Non Supersingular Elliptic Curves for Public Key Cryptosystems

T. Beth
F. Schaefer
Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe
Fasanengarten 5
D-7500 Karlruhe 1

Abstract

For public key cryptosystems multiplication on elliptic curves can be used instead of exponentiation in finite fields. One attack to such a system is: embedding the elliptic curve group into the multiplicative group of a finite field via weilpairing; calculating the discrete logarithm on the curve by solving the discrete logarithm in the finite field. This attack can be avoided by constructing curves so that every embedding in a multiplicative group of a finite field requires a field of very large size.

1 Introduction

In 1985 [10] Miller has suggested to use the chord tangent group law over elliptic curves for public key cryptosystems. These elliptic curve groups are used in a way similar to multiplicative groups of finite fields à la Diffie/Hellman (see [5][6]).

In this paper we discuss different possibilities to choose elliptic curves over finite fields with respect to application for such cryptosystems. The supersingular curves E with $\#E(GF(q)) = q + 1$ elements on the curve earlier proposed by Koblitz ([7]) are not well suited for that purpose. Cryptosystems based on such a type of curves can

D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp. 316-327, 1991.

be attacked by a new discrete logarithm algorithm recently presented by A. Menezes, T. Okamoto and S. Vanstone([9]).

This algorithm uses the Weil pairing for embedding the group of the curve into the multiplicative group of a finite field. By that the discrete logarithm problem on the curve is reduced to the discrete logarithm problem in the finite field. Menezes, Okamoto and Vanstone propose to use some other supersingular elliptic curves instead, because this class of elliptic curves provides some advantages with respect to implementation. However, these curves can still be embedded in finite fields of a somewhat larger size.

In this paper we show, that the crucial embedding can be hardened by using curves over $GF(p^n)$ with p prime and $p \gg 2$ or by using non supersingular curves over $GF(2^n)$. Then the breaking algorithm sketched above cannot feasibly be applied even if some progress in solving discrete logarithms is obtained. Due to the advantages provided by the use of fields with characteristic 2, especially for purposes of VLSI design, we concentrate in this paper on the arithmetic on non supersingular curves over these fields.

2 Mathematical Preliminaries

An elliptic curve $E(k)$ over a field k is defined to consist of all points $(x, y) \in k \times k$, which are solutions of a socalled Weierstrass equation:

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

where $a_1, ..., a_6 \in k$ are constants such that E has no singularities,

together with an additional point 0, the "point at infinity".

We have to regard these curves in projective instead of affine coordinates. Then one has to consider the homogeneous equation:

$$(*) \quad E_h : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2 + a_4X + a_6.$$

Points on this curve are described as equivalence classes of tripels (X, Y, Z) fulfilling the equation $(*)$, where the equivalence relation $(**)$ is defined:

$$(**) \quad (X, Y, Z) \cong (Xo, Yo, Zo) \text{ iff } \exists\, c \in k \text{ with } X = cXo, Y = cYo \text{ and } Z = cZo.$$

Then the point at infinity 0 can be represented by $(0,1,0)$, because this represents the only solution of the homogenous equation $(*)$ for $Z = 0$.

On elliptic curves an additive group operation can be defined in such a way, that the point at infinity becomes the zero element of this group. Using the affine coordinate representation of above, the group operation can be calculated as follows:

Let $P := (x_1, y_1)$ and $Q := (x_2, y_2)$ be two points on the curve $E(k)$ and $P + Q =: (x_3, y_3)$ be the sum. Then:

For $x_1 \neq x_2$ define:

$$\lambda := \frac{y_2 - y_1}{x_2 - x_1}, \quad \nu := \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}.$$

For $x_1 = x_2$, but $y_1 + y_2 + a_1 x_2 + a_3 \neq 0$ define:

$$\lambda := \frac{3x_1^2 + 2a_2 x_1 + a_4 - a_1 y_1}{2y_1 + a_1 x_1 + a_3},$$

$$\nu := \frac{-x_1^3 + a_4 x_1 + 2a_6 - a_3 y_1}{2y_1 + a_1 x_1 + a_3}.$$

Using this definition of λ and ν the line $y = \lambda x + \nu$ passes through P and Q, or is a tangent to E if $P = Q$.

Now $P + Q =: (x_3, y_3)$ is given by:

$$x_3 := \lambda^2 + a_1\lambda - a_2 - x_1 - x_2,$$
$$y_3 := -(\lambda + a_1)x_3 - \nu - a_3.$$

Thus the sum $P + Q$ is the third intersection point of the line $y = \lambda x + \nu$ with $E(k)$ reflected at the symmetry line of the curve E.

If $x_1 = x_2$ and $y_1 + y_2 + a_1 x_2 + a_3 = 0$, then:

$$P + Q = 0$$

From this addition on the curve a multiplication with a scalar $m \in \mathbf{N}$ can be defined:

$$m \in \mathbf{N}, \; m \cdot P := P + \ldots + P.$$

With elliptic curves one usually asserts certain quantities. The fact, that E should not have any singularities can be expressed in terms of the discriminant. Let c_4, c_6 be:

$$c_4 := (a_1^2 + 4a_2)^2 - 24(2a_4 + a_1 a_3)$$
$$c_6 := (a_1^2 + 4a_2)^3 + 36(a_1^2 + 4a_2)(2a_4 + a_1 a_3) - 216(a_3^2 + 4a_6).$$

Then for the discriminant Δ holds:

$$1728\ \Delta = c_4^3 - c_6^2\ .$$

For elliptic curves $\Delta \neq 0$ is supposed. This ensures that there are no singularities on the curve.

Closely related to the dicriminant is the j–invariant. It is defined by:

$$j := \frac{c_4^3}{\Delta}\ ,$$

and characterizes curves over algebraic closed fields up to isomorphism.

Another important tool for analysing elliptic curves is the endomorphism ring over the curve, where the curve is considered over the algebraic closure of the underlying field:

$$End(E) := End(E(\overline{k})).$$

The multiplication defined above gives a natural embedding of the ring of integers Z into $End(E)$. A curve is said to have *complex multiplication*, if $End(E)$ is stricly larger than Z. For curves over finite fields this is always the case.

An elliptic curve is called *supersingular*, iff the endomorphism ring $End(E)$ is non–commutative. The commutativity of $End(E)$ is only dependent on the structure of the curve over algebraic closures. Therefore it can be related to the j–invariant. For curves over fields with characteristic 2 the supersingular curves are exactly those with j–invariant $= 0$ (see [12] for proofs and further details).

3 Elliptic Curve Cryptosystem

A general concept to find one way functions is to construct a large finite cyclic group $(G, \circ)$ together with a generator $g \in G$, such that it is "easy" to calculate $m \cdot g := g \circ \ldots \circ g$ for all $m \in \mathbf{N}$ but "difficult" to retrieve the m for some arbitrary element $h \in G$, such that $m \cdot g = h$ holds. Here "easy" to calculate means solvable in polynomial time, where the polynom has a degree in the size of $\log |G|$, and "difficult" to calculate means has more than polynomial time complexity, at least for the best known algorithms.

One realization using this construction is given by the exponentiation in large finite fields proposed by Diffie and Hellman. This principle is also used here to construct a public key cryptosystem based on elliptic curves using the above defined group operation:

Choose an elliptic curve $E(GF(q))$ over a finite field $GF(q)$ together with a base point $P \in E(GF(q))$. This base point should be of high order. Then it is comparatively easy to calculate a scalar multiplication mP ($:= P + \ldots + P$ m–times), but difficult to calculate m given P and mP.

The later problem is called the discrete logarithm problem for elliptic curves. The best algorithm for descrete logarithms working for every type of elliptic curves is the GiantStepBabyStep algorithm. It has a complexity of $O(\sqrt{l_p})$, where l_p is the largest primefactor of the order of the base point P. This algorithm can be used to calculate discrete logarithms in any finite cyclic groups.

4 The Weil Pairing and the Discrete Logarithm

To motivate our further reasoning on how the curves should be choosen, a short review of the reduction algorithm due to A. Menezes, T. Okamoto and S. Vanstone is given([9]). The main idea is to reduce the calculation of a discrete logarithm in the elliptic curve group $E(k)$ to a discrete logarithm problem in a finite field GF by embedding the elliptic curve group $E(k)$ into the multiplicative group of a finite field GF. This embedding is delivered by the Weil pairing:

$$e_N : E[N] \times E[N] \rightarrow \mu_N,$$

which maps the n-torsion group over the algebraic closure of the field

$$E[N] := \{P \in E(\overline{k}) \mid N \cdot P = 0\}$$

into the set of N-th roots of unity (for details see [12]). Using the bilinearity and the non–degeneracity of the Weil pairing, such an embedding of an elliptic curve group into a finite field can be constructed.

By choosing the second component of the map in such a way, that the image of P under e_N is a primitive root, one gets a multiplication preserving function. Thus calculating a discrete logatithm in an elliptic curve group can be reduced to calculating a discrete logarithm in a finite field containing the N-th roots of unity.

Discrete logarithms in finite fields $GF(2^n)$ can be calculated by Coppersmith's algorithm([3]) with a complexity of

$$O(\exp(cn^{\frac{1}{3}}(\ln n)^{\frac{2}{3}}).$$

Thus the calculation of a discrete logarithm in an elliptic curve group is fastened up by this embedding as long as we can find the N-th roots of unity in a finite field of low extension degree over the basic field.

For the special case of the curve

$$E : y^2 * y = x^3$$

over fields with characteristic 2 the Weil pairing gives an embedding:

$$E(GF(2^n)) \hookrightarrow GF(2^{2n})^*$$

(see [9]). That means that in this case the embedding leads to much faster algorithm for the calculation of discrete logarithms.

In general this is only true, if the roots of unity are in an extension field with low degree over the field $GF(q)$. To avoid this kind of attack in [9] the following curves are proposed:

$$E_1 : \ y^2 * y = x^3 + x \ ,$$
$$E_2 : \ y^2 * y = x^3 + x + 1 \ .$$

For these two curves the Weil pairing deliveres an embedding of $E(GF(q))$ into $GF(q^4)^*$, i.e. a field with a representation of four times the bitlength. Then the time for computing discrete logarithms with the reduction algorithm rises properly.

	GiantStepBabyStep		Discr. Log in $GF(q^4)$
$E(GF(2^{100}))$	$\approx 10^{16}$	$GF(2^{4*100})$	$\approx 10^{23}$
$E(GF(2^{200}))$	$\approx 10^{31}$	$GF(2^{4*200})$	$\approx 10^{31}$
$E(GF(2^{300}))$	$\approx 10^{46}$	$GF(2^{4*300})$	$\approx 10^{37}$

Table 1: The complexity of discrete logarithm algorithms in $E(GF(q))$ by GiantStep-BabyStep method compared to the complexity of discrete logarithms in $GF(q^4)$ calculated by Coppersmith's method.

The figures in Table 1 shows, that the exponent of 4 is not satisfying for implementations with higher level of security and with respect to further progress in algorithm technique and machine speed. Already for curves over fields of size larger than 2^{200} it is easier to attack the cryptosystem by embedding and solving discrete logarithm than by using the GiantStepBabyStep method.

5 The Number of Points on Elliptic Curves

To avoid the attack described in the previous paragraph, we consider different types of curves. The following theorem gives the possible number of points on elliptic curves depending on the supersingularity. It is formulated in this way in Waterhouse' thesis, but its content is going back to Deurings work on elliptic function fields in the 1940os ([4]).

Theorem 1 *Let E be an elliptic curve over the finite field $GF(q)$ with $q = p^n$ a power of the prime p. Let the number of points on E be:*

$$\sharp E(GF(q)) = q + 1 - \beta.$$

In the case of supersingular curves one of the following condition holds:

(i) n *even:* $\beta = \pm 2\sqrt{q}$,

(ii) n *even and* $p \not\equiv 1 \pmod 3$: $\beta = \pm\sqrt{q}$,

(iii) n *odd and* $p = 2$ *or* 3: $\beta = \pm\sqrt{pq}$,

(iv) either n *odd or* n *even and* $p \not\equiv 1 \pmod 4$: $\beta = 0$.

In the case of non supersingular curves, β fulfills the following properties:

- $|\beta| \leq 2\sqrt{q}$ *(Hasse-bound) and*
- $gcd(\beta, p) = 1$.

The inverse statement is also valid in the sense that all the cases above occur.

(For proofs see [14] and [4]). For illustration of the theorem we give an example, which will be used later on:

Example 2 Over the finite field $GF(16)$ the possible sizes of elliptic curves are

$$\sharp E(GF(16)) = 16 + 1 - \beta \in \{9, 13, 17, 21, 25\}$$

for supersingular curves, and

$$\sharp E(GF(16)) = 16 + 1 - \beta \in \{10, 12, 14, 16, 18, 20, 22, 24\}$$

for non supersingular curves.

Note that in general, there are at most 5 supersingular elliptic curves over a finite field $GF(2^n)$ and $2 \times 2^{\lfloor \frac{n}{2} \rfloor}$ non–supersingular ones. As shown in [9] all elliptic curves over $GF(2^n)$ with $j(E) = 0$, i.e. supersingular curves, allow an embedding in a finite field of at most four times the bitlength. For the reasons explained above we look for curves which can not be embedded in extension fields of such small degree.

One possibility is to change the characteristic of the underlying field. For example in fields with characteristic 3 the supersingular curves with $q + 1 \pm \sqrt{q}$ points require fields with six times the bitlength for such embeddings. But due to faster and smaller implementations of arithmetic in fields of characteristic 2 these are of more interest. The non–supersingular curves give suitable candidates.

6 Construction of Suitable Non Supersingular Elliptic Curves

Over finite fields with characteristic 2 the non supersingular curves can be represented as:

$$E: \quad y^2 + xy = x^3 + a_2x^2 + a_6,$$

where the j–invariant $j(E) = \frac{1}{a_6}$. As shown in the previous paragraph we find curves with $\sharp E(GF(2^n)) = 2^n + 1 - \beta$ for any given β with β odd and $|\beta| \leq 2\sqrt{q}$ by choosing the coefficients a_2 and a_6 in $GF(2^n)$ and $a_6 \neq 0$.

An important point for the security of such a crypto system is to guarantee that there is a very large prime factor $p_{E(GF(2^n))}$ in the number of points calculated. Otherwise the GiantStepBabyStep algorithm can be applied succesfully to the subgroups. Thus the cyclic subgroup of order $p_{E(GF(2^n))}$ is the cryptographic essential part of the elliptic curve group.

To embed this cyclic subgroup into the multiplicative group of some extension field of $GF(2^n)$, let say $GF((2^n)^i)$, the following condition is necessary:

$$p_{E(GF(2^n))} | 2^{ni} - 1.$$

We are interested in curves where this *divisibility property* does not hold at least up to some extension degree k. This can be checked by calculating the following *gcdos*:

$$\gcd(p_{E(GF(2^n))}, 2^{ni} - 1) \quad (i = 1, \ldots, k).$$

or respectively:

$$\gcd(\sharp E(GF(2^n), 2^{ni} - 1) \quad (i = 1, \ldots, k).$$

How can we construct examples of curves fulfilling the divisibility property for k large enough and at the same time having a large prime factor in the group order. The algorithms to count the number of points on curves have rather high complexity. Therefore it is difficult to find curves for a given β in very large fields in general. If the number of points of a fixed curve over a finite field is known,then the number of points over any extension of this field can be calculated easily.

One proceeds as follows: We start with some small extension field of $GF(2)$, say $GF(q)$. Here it is easy to determine curves for all the possible number of points:

$$\sharp E(GF(q)) = q + 1 - \beta_0 \qquad \beta_0 \text{ odd and } |\beta| \leq 2\sqrt{q}.$$

Using the weil conjecture (see [12]) we find the number of points of this curves for any extension field of $GF(q)$:

$$\sharp E(GF(q^k)) = q^k + 1 - \beta_k,$$

where $\beta_k = a^k + b^k$ and a, b are the complex solutions of $1 - \beta_o T + qT^2 = (1 - aT)(1 - bT)$.

Obviously $E(GF(q))$ is a subgroup of $E(GF(q^k))$. Therefore $\sharp E(GF(q^k))$ has a small factor, namely $\sharp E(GF(q))$. It can be checked, whether the remaining factor

$$\frac{\sharp E(GF(q^k))}{\sharp E(GF(q))}$$

is prime. For $l \in \mathbf{N}$ with $l \nmid k$, $E(GF(q^l)) \subseteq E(GF(q^k))$ is a subgroup. Thus it is sufficient to consider only extensions of prime degree over $GF(q)$.

By this method it is possible to find curves over $GF(2^n)$ fulfilling the divisibility property for relatively large k with a large prime factor in the group order.

For illustration a special example suited for public key cryptosystems is constructed. In Example 2 we considered the number of elements on curves over $GF(16)$. The smallest non–supersingular elliptic curve group over this field has order 10. Computer search gives all curves with 10 elements over $GF(16)$. Representing the field as $GF(2)[x]/(x^4 + x + 1)$ and the curve as $y^2 + xy = x^3 + a_2x^2 + a_6$, the coefficient a_2 can be choosen as a polynomial of degree 3 and a_6 as one of the polynomials $x^2 + x$ or $x^2 + x + 1$.

Enlarging $GF(16)$ by a finite field extension of degree 47 we get a group of order

$$2 * 5 * 392318858461667547739736838942997715128064667934031507 29,$$

where the last factor is a probable prime with 56 digits. These curves fulfills the divisibility property for $k = 2 \ldots 100$. The factor 10 is due to the subgroup $E(GF(16))$.

7 Implementation

In this chapter some ideas are given how to implement the group operation on non–supersingular curves over fields of characteristic 2. In comparison to supersingular curves the addition here is slightly harder to compute, because doubling of a point is more complicated to calculate.

The complexity of the basic arithmetic operations in finite fields differs considerably. The additions are negligable in comparison to multiplications. The inversions are by fare the most time consuming operations. Therefore the curve is represented in projective coordinates. Then the inversions can be eliminated. Only at the end of each calculation two inversions are needed to get a unique representation (see [2],[8],[11]).

The homogeneous equation for non–supersingular curves over fields with characteristic 2 is:

$$Y^2Z + XYZ = X^3 + a_2X^2Z + a_6Z^3$$

with points $P = (X, Y, Z)$.

From a base point $P = (x, y, 1)$, we calculate $m*P$ with a double and add algorithm. Starting with the highest bit of m, we need only doublings of a point and additions of two different points of the form $(x, y, 1) + (x_i, y_i, z_i)$, i.e. we can assume that one of the points in the sums is given in affine coordinates. Then the following addition formulas are obtained:

Given $P_1 = (x_1, y_1, 1)$ and $P_2 = (x_2, y_2, z_2) \cong \left(\frac{x_2}{z_2}, \frac{y_2}{z_2}, 1\right)$,
and let $A := (z_2x_1 + x_2)$ and $B := (z_2y_1 + y_2)$, then

$$\begin{aligned} z_2A^2x_3 &= z_2B^2 + z_2AB + A^2(x_1z_2 + x_2 + a_2z_2) \\ z_2A^3y_3 &= z_2A^2(y_1x_2 + x_1y_2) + (A + B)(z_2A^2x_3), \end{aligned}$$

thus :

$$P_1 + P_2 = (z_2A^3x_3,\ z_2A^3y_3,\ z_2A^3) \cong (x_3, y_3, 1).$$

For the doubling of a point $P = (x, y, z)$, defining $A = (yz + x^2)$, we have:

$$\begin{aligned} x^2z^2x_d &= A^2 + xzA + a_2x^2z^2 \\ x^3z^3y_d &= x^5z + (A + xz)(x^2z^2x_3) \end{aligned}$$

$$\text{thus } 2P = ((x^3z^3)x_d,\ (x^3z^3)y_d,\ (x^3z^3)) \cong (x_d, y_d, 1).$$

The addition of above can be calculated with 12 multiplications and 1 squaring and the doubling of a point with 7 multiplications and 2 squarings.

Implementing the elliptic curve group operation in a VLSI design, the squarings can be calculated parallel with the multiplications, if the polynomial bases multiplier unit, invented by D. Gollman ([1]), is used. By using three multiplier units, these multiplications can be executed parallel. Thus the computing time is reduced to 3 repectively 4 multiplication steps for doubling and addition. Assuming that the factor m has a bit representation with half zeros and half ones, this means, that the average computation time would be 5 multiplication steps per bit. Additionaly there are around $4 \cdot \log n$ final multiplications for two inversions.

8 Conclusion

For public key cryptosystems based on problems like discrete logarithms, large groups are needed. The security depends on the structure of these groups.

Elliptic curves give the possibility to choose between a lot of different groups with different orders, especially if non supersingular curves are considered. This variety is the main advantage in comparison to the use of multiplicative groups of finite fields, where we have only one candidate for every field.

For algorithms, as the different index calculus methods, a large data base is calculated once for every candidate of group and out of this database single logarithms can be derived quickly. Also in this respect elliptic curves are a powerfull tool because of the richness of the many occuring cases.

References

[1] T. Beth, D. Gollmann; *Algorithm Engineering for Public Key Algorithms*; IEEE Journal on Selected Areas in Comm., Vol. 7, No. 4, 1989, pp. 458-466.

[2] T. Beth, W. Geiselmann, F. Schaefer; *Arithmetics on Elliptic Curves*; Algebraic and Combinatorical Coding Theory, 2nd int.workshop, Leningrad, 1990, pp. 28-33.

[3] D. Coppersmith; *Fast evaluation of logarithms in fields of characteristic two*; IEEE Trans. Inform. Theory, IT 30, 1984, pp. 587-594.

[4] M. Deuring; *Die Typen der Multiplikatorenringe elliptischer Funktionenkoerper*; Abh. Math. Sem. Hamburg, Bd. 14, 1941, pp. 197-272.

[5] W. Diffie, M. Hellman; *New directions in cryptography*; IEEE Trans. Inform. Theory, IT 22, 1976, pp. 644-654.

[6] T. ElGamal; *A public key cryptosystem and a signature scheme based on discrete logarithms*; IEEE Trans. Inform. Theory, IT 31, 1985, pp. 469-472.

[7] N. Koblitz; *Elliptic Curve Cryptosystems*; Mathematics of Computation, Vol. 48, No177, 1987, pp. 203-209.

[8] A. Menezes, S. A. Vanstone; *The Implementation fo Elliptic Curve Cryptosystems*; Advances in Cryptology-Auscrypt 90, Springer LNCS 453,1990, pp. 2-13.

[9] A. Menezes, T. Okamoto, S. A. Vanstone; *Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field*; University of Waterloo, preliminary version, sep. 1990.

[10] V. S. Miller; *Use of Elliptic Curves in Cryptography*; Advances in Cryptology: Proceedings of Crypto 85, Springer LNCS 218, 1986, pp. 417-426.

[11] P. Montgomery; *Speeding the Pollard and elliptic curve methods of factorization*; Math. Comp., Vol. 48, 1977, pp 243-264.

[12] J. H. Silverman; *The Arithmetic of Elliptic Curves*; Springer, New York, 1986.

[13] J. T. Tate; *The Arithmetic of Elliptic Curves*; Inventiones math. 23, Springer, 1974, pp. 179-206.

[14] W. C. Waterhouse, *Abelian Varieties over finite fields*; Ann. scient. Ec. Norm. Sup., 4th serie, 1969, pp. 521-560.