ELLIPTIC CURVES OVER SMALL FIELDS OF ODD CHARACTERISTIC

N.P. SMART

In recent years attention has focused on the use of elliptic curves in public key cryptography, starting with the work of Koblitz, [3], and Miller, [11]. This is because there is no known sub-exponential type algorithm to solve the discrete logarithm problem on an elliptic curve. The standard protocols in cryptography which make use of the discrete logarithm problem in finite fields, such as Diffie-Hellman key exchange, El Gamal and Massey-Omura, can all be made to work in the elliptic curve case.

However elliptic curves come with some disadvantages; for example addition on the curve is more expensive than multiplication in a finite field and determining a suitable curve to use is a rather cumbersome procedure. In this short note we propose using elliptic curves defined over small finite fields of odd characteristic. Such a proposal has been made many times before but over fields of even characteristic.

In the first section we show how one can perform a Frobenius expansion method to speed up the multiplication step over fields of odd characteristic. This procedure is almost identical to the procedure described in Müller, [12], for characteristic two, which is itself based on ideas in [4] and [9]. In the second section we shall describe how easy it is to determine suitable curves. Finally in the last section we shall discuss the advantages and disadvantages of using odd characteristic fields.

We shall assume that an elliptic curve is given by an equation of the form

$$E : Y^2 = X^3 + aX + b$$

where $a, b \in \mathbb{F}_q$, with $q = p^r$. To simplify our discussion we shall assume that $p \geq 5$. Our curve will be non-singular, so we shall assume that $4a^3 + 27b^2 \neq 0$. In addition, due to the results in [10], we shall assume that the curve is not supersingular. So in particular we have that p does not divide the trace of Frobenius, $t = q + 1 - |E(\mathbb{F}_q)|$. By Hasse's Theorem we know that $|t| \leq 2\sqrt{q}$, a fact which we shall be using throughout.

The q^{th}-power Frobenius endomorphism we shall denote by

$$\Phi : \begin{array}{ccc} E & \to & E \\ (x, y) & \to & (x^q, y^q). \end{array}$$

The map Φ satisfies the following equation

$$\Phi^2 - t\Phi + q = 0.$$

We shall be mainly interested in the group of points on E over some finite extension of $\mathbb{F}_q$, say $\mathbb{F}_{q^n}$.

1. Frobenius Expansions

In this section we shall show how to expand the multiplication by m map on $E(\mathbb{F}_{q^n})$ in terms of a polynomial in Φ. This allows us to replace the usual binary method of computing the multiplication by m map by a Frobenius expansion. The method is just a small generalization of the method in [12].

Lemma 1. *Let $S \in \mathbb{Z}[\Phi]$. Then there exists a unique integer, $R \in \{-(q-1)/2, \ldots, (q-1)/2\}$ and a unique element $Q \in \mathbb{Z}[\Phi]$ such that*

$$S = Q\Phi + R.$$

Proof. Easy. $\square$

Lemma 2. *Let $S \in \mathbb{Z}[\Phi]$ such that*

$$N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(S) \leq (\sqrt{q}+2)^2/4,$$

then we can write

$$S = \sum_{i=0}^{3} a_i \Phi^i$$

with $a_i \in \{-(q-1)/2, \ldots, (q-1)/2\}$.

Proof. Write $S = a + b\Phi$ with $a, b \in \mathbb{Z}$ then we have

$$N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(S) = a^2 + abt + b^q = \left(a + \frac{tb}{2}\right)^2 + \frac{1}{4}\left(4q - t^2\right) b^2.$$

As E is not supersingular and $|t| \leq 2\sqrt{q}$ we must have $b^2 \leq \left(4q - t^2\right) b^2 \leq (\sqrt{q}+2)^2$. So in particular $|b| \leq \sqrt{q} + 2$, and if $q \geq 11$ then $|b| \leq (q-1)/2$.

Suppose $|a| = (q-1)/2$ then

$$\frac{(q-1)^2}{4} + abt + b^2 q \leq (\sqrt{q}+2)^2/4,$$

which means $g_{q,t}(b) = 4qb^2 + 4atb + (q^2 - 3q - 3 - 4\sqrt{q}) \leq 0$. If this has any integral solutions for b then the discriminant of $g_{q,t}$, with respect to b, must be non-negative. So $16a^2t^2 - 16q(q^2 - 3q - 3 - 4\sqrt{q}) \geq 0$, but as $|a| = (q-1)/2$ and $|t| \leq 2\sqrt{q}$ we have $-16q(5q-4) + 64q\sqrt{q} \leq 0$. But this means $q \leq 1.904$ and so $|a| < (q-1)/2$.

Hence we have the upper bounds on a and b given by

$$|a| < (q-1)/2 \ , \ |b| \leq \sqrt{q} + 2.$$

If $q \geq 11$ then we have $|a|, |b| \leq (q-1)/2$ and we are done, so we can assume that $q = 5$ or 7. First, assume that $(q-1)/2 < b \leq \sqrt{q} + 2$, then we can write

$$S = a + b\Phi = a + (b-q)\Phi + q\Phi = a + (b-q)\Phi + t\Phi^2 - \Phi^3.$$

But then $0 > b - q > (q-1)/2 - q = -(q+1)/2$ and $|b - q| \leq (q-1)/2$. A similar argument holds when $-(\sqrt{q}+2) \leq b < -(q-1)/2$. So we are left with trying to show that $|t| \leq (q-1)/2$, in the above situation. As $N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(S) \leq (\sqrt{q}+2)^2/4$ we must also have

$$N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(t - \Phi) \leq (\sqrt{q}+2)^2/4.$$

If we assume that $(q+1)/2 \leq |t| \leq 2\sqrt{q}$ and $q = 5$ or 7 then we find that there are eight cases to check given by $q = 5$ (resp. $q = 7$) and $t = \pm 3, \pm 4$ (resp. $t = \pm 4, \pm 5$). None of these satisfy our inequality so we can deduce that $|t| \leq (q-1)/2$, as required. $\square$

That the required Frobenius Expansions exist and are not arbitrarily long follows from the following theorem.

Theorem 3. *Let $S \in \mathbb{Z}[\Phi]$ then we can write*

$$S = \sum_{i=0}^{k} r_i \Phi^i$$

where $r_i \in \{-(q-1)/2, \dots, (q-1)/2\}$ and $k \leq \lceil 2 \log_q 2\sqrt{N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(S)} \rceil + 3$.

Proof. From Lemma 1 we can obtain an expansion of the form

$$\begin{aligned} S &= S_0 = S_1\Phi + r_0 = (S_2\Phi + r_1)\Phi + r_0 \\ &= \sum_{i=0}^{j} r_i \Phi^i + S_{j+1}\Phi^{j+1} \end{aligned}$$

Using the triangle inequality we see, putting $||.|| = \sqrt{N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(.)}$,

$$\begin{aligned} ||S_{j+1}|| &\leq \frac{||S_j|| + ||r_i||}{||\Phi||} \leq \frac{||S_j|| + (q-1)/2}{\sqrt{q}} \\ &= \frac{||S_0||}{q^{(j+1)/2}} + \frac{(q-1)}{2} \sum_{i=1}^{j+1} q^{-i/2} \\ &= \frac{||S_0||}{q^{(j+1)/2}} + \frac{(q-1)}{2} \left(\frac{1 - q^{-(j+1)/2}}{\sqrt{q} - 1} \right) \\ &\leq \frac{||S_0||}{q^{(j+1)/2}} + \frac{\sqrt{q}+1}{2}. \end{aligned}$$

Now if $j \geq \lceil 2 \log_q 2||S_0|| \rceil - 1$ then

$$\frac{||S_0||}{q^{(j+1)/2}} \leq 1/2.$$

Hence

$$N_{\mathbb{Z}[\Phi]/\mathbb{Z}}(S_{j+1}) \leq \frac{(\sqrt{q}+2)^2}{4}$$

and so by Lemma 2 we know that S_{j+1} has a Frobenius expansion of length at most 4. □

We can then implement the multiplication by m map on the elliptic curve using Frobenius expansions. We first consider m as an element of $\mathbb{Z}[\Phi]$ and compute its Frobenius expansion,

$$m = \sum_{i=0}^{k} r_i \Phi^i,$$

where $k \leq \lceil 2 \log_q 2m \rceil + 3$. We can then compute mP for $P \in E(\mathbb{F}_{q^n})$ using Horner's method;

$$\begin{aligned} mP &= \sum_{i=0}^{k} r_i \Phi^i(P) \\ &= \Phi\left(\dots \Phi\left(r_k \Phi(P) + r_{k-1}P\right) + \dots + r_1 P\right) + r_0 P. \end{aligned}$$

Note at each stage of the expansion we add on an element of the form rP where $|r| \leq (q-1)/2$. To speed up this step we could precompute a table of such multiplications, this would be particularly useful if we wanted to perform many multiplications of the same point.

What Frobenius expansions have allowed us to do is replace many expensive elliptic curve doublings and additions with fewer elliptic curve additions and some power evaluations in a finite field. Just as in [12] one can also derive block versions of the Frobenius expansion method.

As an example, suppose we have an elliptic curve E defined over $\mathbb{F}_{23}$ with trace of Frobenius, $t = -1$. Let P denote a point on $E(\mathbb{F}_{23^n}$ and suppose we wish to compute $[m]P$ where $m = 10^6$. Using the standard binary method we would compute

$$[m]P = [2^6]\left(P + [2^3]\left(P + [2^5]\left(P + [2^2]\left(P + [2]\left(P + [2]\left(P + [2]P\right)\right)\right)\right)\right)\right).$$

So we require 6 elliptic curve additions and 19 elliptic curve doublings. The worst case situation for a six digit multiplier would require 19 additions and 19 doublings.

Now look at the Frobenius expansion of $[m]P$ for this curve,

$$\begin{aligned}[m]P &= \Phi(\Phi(\Phi(\Phi(\Phi(\Phi(\Phi(\Phi(-\Phi(P)+[2]P))+[7]P)-[3]P)-[9]P)-[5]P)\\ &\qquad\qquad\qquad\qquad\qquad -[4]P)-[8]P)+[6]P.\end{aligned}$$

Assuming we have a precomuted table of values of $[l]P$ for $l \in \{1,\ldots,11\}$ and noting that negation on an elliptic curve takes negligible time, we see that the Frobenius expansion method requires 9 elliptic curve additions, 9 table look ups, 9 applications of the Frobenius morphism and a single multiplication by a small integer. Each action of the Frobenius morphism requires only two powering operations in the field $\mathbb{F}_{23^n}$. The worst case situation of the Frobenius method for a six digit multiplier would require 12 elliptic curve additions and applications of the Frobenius morphism.

As in the case considered by Solinas, [14], we can reduce the length of the Frobenius expansion by nearly fifty percent. To show this we need to consider a small generalization of Euclidean domains:

Definition 1. *Let λ be a positive real number, let A denote a commutative ring and suppose that there is exists a multiplication function*

$$\Psi : A \setminus \{0\} \to \mathbb{N}.$$

The ring will be called λ-Euclidean if for all $a, b \in A$, with $b \neq 0$, we can find $q, r \in A$ with

$$a = bq + r$$

such that either $r = 0$ or $\Psi(r) < \lambda\Psi(b)$.

Such an idea is not new as one can see by looking at the survey article [8]. Suppose A has field of fractions K then we can extend Ψ to $K \setminus \{0\}$ in the obvious way. We then have

Lemma 4. *The ring A will be λ-Euclidean if for all $x \in K$ we can find a $y \in A$ such that*

$$\Psi(x - y) < \lambda.$$

Proof. Let $a, b \in A$ with $b \neq 0$ then set $x = a/b \in K$. So by the condition there exists a $y \in A$ such that $\Psi(a/b - y) < \lambda$. But as Ψ is multiplicative we see that, if $a \neq by$,

$$\Psi(a - by) < \lambda\Psi(b).$$

The result follows on setting $r = a - by$. □

The main result we shall use on λ-Euclidean rings is the following;

Theorem 5. *Suppose $\Phi^2 - t\Phi + q = 0$ then $\mathbb{Z}[\Phi]$ is λ-Euclidean for some λ such that* $0 < \lambda \leq (9+4q)/4$.

Proof. Let Ψ denote the standard norm function on $\mathbb{Z}[\Phi]$ and set $D = t^2 - 4q < 0$. We have two cases to consider:

Case 1 $t \equiv 0 \pmod 2$.

In this casse a basis of $\mathbb{Z}[\Phi]$ is given by $1, \sqrt{d}$, where $4d = D$. If $x = r + s\sqrt{d} \in \mathbb{Q}[\Phi]$ then we set $y = m + n\sqrt{d}$ with m being the nearest integer to r and n being the nearest integer to s, with some fixed convention for numbers of the form $(2i+1)/2$. Then, as $-d = q - t^2/4 \leq q$, we have,

$$\begin{aligned}\Psi(x-y) &= N_{\mathbb{Q}[\Phi]/\mathbb{Q}}(x-y) = (r-m)^2 - d(s-n)^2 \\ &\leq \frac{1-d}{4} \leq \frac{4+q}{4} \\ &\leq \frac{9+4q}{4}.\end{aligned}$$

Case 2 $t \equiv 1 \pmod 2$.

In this case a basis of $\mathbb{Z}[\Phi]$ is given by $1, (1+\sqrt{D})/2$. Let $x = r + s(1+\sqrt{D})/2 \in \mathbb{Q}[\Phi]$. As before we set m to be the nearest integer to r, n to be the nearest integer to s, and let $y = m + n(1+\sqrt{D})/2$. Then

$$\begin{aligned}\Psi(x-y) &= N_{\mathbb{Q}[\Phi]/\mathbb{Q}}(x-y) = ((r-m) + (n-s)/2)^2 + (n-s)^2 D/4 \\ &\leq (1\frac{1}{2})^2 - D/4 = \frac{1}{4}(9-D) \\ &\leq \frac{9+4q}{4}.\end{aligned}$$

□

We can now apply this result to reduce the length of our Frobenius expansion. Consider the integer, m, we wish to multiply P by as being an element of $\mathbb{Z}[\Phi]$. As $m \approx q^n$ then the norm of m will be equal to $m^2 \approx q^{2n}$. However we note that if we are considering points $P \in E(\mathbb{F}_{q^n})$ then we have the identity

$$\Phi^n P = P.$$

So we can "divide" m by $\Phi^n - 1$ to obtain a remainder r with

$$N_{\mathbb{Q}[\Phi]/\mathbb{Q}}(r) < \lambda N_{\mathbb{Q}[\Phi]/\mathbb{Q}}(\Phi^n - 1) \leq \frac{9+q}{4} N_{\mathbb{Q}[\Phi]/\mathbb{Q}}(\Phi^n - 1) \approx \frac{q^{n+1}}{4}.$$

Hence we can replace multiplication by m by multiplication by r. As r has norm roughly q^{n+1} its Frobenius expansion will be nearly half the length of the Frobenius expansion of m. This should provide a fifty percent improvement in the performance of our algorithm.

For completeness we give the formulae to compute r from m, t, n and q; First define $a_1, a_2 \in \mathbb{Z}$ by

$$a_1 + a_2\Phi = \Phi^n - 1 \pmod{\Phi^2 - t\Phi + q}.$$

Let $\Delta = a_1^2 + ta_1a_2 + qa_2^2$ and define $x_1, x_2 \in \mathbb{Q}$ by

$$x_1 = m(a_1 + ta_2)/\Delta \ , \ x_2 = -a_2 m/\Delta.$$

We can then let $r = r_1 + r_2\Phi$ with $r_1, r_2 \in \mathbb{Z}$ given by the following formulae

Case 1 $t \equiv 0 \pmod 2$. Let $h = [x_1 + \frac{x_2 t}{2}]$, where $[.]$ denotes the function which returns the nearest integer, then

$$r_1 = h - [x_2]t \ , \ r_2 = 2[x_2].$$

Case 2 $t \equiv 1 \pmod 2$. Now let $t = 2v + 1$ and set $h = [x_1 + x_2 v]$ then

$$r_1 = h - [x_2]v \ , \ r_2 = [x_2].$$

We end this section with some timings we have obtained. The Frobenius expansion method was implemented in software using two methods: The first used the LiDIA, [6], C++ library. To represent the finite fields we used the standard LiDIA data type **gf_base**. In particular this meant that we did not use a normal basis presentation, instead we used the more standard polynomial representation. The LiDIA library does not use special code for characteristics which fit into a single word, so some efficiency was lost in this implementation. The elliptic curve routines were implemented using an affine representation, so more divisions were carried out than would be strictly necessary.

Our second implementation used a normal basis representation of the field. This made the implementation of the Frobenius map rather simple as it then becomes a cyclic shift of the coefficients of the representation of the field elements. This field arithmetic was implemented in a way which allowed us to make use of the fact that the characteristic fitted into a single word. Again arithmetic on the curve was implemented in affine representation.

For comparison we also implemented the standard binary method of multiplication. As the binary method does not make use of exponentiation by q in the finite field we only implemented this using the polynomial representation of the field elements.

The timings in Table 1 do illustrate the speed up one achieves by using Frobenius expansions. All times are in hundreths of a second and are averages taken over a number of multiplications by random numbers of size the order of q^n. For larger base fields the time to perform the initial precomputation step will start to dominate the time to perform a multiplication, hence the average timings for the Frobenius expansion method do not include the time for the precomputation step, which is why we list this time separately.

For large degree finite fields we found the polynomial representation superior as multiplication of field elements in this representation is an $O(n^2)$ algorithm, for multiplication in a normal basis representation we used a $O(n^3)$ algorithm.

2. Finding Suitable Curves

There are three standard techniques that one uses to determine elliptic curves which are suitable for use in cryptography. The problem is that we need to determine non-supersingular curves which have a large cyclic subgroup. In particular this cyclic subgroup should have order larger than 10^{40}.

If we choose a field of definition of the elliptic curve of order around 10^{40} then we encounter two problems. Firstly we cannot use a Frobenius expansion method to perform multiplications and secondly determining the group order of a given curve is in general complicated as we need to apply Schoof's algorithm, see [13], [2] and [5]. We can choose curves at random, compute their orders and then factor their orders to see if we can find one with a large prime factor but this is a lot of work, especially if we wish to produce suitable elliptic curves "on line". For example,

TABLE 1. Comparison of the Binary and Frobenius methods

				Normal Basis		Polynomial Basis	
				Frobenius	Table	Frobenius	Table
p	n	$\log_2 \lvert E(\mathbb{F}_{p^n})\rvert$	Binary	Expansion	Creation	Expansion	Creation
5	32	74	447	167	10	186	5
5	53	123	1040	1212	44	398	6
5	75	174	2567	4771	125	975	11
11	22	76	249	48	10	92	10
11	36	124	799	318	42	278	18
11	51	176	1542	1241	119	497	25
23	17	76	178	20	12	59	19
23	28	126	625	135	48	214	36
23	39	176	1216	458	126	372	50
41	14	75	121	11	13	35	24
41	23	123	456	63	51	129	53
41	33	177	1232	254	145	354	96
127	11	76	87	4	23	26	53
127	18	125	338	26	84	97	121
127	25	175	750	89	211	227	196

Lercier, [5], gives a time of 86 seconds to compute the order of group of points on an elliptic curve defined over $\mathbb{F}_{2^{155}}$. Whilst his program takes 235 minutes to determine 5 suitable curves for cryptographic purposes over $\mathbb{F}_{2^{155}}$.

Another way of proceeding is to decide on a prime base field of large order and then using the theory of curves with complex multiplication to produce curves with a cyclic subgroup of large prime order, [7]. Again this is possible but it involves extracting roots of large degree polynomials over large finite fields so this may be far too slow for the generation of suitable curves.

Although Schoof's algorithm and root extraction in finite fields both run in polynomial time they are non-trivial algorithms which require careful coding. It is not surprising that although they are asymptotically efficient they are not particular quick at finding suitable curves than more naive methods.

One way around these problems, which has often been proposed, is to use elliptic curves defined over very small finite fields of characteristic two, [9], [12] and [14]. The reason is that it is easy to compute the number of points over a small field, $\mathbb{F}_q$, and computing the number of points over extension fields is then simple due to the following result.

Theorem 6. *Let $\mathbb{F}_q$ denote any finite field and let E denote an elliptic curve defined over $\mathbb{F}_q$. Write $E(\mathbb{F}_q) = q + 1 - c_1$ and $E(\mathbb{F}_{q^n}) = q^n + 1 - c_n$ then*

$$c_n = c_1 c_{n-1} - q c_{n-2}$$

where $c_0 = 2$.

The trouble with restricting attention to characteristic two is that there are not many curves defined over small finite fields with the required subgroup of large prime order, unless one uses a very large extension field. Table 2 demonstrates this by listing those values of t, q and n with $q = 2^r \leq 32$ and $q^n \leq 2^{200}$ which give rise to curves defined over $\mathbb{F}_q$ with group orders over $\mathbb{F}_{q^n}$ divisible by a prime, l, with more than forty decimal digits in it.

TABLE 2. Curves over even prime fields with a large prime divisor in the group order over an extension field

r	n	t	$\log_{10} l$	r	n	t	$\log_{10} l$	r	n	t	$\log_{10} l$
1	163	1	49	1	181	1	44				
2	79	1	47	2	97	1	53				
3	47	-1	41	3	59	-1	52	3	47	3	41
3	59	3	52								
4	37	-5	43	4	37	-3	43	4	37	-1	43
4	41	-1	44	4	47	-1	49	4	37	3	43
4	47	3	45	4	47	7	55				
5	31	-9	40	5	37	-5	49	5	31	-3	45
5	29	9	40								

However if we allow finite fields of characteristic larger than three then we can find a lot more suitable curves. Table 3 lists the curves we found over prime fields, $\mathbb{F}_q$, of order less than twenty four. We looked at extensions, $\mathbb{F}_{q^n}$, of these prime fields with $q^n \leq 2^{200}$. We tried to find curves with a prime, l, dividing the group order of at least forty decimal digits. To reduce the number of curves with a "smooth" group order we restricted our attention to prime values of n. In addition for very large group orders we did not try to produce a complete factorization so some suitable curves we may have missed. The whole computation took only 7 minutes or around 5 seconds per suitable curve found.

One could extend this table further. We found 11 examples of suitable curves defined over $\mathbb{F}_{41}$ with groups of points defined over $\mathbb{F}_{41^n}$ with $n \leq 37$, and 8 examples of suitable curves defined over $\mathbb{F}_{127}$ with groups of points defined over $\mathbb{F}_{127^n}$ with $n \leq 23$.

3. Advantages and Disadvantages

Using elliptic curves defined over small finite fields but with the group of points defined over a prime extension allows us to easily compute the order of the group of points. This in turn allows us to determine suitable elliptic curves for cryptographic purposes in a fast and efficient manner. In addition for such curves we can replace the standard binary multiplication method with a Frobenius expansion method. If we are computing a multiple of a fixed point then the use of look-up tables will greatly speed up the multiplication step in such a system.

If the base field has characteristic two then we can use efficient algorithms for the field arithmetic which are more suitable for a practical system. Our very rough timings seem to indicate, compared to [12], that using fields of odd characteristic is between ten and one hundred times slower than using even characteristic base fields of the same order. On the other hand with small fields of odd characteristic we have far more suitable elliptic curves at our disposal.

So why do some practical systems make use of elliptic curves defined over $\mathbb{F}_p$, where p is a large prime, or $\mathbb{F}_{2^n}$? This is because it is not beyond the realms of possibility that the extra structure obtained in having the curve defined over a small

TABLE 3. Curves over odd prime fields with a large prime divisor in the group order over an extension field

q	n	t	$\log_{10} l$	q	n	t	$\log_{10} l$	q	n	t	$\log_{10} l$
5	61	-4	41	5	83	-2	45	5	79	-1	50
5	83	-1	52	5	71	1	44	5	73	1	50
5	79	1	54	5	73	4	45				
7	53	-4	43	7	59	-4	49	7	61	-2	46
7	53	-1	43	7	59	-1	41	7	59	2	49
7	67	3	52	7	47	4	39	7	61	5	51
7	67	5	51								
11	53	-5	54	11	47	-3	42	11	41	-2	41
11	47	-2	44	11	53	-2	54	11	53	-1	50
11	47	2	43	11	53	4	50	11	53	6	50
13	47	-6	47	13	53	-6	51	13	37	-5	40
13	47	-5	51	13	47	-4	47	13	47	-3	48
13	53	1	54	13	41	2	40	13	47	4	45
13	43	5	46	13	41	6	40				
17	43	-8	51	17	37	-7	44	17	41	-7	46
17	43	-6	51	17	47	-6	54	17	37	-4	44
17	43	-3	51	17	43	-2	48	17	47	-1	53
17	41	2	41	17	37	3	41	17	47	4	52
17	47	5	51	17	43	6	42	17	47	7	56
19	37	-8	46	19	37	-7	43	19	41	-7	44
19	41	-5	42	19	37	-4	46	19	37	-3	42
19	43	-1	49	19	37	2	42	19	41	2	49
19	37	4	41	19	37	5	43	19	41	7	40
19	37	8	46	19	41	8	51				
23	37	-7	45	23	41	-7	51	23	37	-6	45
23	43	-5	49	23	31	-4	40	23	37	-4	42
23	41	-4	51	23	31	-3	40	23	43	-3	57
23	41	-1	50	23	43	1	57	23	37	3	49
23	43	3	52	23	37	5	46	23	31	8	41
23	41	8	48	23	43	8	57				

finite field will render the system less secure. After all the group $E(\mathbb{F}_{q^n})$ contains a subgroup $E(\mathbb{F}_q)$ which is stable under the action of the Frobenius morphism. However nobody has yet used such a structure to show that the proposed curves are any weaker than general curves over $\mathbb{F}_p$ or $\mathbb{F}_{2^n}$.

Our timings of the multiplication routines show that some research still needs to be done as to what size the base field should be. A larger value for the base field implies that the degree of the field extension needed can be smaller. This in turn means that arithmetic will be much faster. A large value for the size of the base field also means that the length of the Frobenius expansion will be short. However the larger the base field then the greater the size of the look up table required to perform the multiplication step.

References

[1] L.M. Adleman and M-D. Huang, editors. *ANTS-1 : Algorithmic Number Theory.* Springer-Verlag, LNCS 877, 1994.

[2] J.-M. Couveignes and F. Morain. Schoof's algorithm and isogeny cycles. In [1], pages 43–58.

[3] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, Vol 48, 203–209, 1987.

[4] N. Koblitz. CM-curves with good cryptographic properties. In *Advances in Cryptology, CRYPTO 91*, pages 279–287. Springer-Verlag, LNCS 576, 1992.

[5] R. Lercier. Finding good random elliptic curves for cryptosystems defined over $\mathbb{F}_{2^n}$. In *Advances in Cryptology, EUROCRYPT 97*, pages 379–392. Springer Verlag, LNCS 1233, 1987.

[6] LiDIA Group LiDIA v1.3 - A library for computational number theory. TH-Darmstadt, (1997).

[7] G.-J. Lay and H.G. Zimmer Constructing elliptic curves with given group order over large finite fields. In [1], pages 250–263.

[8] F. Lemmermeyer The Euclidean algorithm in algebraic number fields. *Expo. Math.*, Vol 13, 385–416, 1995.

[9] W. Meier and O. Staffelbach. Efficient multiplication on certain non-supersingular elliptic curves. In *Advances in Cryptology, CRYPTO 92*, pages 333–344. Springer Verlag, LNCS 740, 1992.

[10] A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Transactions on Information Theory*, Vol 39, 1639–1646, 1993.

[11] V. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology, CRYPTO 85*, pages 417–426. Springer Verlag, LNCS 218, 1986.

[12] V. Müller. Fast multiplication on elliptic curves over small fields of characteristic two. *To Appear: J. Crypto*, 1997.

[13] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, Vol 44, 483–494, 1985.

[14] J.A. Solinas. An improved algorithm for arithmetic on a family of elliptic curves. In *Advances in Cryptology, CRYPTO 97*, pages 357-371. Springer Verlag, LNCS 1294, 1997.

Hewlett-Packard Laboratories,Filton Road, Stoke Gifford, Bristol BS12 6QZ, U.K.
E-mail address: `nsma@hplb.hpl.hp.com`