j-invariant of the elliptic curve can be calculated due to class field theory. Once the j-invariant is known, the elliptic curve with $p+1-t$ points is easily constructed utilizing Lemma 1. Actually, the method gives an elliptic curve with either $p + 1 - t$ or $p + 1 + t$ points. If the constructed elliptic curve has $p + 1 + t$ points, then one must take the twist of this elliptic curve to obtain an elliptic curve with $p + 1 - t$ points. Fortunately, it is trivial to construct the desired curve when its twist is known, due to Theorem 3. This technique for constructing elliptic curves of known order is called the *Complex Multiplication* (CM) method.

A detailed explanation of the CM method is given in the P1363 standards. One can also profitably refer to [2]. We summarize the method in the following:

1. Given a prime number p, find the smallest D in (7) along with t (s is not needed in the computations).
2. The orders of the curves which can be constructed are $\#\mathcal{E}(\mathbb{F}_p) = p + 1 \pm t$. Check if one of the orders has an admissible factorization (by admissible factorization we mean a prime or nearly prime number as defined in the standards). If not, find another D and corresponding t. Repeat until an order with admissible factorization is found.
3. Construct the class polynomial $H_D(x)$ using the formulas given in the standards. (The class polynomial for a D is a fixed monic polynomial with integer coefficients. In particular, it is independent of p).
4. Find a root j_0 of $H_D(x) \pmod p$. This j_0 is the j-invariant of the curve to be constructed.
5. Set $k = j_0/(1728-j_0) \pmod p$ and the curve will be $\mathcal{E}$: $y^2 = x^3 + 3kx + 2k$.
6. Check the order of the curve. If it is not $p + 1 - t$, then construct the twist using a randomly selected nonsquare $c \in \mathbb{F}_p$.

With the CM method, one may first fix a prime number p, and thereafter construct an elliptic curve over $\mathbb{F}_p$. This has the possible advantage of allowing the use prime numbers of special forms, possibly permitting an improvement in efficiency of the underlying modular arithmetic for the curve operations. On the other hand, the method is efficient only when the degree of the class polynomial is small; in general, factoring a high degree polynomial is time consuming. Furthermore, the construction of the class polynomials requires multi-precision floating-point and complex number arithmetic.

3 A Variant of the CM Method

The variant is straightforward: Construct and store the corresponding class polynomials for D in $\mathcal{D}$ and search for primes whose CM discriminants are in this set. We thus avoid repeatedly calculating class polynomials; hence multi-precision floating and complex number arithmetic as well as the factorization of high degree class polynomials is avoided. Indeed, the original CM method as specified in the standards becomes inefficient if not impractical as the class polynomial degree becomes large.

Our algorithm is thus:

1. Off-line: Determine a set $\mathcal{D}$ of CM discriminants such that the corresponding class numbers are small.
2. Off-line: Calculate and store the class polynomials of CM discriminants in $\mathcal{D}$.
3. Select randomly a CM discriminant D in $\mathcal{D}$ and obtain the corresponding class polynomial $H_D(x)$.
4. Search for prime number p satisfying the equation $4p = t^2 + Ds^2$. (First, we select random t and s values of appropriate sizes and then determine if p is prime)

5. Compute $u_1 = p+1-t$ and $u_2 = p+1+t$, the orders of the candidate elliptic curves and determine if either of them has an admissible factorization (i.e. is a prime or nearly-prime number). If not, go to Step 4 and pick another random pair of t and s.
6. If u_1 has proper factorization set $u = q_1$, otherwise $u = q_2$.
7. Find a root j_0 of $H_D(x) \bmod p$ (this is the j-invariant of the curve).
8. Set $k = j_0/(1728 - j_0) \bmod p$ and the curve of order u_1 or u_2 will be

$$\mathcal{E}_c : y^2 = x^3 + ax + b \tag{8}$$

   where $a = 3kc^2$, $b = kc^3$ and $c \in \mathbb{F}_p$ is randomly chosen.
9. Check the order of the curve. If it is u then stop. Otherwise, select a non-square number $e \in \mathbb{F}_p$ and calculate the twist by e, $\mathcal{E}_e(F_p) = x^3 + ae^2 + be^3$.

Our experiments and heuristics confirm that pairs p and u of the type sought can be found quickly.

As stated in the introduction, the above is a generalizing variation of Miyaji's simplification of the general CM method. Recently, A.K. Lenstra [11] has also suggested using restricted sets of discriminants. But, as Miyaji, Lenstra only considers the class number one candidate discriminants.

4 Constructing Class Polynomials

Although there are different methods to calculate class polynomials, we adopt that of [1], see also [4]. Let $D = b^2 - 4ac$ be the discriminant of a quadratic form

$$f(x, y) = ax^2 + bxy + cy^2$$

where a, b, c are integers. The quadratic form, $f(x, y)$ is commonly represented by the compact notation $[a, b, c]$. If the integers a, b, c have no common factor, then the quadratic form $[a, b, c]$ is called *primitive*. There are infinitely many quadratic forms of any possible discriminant. We reduce to a finite number by demanding that a root of $f(x, 1)$ lie in a certain region of the complex plane. Let the primitive quadratic form $[a, b, c]$ be of negative discriminant. Let τ be the root of $f(x, 1)$ which lies in the upper half-plane:

$$\tau = (-b + \sqrt{D})/2a$$

The $[a, b, c]$ is a *reduced form* if τ has complex norm greater than or equal to 1, and $\Re(\tau) \in [-1/2, 1/2]$. Given a discriminant $D < 0$, we can easily find all of the reduced quadratic forms of discriminant D. We then compute the class polynomial $H_D(x)$ which is the minimal polynomial of the $j(\tau)$. For each value of τ, the j-value (denoted j_i below) is computed as follows:

$$j(\sqrt{D}) = (256f(\tau) + 1)^3/f(\tau)$$

where

$$f(\tau) = \Delta(2\tau)/\Delta(\tau),$$

$$\Delta(\tau) = q \cdot [1 + \sum_{n \geq 1} (-1)^n (q^{3n(n+1/2} + q^{3n(n-1/2)})]^{24},$$

and

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

Finally, the class polynomial can be constructed by using the following formula:

$$H_D(x) = \prod_{i=1}^{h} (x - j_i)$$

where h is the number of the reduced forms of D, commonly known as the *class number* of D. Since $H_D(x)$ has integer coefficients one must use sufficient accuracy during the computations.

Our approach, as stated earlier, is to construct class polynomials beforehand for given D values. We do this using some software tool specialized for mathematical calculations. In our implementation, we use Maple. Following [1], we set the precision for floating point arithmetic as follows:

$$\text{precision} = 10 + \left(\begin{array}{c} h \\ \lfloor h/2 \rfloor \end{array} \right) \cdot \pi \sqrt{D} \cdot \sum_{i=1}^{h} 1/a_i \ ,$$

$$N = 10 + \left(\begin{array}{c} h \\ \lfloor h/2 \rfloor \end{array} \right) \cdot \sum_{i=1}^{h} 1/a_i \ .$$

Here N gives the number of terms to keep in the calculations involving the various $\Delta(\tau)$.

As stated earlier, other methods than the basic use of the j-function applied here can be employed to construct class polynomials. In each of these, one obtains some class-invariant polynomial for the CM discriminant D. One advantage of using different methods is to obtain class polynomials with relatively small integer coefficients. This is particularly important when the processor used to store the polynomial coefficients is of limited memory.

5 Implementation Results

We implemented the algorithm using the NTL number theory and algebra package [17] on a 450-MHz Pentium II based PC. We restricted to $t = 2v + 1$ and $s = 2w + 1$ where $v, w \in \mathbb{Z}$. Thus, the prime numbers found in this setting are of the form

$$p = v^2 + v + (w^2 + w)D + \frac{D+1}{4} \tag{9}$$

where D satisfies

$$D \equiv 3 \pmod 4.$$

Furthermore, D is chosen such that $(D+1)/4$ is odd, hence p is odd for any choice of v and w. Throughout, we avoid the imaginary quadratic field of exceptionally many units: We exclude $D = 3$. We obtained average times to find the prime p and prime u as well as to calculate the corresponding curve for the following values of D. Again, were u merely required to be nearly prime number, the search time for admissible pairs would decrease.

For

$$\mathcal{D}= \{\, 163,\, 403,\, 883\},$$

the corresponding class polynomials are given in the following:

$$H_{163}(x) = x + 640320;$$

$$H_{403}(x) = x^2 - 10884420340249105583308800000\, x$$

$$+24528113892293313919795200000;$$

$$H_{883}(x) = x^3 + 16799028538162731818757552080012338790400000000\, x^2$$

$$-15196011112524528203387561952912447897600000\, x$$

$$+34903934341011819039224295011933392896000.$$

We obtained efficiency results for these three cases. When the class number is one, the class polynomial is of degree one; hence the root is obtained without any computation. In the two other cases, we must determine a root for each p of the quadratic or cubic polynomial, respectively. The results are given in Table 1.

Table 1: Timings to build curves of known prime order.

D	class no	bitsize	Average time (s)	N_p	N_u
163	1	192	1.22	23	11
163	1	224	2.29	27	14
403	2	192	1.57	30	14
403	2	224	3.29	36	21
883	3	192	1.63	30	14
883	3	224	3.01	36	19

To find a root modulo p of a class polynomial takes approximately a constant time determined by the size of the modulus p and the degree of the polynomial. However, the time or the number of trials to find admissible pairs of p and u is of a more complicated nature. We have run our program repeatedly to build 1000 different curves with each value of D in Table 1. In the table, N_p indicates the approximate number of random pairs of v and w to be tried before a prime $p = v^2 + v + (w^2 + w)D + (D + 1)/4$ is found. Similarly, N_u is the average trial number of p of the form (9) to obtain a prime u.

The method remains efficient for larger class numbers, as shown in Table 2 and Figure 5.

Table 2: Timings to build curves of prime order for large class numbers.

bitsize	D	class no	Average time (s)	N_p	N_u
192	555	4	3.54	51	35
	1051	5	2.78	48	26
	451	6	5.70	86	57
	811	7	4.61	76	44
	1299	8	5.91	69	59
	1187	9	7.35	79	72
	611	10	12.53	126	128
	1283	11	9.42	99	92
	1235	12	10.62	107	104
	1451	13	11.08	106	108
	1211	14	14.22	124	142
	1259	15	15.61	132	154
	1379	16	13.54	135	131
	1091	17	17.46	159	168
	1691	18	15.35	136	146
	2099	19	14.64	128	139
	1739	20	17.45	150	166
	25259	72	23.20	140	160
	37571	95	24.90	152	157

Figure 1: Performance of the method with increasing class numbers.

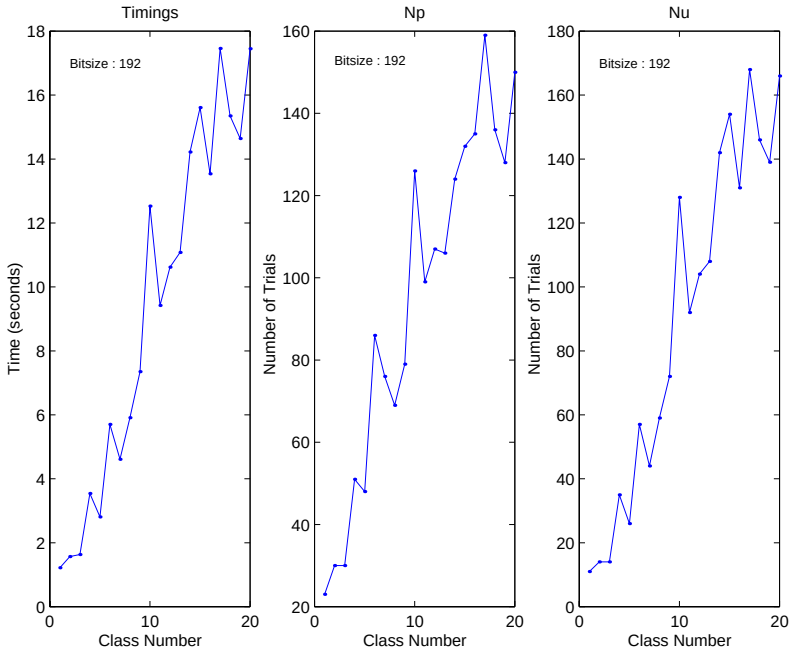


Table 2 clearly indicates that the admissible pair search time increases with the class number. Although this increase is not monotone — the timing for class number 10 is much higher than those for class numbers 11, 12, and 13 — it is reasonable to claim that the time needed to find proper pairs is directly proportional to the class number. This result is consistent with the theoretical considerations in [12]; see the next section for specific comments. The dependence of the construction process on the particular value of D seems to account for the deviation from simple monoticity. Note also, just as the theoretical heuristics of the next section suggest, that the time to find an admissible pair (p, u) decreases with the size of D. This can be observed in Table 3. See also the Figures 2, 3, 4, 5, 6, 7.

Table 3: Timings for various class numbers.

field type		*bitsize 192*			*bitsize 224*		
class no	D	Average time (s)	N_p	N_u	Average time (s)	N_p	N_u
1	11	9.10	95	94	16.20	109	113
	19	3.86	68	39	7.15	81	49
	43	2.30	46	23	4.19	55	28
	67	1.87	37	18	3.55	44	23
	163	1.22	23	11	2.29	27	14
2	35	10.38	105	108	15.74	120	110
	123	3.49	57	35	5.93	64	40
	187	2.42	45	23	4.31	52	28
	235	2.09	40	20	3.98	48	26
	403	1.57	30	14	3.29	36	21
3	59	11.37	121	118	21.17	141	128
	83	10.01	102	104	16.93	118	117
	107	7.90	92	82	14.33	106	99
	379	2.63	47	25	4.85	56	32
	883	1.63	30	14	3.01	36	19
4	155	9.50	99	99	16.14	116	112
	195	6.46	88	66	11.90	105	82
	259	4.77	78	49	8.46	91	58
	355	3.76	64	37	6.87	77	46
	555	3.54	51	35	6.54	63	44
5	179	11.54	113	119	20.65	140	142
	227	9.33	103	97	17.42	122	120
	347	7.64	83	79	12.64	98	86
	443	6.65	73	68	11.81	86	81
	1051	2.78	48	26	5.52	55	36

Figure 2: Timings to build curves with increasing discriminants.

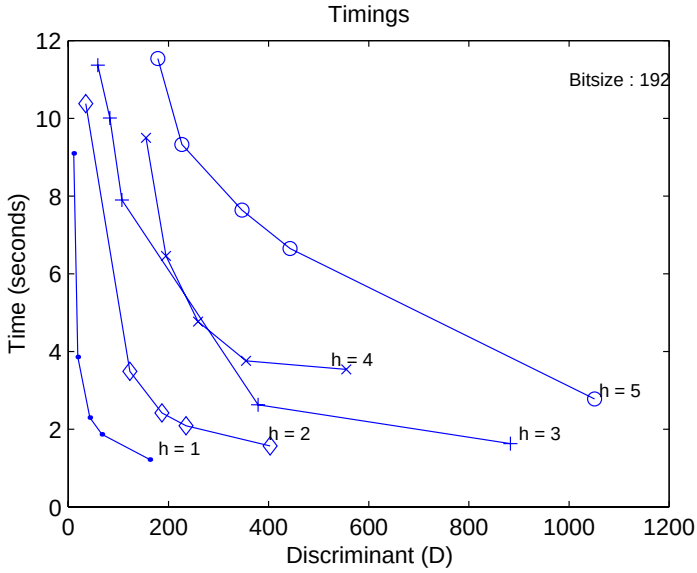


Figure 3: Number of trials for p with increasing discriminants.

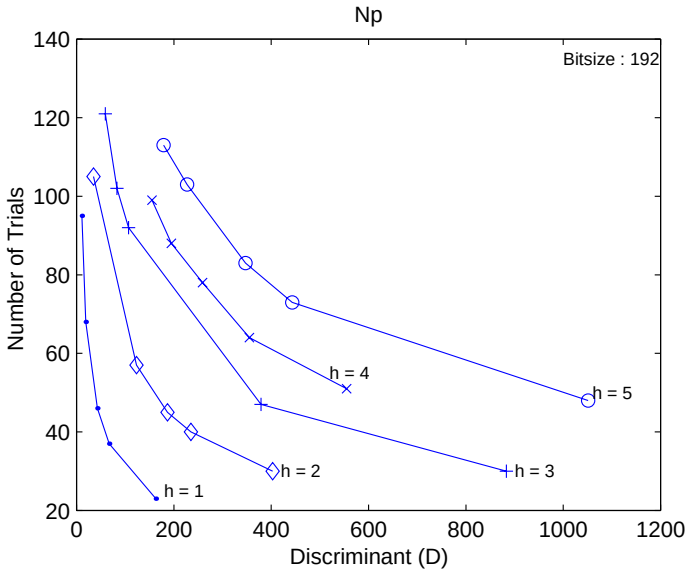


Figure 4: Number of trials for u with increasing discriminants.

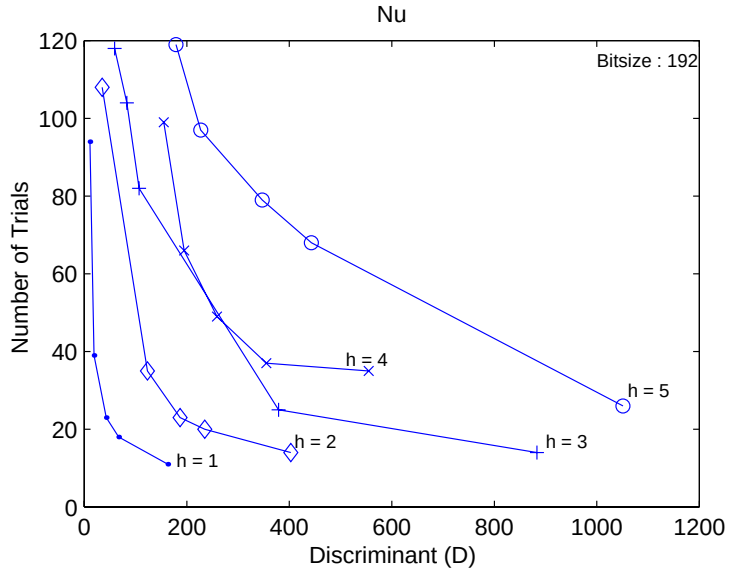


Figure 5: Timings to build curves with increasing discriminants.

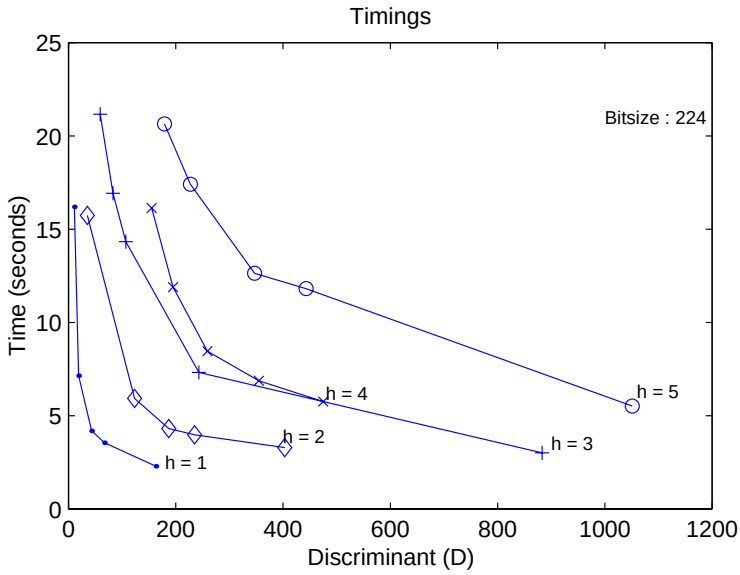


Figure 6: Number of trials for p with increasing discriminants.

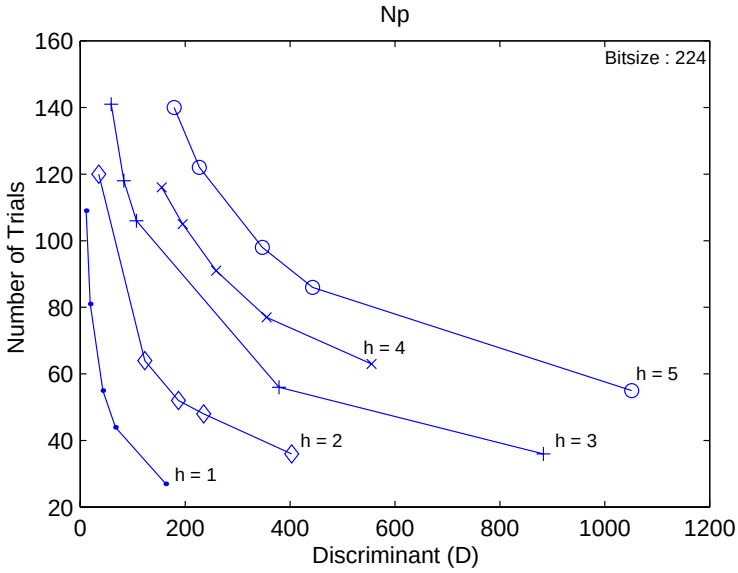
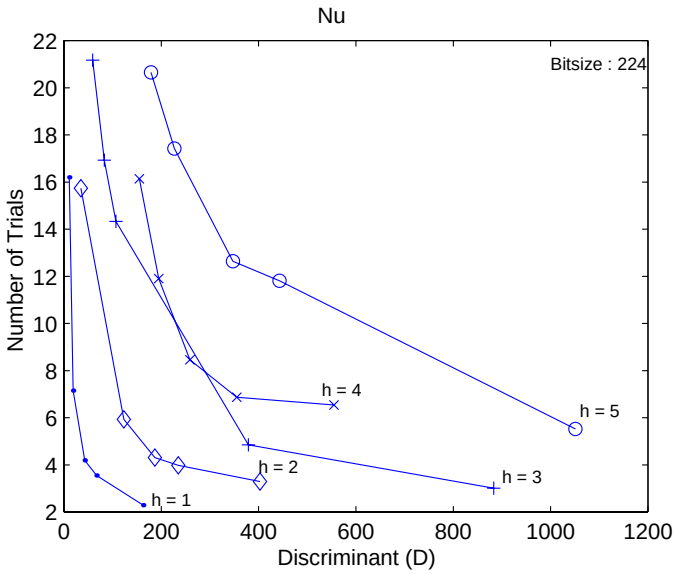


Figure 7: Number of trials for u with increasing discriminants.



Another important implementation aspect is code size. While one implementation [15] of the full CM method [7] requires 204KB on a PC with Windows NT, our implementation with NTL requires only 164KB code space on the same platform. In fact, the code space can be made much smaller when code is writ-

ten expressly for curve generation. For sake of simplicity, we have written such a program which treats only the class number one case. We found that only an extra 10 KB of object code space is needed for curve generation routines (assuming that the basic subroutines for arithmetic operations needed for elliptic curve arithmetic are already available).

6 Heuristics: Twin Primes and Prime Order Elliptic Curves

6.1 Finding Primes

The Prime Number Theorem states that for sufficiently large M, the number of primes in $[2, M]$ is approximately $M/\ln M$. But, with D as chosen, $4p = t^2 + s^2 D$ expresses that p is a norm of an element in the ring of integers $\mathbb{Q}(\sqrt{-D})$. The density of rational primes which are of this type is $1/(2h_D)$, where h_D is the class number of $\mathbb{Q}(\sqrt{-D})$. See [2–4]. We thus have that some $M/(2h_D \ln M)$ primes of size up to M are of our type.

With $p \leq M$, each pair $(s, t) \in \mathbb{Z}^2$ gives an integral lattice point inside the ellipse of equation $t^2 + s^2 D = M/4$. Gauss, see for example [3], found an asymptotic formula for the number of lattice points interior to an ellipse. Here, this gives that the the number of the lattice points (s, t) with s, t both positive is $L(M) = \pi(M)\sqrt{D} + O(\sqrt{M})$. Furthermore, our p are odd, we work with odd D and we desire the elliptic curve order $u = p + 1 \pm t$ to be prime, hence certainly odd. We thus only consider s and t odd. We thus search through a possible $L(M)/4$ distinct values of $t^2 + s^2 D$ for (s, t) interior to the ellipse.

We search for prime p in specific ranges of the form $[S, 2S]$, and hence expect to have find a prime p after a total number of trials of (v, w) of some $\bar{N}_p := c(\pi h_D \ln S)/\sqrt{D}$, for some constant c. Our experimental data confirms this, see Tables 1,2,3, where S is variously 2^{191} and 2^{223}.

6.2 Prime Order Elliptic Curves and Twin Primes

The order of the curve we seek is $u = p + 1 \pm t$, we ask for it to be prime. Now, p of our form is the norm of the element $\mathcal{P} = (t + s\sqrt{-D})/2$; note that t is the trace of $\mathcal{P}$. The norms of $\mathcal{P} \pm 1$ are easily seen to be the two possibilities for u. Thus, we are seeking twin pairs $(\mathcal{P}, \mathcal{P} \pm 1)$. Indeed, the theory of complex multiplication ensures that associated to each pair of this form is an elliptic curve defined over $\mathbb{F}_p$ where p is the norm of $\mathcal{P}$ and whose exact number of points over this field equals the norm of $\mathcal{P} \pm 1$.

Although it is not known if there are infinitely many twin prime (principal ideal) pairs in any quadratic field, there are conjectures as to their numbers within bounded regions. This is also the case for twin rational primes, for which Hardy and Littlewood [6] conjectured that there are some $C_2 \int_2^M 1/(\ln y)^2 \, dy$ twin primes of size less than M, with $C_2 = 2 \prod_{\text{odd prime } p} 1 - 1/(p-1)^2$. This

constant is approximately 1.32032. The integral $\int_2^M 1/(\ln y)^2\,dy$ is $M/(\ln M)^2 \times \gamma(M)$, where $\gamma(M)$ is $(1+2!/\ln M+3!/(\ln M)^2+\cdots+n!/(\ln M)^{n-1})+O((\ln M)^{n-1})$.

Recently, Gross and Smith [5] have stated general conjectures for the number of twin primes in algebraic number fields. For $\mathbb{Q}(\sqrt{-D})$ with D congruent to 3 modulo 8, their conjecture is that the number of twin primes of norm less than M should be

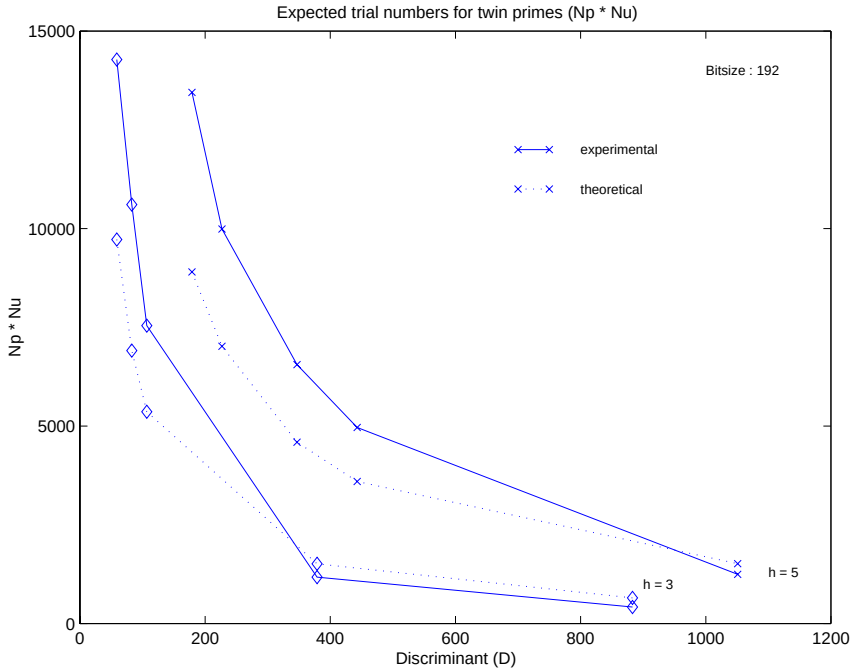
$$P(D,M) = 2\sqrt{D}/(\pi h_D^2) \times \beta(D) \times \int_2^M 1/(\ln y)^2\,dy\ ,$$

with $\beta(D) = \prod_{\mathcal{Q}} (1 - 1/(\,N(\mathcal{Q}) - 1))^2$ where $\mathcal{Q}$ runs through the prime ideals of $\mathbb{Q}(\sqrt{-D})$ and $N(\mathcal{Q})$ denotes the norm to $\mathbb{Z}$. We thus see that the number of (v,w) which lead to elliptic curves of prime order over a prime field $\mathbb{F}_p$ with p of norm less than M should be $2\sqrt{D}/(\pi h_D^2) \times M/(\ln M)^2 \times \beta(D) \times \gamma(M)$.

We bound $\beta(D)$ for D congruent to 3 modulo 8 by considering (unachievably) extremal splitting behavior of rational prime ideals (p). Were every odd prime to split as the product of two distinct primes to such a field, then $\beta_{\text{split}} = 2/9 \times C_2^2 = 0.3874\ldots$. If all odd primes were to remain inert, one finds $\beta_{\text{inert}} = 0.87299$.

We conclude that the number of trials of pairs (v,w) to find a prime pair (p,u) with p of norm in an interval $[S,2S]$ should be $\bar{N}_p \times \bar{N}_u$ with $\bar{N}_u$ approximately a constant times $h_D \ln S/\beta(D)\sqrt{D}$. Again, our data confirm this. See in particular Figure 8.

Figure 8: A comparison of theoretic and experimental values for $N_p \times N_u$.



6.3 Special Case: Class Number One

The *reduction* of an equation over the integers $\mathbb{Z}$ with respect to a prime number p is given by reducing each coefficient of the equation modulo p. This can be extended to equations of the rational numbers; and indeed to equations over algebraic number fields, where one reduces by prime ideals.

Koblitz [8] used the Hardy-Littlewood heuristics to derive conjectures on the number of primes p for which the reduction of an elliptic curve defined over $\mathbb{Q}$ is an elliptic curve of prime order. In the class number one CM setting this number should be asymptotic to a constant times $M/(\ln M)^2$; the constant is explicit.

In deriving his conjecture, Koblitz does not directly use twin primes in $\mathbb{Q}(\sqrt{-D})$. It would be very interesting to relate his constant to the Gross-Smith $\beta(D)$ in this restricted case of class number one. We briefly review why there might well be such a relationship.

An elliptic curve of j-value $j_0 \pmod p$ found with the CM method is the reduction of an elliptic curve defined over the complex numbers having j-value the corresponding root of the class polynomial $H_D(x)$. The reduction is with respect to a prime lying above p in the algebraic number field in which the root lies. In the class number one case, the single root of $H_D(x)$ is in $\mathbb{Z}$. The corresponding elliptic curve is defined over $\mathbb{Q}$, and the CM method amounts to reducing the equation of this curve modulo primes which split to principal ideals in $\mathbb{Q}(\sqrt{-D})$. Thus, Conjecture B of [8] then predicts the number of primes up to M (up to choosing twists) that give prime order elliptic curves.

Table 4 gives a comparison between the Koblitz predicted value, the Gross-Smith twin primes value, and actual counts of twin primes and of anomolous primes. The anomolous values are primes naturally paired with themselves in our construction. (These are not counted as acceptable values of u in our timing and counts for the various N_u.) Whereas the Gross-Smith formula should give the number of twins, the Koblitz formula reasonably interpreted should give the number of twins plus half the number of the anomolous curves.

7 Conclusion

We present a variant of the complex multiplication (CM) elliptic curve generation algorithm for $\mathbb{F}_p$. We show that the new variant of the CM method allows off-line precalculation and therefore provides smaller, faster and more easily coded software on-line implementation. The theoretical analysis shows that there are numerous prime numbers in this subset and experimental results confirm that it is highly probable to construct a prime number belonging to this set with a fairly small number of searches. Our experiments also reveal the fact that the on-line performance of the modified CM method increases as the class number decreases. Another interesting result is that the new CM method performs better for larger discriminants of the same class.

Table 4: Twin primes: estimates and counts.

D	M	Koblitz	Gross-Smith	Twins	Anomolous
11	2000	10.9	12.1	12	4
	4000	17.9	19.2	20	4
	6000	24.1	25.5	23	5
	8000	30.1	31.3	26	5
	10000	35.7	36.7	33	5
19	2000	24.2	25.9	23	5
	4000	37.9	41.1	36	7
	6000	51.2	54.5	51	7
	8000	63.1	66.9	63	7
	10000	75.2	78.6	78	9
43	2000	41.7	46.1	45	4
	4000	67.1	73.2	72	5
	6000	89.2	97.0	88	5
	8000	111.1	119.0	105	6
	10000	131.5	139.9	122	7
67	2000	54.8	59.2	56	4
	4000	88.2	93.9	91	6
	6000	117.2	124.5	125	7
	8000	144.8	152.7	157	7
	10000	172.4	179.4	189	8
163	2000	76.6	94.3	72	4
	4000	128.9	149.6	127	5
	6000	180.0	198.3	183	6
	8000	225.4	243.3	234	6
	10000	265.4	285.8	272	6

References

1. A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61(203):29–68, July 1993.
2. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer, Berlin, Germany, 1997.
3. H. Cohn. *Advanced Number Theory*. Dover Publications, New York, NY, 1980.
4. D. A. Cox. *Primes of the Form x^2+ny^2: Fermat, Class Field Theory and Complex Multiplication*. John Wiley & Sons, New York, NY, 1989.
5. R. Gross and J. H. Smith. A generalization of a conjecture of hardy and littlewood to algebraic number fields. *Rocky Mountain J. Math*, 30(1):195–215, 2000.
6. G. H. Hardy and J. E. Littlewood. Some problems of 'partitio numerorum' iii: On the expression of a number as a sum of primes. *Acta. MAth*, 44:1–70, 1922.
7. IEEE. P1363: Standard specifications for public-key cryptography. Draft Version 13, November 12, 1999.
8. N. Koblitz. Primality of the number of points on an elliptic curve over a finite field. *Pacific J. Math.*, 131(1):157–165, 1988.

9. N. Koblitz, A. Menezes, and S. Vanstone. The state of elliptic curve cryptography. towards a quarter-century of public key cryptography. *Designs, Codes and Cryptography*, 19(2-3):173–193, 2000.
10. G.-H. Lay and H. G. Zimmer. Constructing elliptic curves with given group order over large finite fields. *Algorithmic number theory (Ithaca, NY, 1994)*, pages 157–165, 1994.
11. A. K. Lenstra. Efficient identity based parameter selection for elliptic curve cryptosystems. *Information Security and Privacy—ACISP '99 (Wollongong)*, pages 294–302, 1999.
12. H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126(3):649–673, 1987.
13. A. Miyaji. Elliptic curves over F_p suitable for cryptosystems. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology – AUSCRYPT 92*, Lecture Notes in Computer Science, No. 718, pages 492–504. Springer, Berlin, Germany, 1992.
14. T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Commentarii Math. Univ. St. Pauli*, 47:81–92, 1998.
15. M. Scott. A C++ implementation of the complex cultiplication (CM) elliptic curve generation algorithm from Annex A. `http://grouper.ieee.org/groups/1363/P1363/implementations.html`, March 14, 2000.
16. I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation*, 67(221):353–356, January 1998.
17. V. Shoup. NTL: A Library for doing Number Theory (version 5.0c). http://shoup.net/ntl/, 2001.
18. J. H. Silverman. *The Arithmetic of Elliptic Curves*. Springer, Berlin, Germany, 1986.
19. N.P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptography*, 12:193–196, 1999.