FINDING CRYPTOGRAPHICALLY STRONG ELLIPTIC CURVES: A TECHNICAL REPORT

HAMISH IVEY-LAW AND ROBERT ROLLAND

Abstract. Elliptic curve cryptography is becoming the standard for public key cryptography. Unfortunately, there are no databases for elliptic curves with large parameters. In this report we first outline recommended parameters for use with elliptic curve cryptosystems. In particular we give the size of the base field, the requirements on the group of rational points, and the sizes of the different coefficients of the curves. We then describe a practical means for constructing a database of curves satisfying the given parameters and our implementation of such a system. Our database includes curves of size 256, 384, and 512 bits in both short Weierstrass form and in Edwards form.

1. Introduction

In a typical scenario, encrypting a large amount of data is done using a symmetric cryptosystem, for example using the AES block cipher with a particular mode of operation, while the exchange of the secret key used in the symmetric encryption is achieved using a public key cryptosystem. The combination of these two cryptosystems is often called a hybrid cryptosystem. The building blocks of most hybrid cryptosystems include

- ◇ a hash function;
- ◇ a random number generator used once to initialize a pseudo-random generator (for example `/dev/random` on Unix-like systems);
- ◇ a pseudo-random number generator used to draw initialization vectors, and keys; it is often constructed by repeatedly applying a hash function to a truly random seed, in which case it is often known as a key derivation function or mask generator;
- ◇ a symmetric cipher which operates on fixed length blocks of data, used via a mode of operation such as CBC, CTR or GCM;
- ◇ a keyed hash function, also known as a message authentication code (MAC);
- ◇ a key exchange system based on elliptic curve public key cryptography; and
- ◇ a signature protocol based on elliptic curve public key cryptography.

In 2005, the National Security Agency released its "Suite B" recommendations for cryptographic primitives [NSA09]. These recommendations include the use of

- ◇ the AES block cipher with key sizes of 128 and 256 bits;
- ◇ the Galois/Counter Mode (GCM) mode of operation when encrypting streaming data;
- ◇ the Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signatures;

Date: Febrary, 7, 2010.

Key words and phrases. elliptic curve cryptography, secure curve database, Edwards curves.

- ◇ the Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithm; and
- ◇ the SHA-256 and SHA-384 hash functions.

In particular, the NSA provides the following recommendations on key sizes (*op. cit.*):

> AES with 128-bit keys provides adequate protection for classified information up to the SECRET level. Similarly, ECDH and ECDSA using the 256-bit prime modulus elliptic curve as specified in FIPS PUB 186-3 ([NIS09]) and SHA-256 provide adequate protection for classified information up to the SECRET level. During the transition to the use of elliptic curve cryptography in ECDH and ECDSA, DH, DSA and RSA can be used with a 2048-bit modulus to protect classified information up to the SECRET level.
>
> AES with 256-bit keys, Elliptic Curve Public Key Cryptography using the 384-bit prime modulus elliptic curve as specified in FIPS PUB 186-3 and SHA-384 are required to protect classified information at the TOP SECRET level. Since some products approved to protect classified information up to the TOP SECRET level will only contain algorithms with these parameters, algorithm interoperability between various products can only be guaranteed by having these parameters as options.

These recommendations have been widely implemented and deployed.

So public key cryptography based on classical arithmetic problems (RSA, ElGamal, DSA) is no longer recommended. The development of public key cryptography seems to be focused now on elliptic curves. This is not surprising when we consider the progress on the factorization problem and on the discrete logarithm problem over $\mathbb{Z}/p\mathbb{Z}$.

Unfortunately, NIST gives only one elliptic curve over a prime finite field for each size. More precisely, NIST defines (cf. FIPS 186-3 [NIS09]) the curves P-256, P-384 and P-521 (respectively of size 256, 384 and 521 bits). On the other hand, there is no recommended curve in Edwards form. It is this situation that has motivated us to create databases containing

- ◇ many elliptic curves over $\mathbb{F}_p$ (for various p) in short Weierstrass form where p is 256, 384 or 512 bits long and with a prime number of $\mathbb{F}_p$-rational points; and
- ◇ many elliptic curves over $\mathbb{F}_p$ in Edwards form with where p is 256, 384 or 512 bits long and with a number of $\mathbb{F}_p$-rational which is $n = 4u$ where u is a prime.

2. Elliptic curves

From now on we suppose that k is a prime finite field $\mathbb{F}_p$ where p is a prime. We denote by $\overline{k}$ an algebraic closure of k. Recall that an elliptic curve over $\overline{k}$ is an algebraic curve of genus one. In the following we are interested in elliptic curves over $\overline{k}$ that are in fact defined over k.

We will denote by $\mathcal{C}$ the curve over $\overline{k}$ and by $\mathcal{C}(k)$ its set of k-rational points, that is the points on $\mathcal{C}$ whose coordinates lie in k. An elliptic curve together with a fixed k-rational point O on it has a group structure where O is the neutral element. Then we can define an elliptic curve as a pair $(\mathcal{C}, O)$ where $\mathcal{C}$ is a smooth curve of genus one and O a point on the curve. Note that if $\mathcal{C}$ is defined over k, and if O is

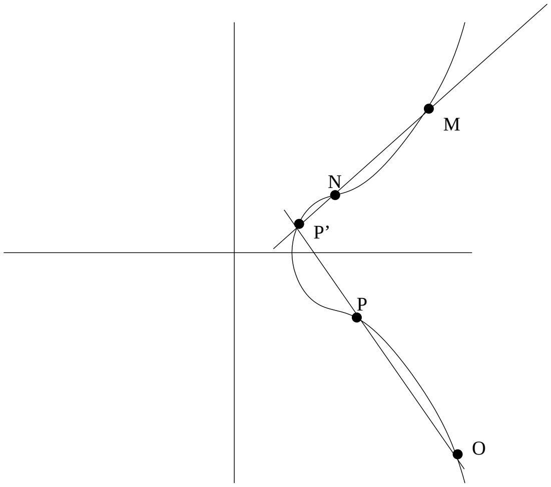

FIGURE 1. Addition $M + N = P$, neutral point O

a k-rational point on $\mathcal{C}(k)$, then $\big(\mathcal{C}(k), O\big)$ has a group structure whose operations are defined by rational functions. These functions describe the original chord and tangent method for adding points on an elliptic curve, as shown in Figure 2.

If we insist that the curve's model be smooth, any elliptic curve can be expressed as a cubic equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where the discriminant

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6,$$

with

$$d_2 = a_1^2 + 4a_2 \qquad d_4 = 2a_4 + a_1a_3 \qquad d_6 = a_3^2 + 4a_6$$
$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

is non-zero (this is equivalent to the condition that the curve be smooth). Equation 1 is known as the Weierstrass form of the curve. When the characteristic of the base field is not 2 or 3, we can reduce the equation to the short Weierstrass form:

$$y^2 = x^3 + a_4x + a_6, \tag{2}$$

where the discriminant is

$$\Delta = -16\left(4a_4^3 + 27a_6^2\right).$$

In what follows we will always consider curves over a prime field $\mathbb{F}_p$ with p a large prime, thus we can always work with such a short Weierstrass equation. However it turns out that recent work by Harold Edwards shows that a certain quartic model with two ordinary double points (giving curves of genus one but not all the curves of genus one) produces a particularly compact form for the addition of points of the curve which can lead to dramatic improvements in the speed of arithmetic calculations. In this report we will consider both elliptic curves in short Weierstrass form and elliptic curves in Edwards form or twisted Edwards form.

Other interesting models for elliptic curves exist, notably the Montgomery form, however we will not consider those other forms here.

3. Curves in short Weierstrass form

3.1. **Global processing.** As mentioned above, when the characteristic of the base field is not 2 or 3, every elliptic curve can represented in short Weierstrass form (Equation 2). The group G of rational points of an elliptic curve over a prime finite field $\mathbb{F}_p$ is the group used for elliptic curve public key cryptography. We choose curves for which the cardinality of G is a prime number. We also choose at random a point $g = (g_x, g_y)$ on the curve generating the prime order cyclic group G. When finding g it is necessary to calculate square roots in the base field. For this reason we will assume the prime p satisfies $p \equiv 3 \pmod 4$, since for such p it is easier to extract a square root in $\mathbb{F}_p$.

In order to prove that the coefficients of the curve are drawn at random and thus that the curve is not of a special form, the values a_4 and a_6 are respectively the hashes of two random numbers r_4 and r_6, and g_x is the hash of a random value r. The procedure for selecting the curve parameters is as follows:

(1) Draw at random a prime p such that $p = 4k + 3$. The size of p may be 256, 384 or 512 bits.
(2) Compute the following:
  (a) Draw at random two numbers r_4 and r_6 and compute $a_4 = \text{Hash}(r_4) \bmod p$, $a_6 = \text{Hash}(r_6) \bmod p$.
  (b) Compute the cardinality n of the group G of the curve $y^2 = x^3 + a_4x + a_6$ using the SEA algorithm.
  until n is prime.
(3) Verify that we do not have some borderline case, for which the curve is bad.
(4) Compute the following:
  (a) Draw at random a number r and compute $g_x = \text{Hash}(r) \bmod p$.
  (b) Compute $z = g_x^3 + a_4 g_x + a_6$.
  until z is a square modulo p.
(5) Finally compute a square root g_y of z by the formula $g_y = z^{\frac{p+1}{4}} \bmod p$. This formula holds because of our choice of p in step 1.

Then the curve $y^2 = x^3 + a_4x + a_6$ is defined over $\mathbb{F}_p$, its group G of $\mathbb{F}_p$-rational points has prime order n, and G is generated by the point $g = (g_x, g_y)$.

We use a version of SEA (algorithm for calculating the number of points on an elliptic curve over a finite field due to R. Schoof, N. Elkies, A. Aitkin) included in the development version of PARI-GP, written in C by Bill Allombert, based on the GP implementation by Christophe Doche and Sylvain Duquesne. This version was amended by the first author in the following two ways:

- ⋄ The database of modular polynomials was expanded in order to compute on curves over 512-bit base fields; these polynomials where originally computed by David Kohel.
- ⋄ A patch to the PARI-GP system was applied to disable the early abort of the SEA calculation when a divisor of 2 was found in the order of G. This was necessary to find elliptic curves in Edwards form, since the number of rational points on such curves is always divisible by 4.

Let us describe the use of the hash function above in more detail. The coefficient a_4 (respectively a_6) of the curve is obtained from the random number r_4 (respectively r_6) by calculating

$$a_4 = \mathrm{hextodec}(\mathrm{sha512.hex}(\mathrm{dectostr}(r_4))) \mod p.$$

More precisely, we write r_4 in decimal and we hash the octet stream given by this decimal number with the SHA-512 function. The digest produces a hexadecimal string which is then translated back to decimal, giving the result a_4.

3.2. **Details.** We wrote two PARI scripts to generate the database of elliptic curves in Weierstrass form. The first generates Weierstrass curves defined over a prime finite field $\mathbb{F}_p$ with a prime number of $\mathbb{F}_p$-rational points. For each candidate curve, we also perform some tests to ensure that it is not of a kind known to be insecure. Specifically we check that

(1) the number n of rational points is not equal to p (*i.e.* curves with trace equal to 1) to avoid the anomalous case for which the attack of Semaev-Smart-Satoh-Araki applies; and

(2) $p^d \not\equiv 1 \pmod{n}$ for $d = 1, \ldots, 30$, to avoid the pairing attack (MOV attack) of Weil-Tate (Menezes-Okamoto-Vanstone and Frey-Rück)—if this condition is satisfied, the embedding degree is clearly larger than 30, so in particular we avoid supersingular curves (where the trace is zero) and curves with trace equal to 2.

The second script verifies whether a given curve was correctly generated. The first script generates files describing curves in the following form

```
p = 88849331028320216703108566011123832795
    07496491807071433260928721853918699951
n = 88849331028320216703108566011123832794
    54437918059397120004264665392731659049
a4= 24815133168353065184960919504888673668
    05208929993787063131352719741796616329
a6= 43873059585863478905292603208312861397
    99795892409507048422786783411496715073
r4= 54739537861363309295053728858641261239
    58065998198197694258492204115618878079
r6= 58312739525090925557761162256886910725
    12584265972424782073602066621365105518
gx= 76381663548487413330901760682863114793
    65713946232310129943505521094105356372
gy= 76268736705197597776108991270168627406
    0655281117983501949286086861823169994
r = 80944585957702065420031500895142393857
    61983350496862878239630488323200271273
```

(This is a real example of a curve over a 256-bit prime field.) Given such a file the test script verifies that

- ◇ p is a prime;
- ◇ $p \equiv 3 \pmod{4}$;
- ◇ n is prime;
- ◇ $a_4 \equiv \mathrm{Hash}(r_4) \pmod{p}$;

- ◇ $a_6 \equiv \text{Hash}(r_6) \pmod p$;
- ◇ $g_x \equiv \text{Hash}(r) \pmod p$;
- ◇ n is the number of rational points of the curve;
- ◇ the curve is not of an insecure type;
- ◇ the point $g = (g_x, g_y)$ is on the curve;

Remark 3.1. The SEA algorithm is a rather long computation, especially for parameters in the order of 384 or 512 bits. Thanks to the early-abort capability due to R. Lercier, the algorithm can be prematurely stopped when a small factor is detected and restarted with a new curve.

3.3. **Database of elliptic curves in Weierstrass form.** Unfortunately, NIST gives only one curve for each recommended size of the base (*i.e.* in the range of 256, 384 or 512 bits). Using the program described in the previous section, we have computed a database of elliptic curves which can be found on the website

`http://galg.acrypta.com`

In the folders "cw*xxx*", where *xxx*=256, 384 or 512, one can find the files "w*xxx*-001.gp", . . . , "w*xxx*-018.gp". These files describe elliptic curves with *xxx* bits in the format given in the previous section.

4. Edwards curves an twisted Edwards curves

4.1. **Introduction.** The background material on Edward curves can be found on the website of Daniel Bernstein and Tanja Lange:

`http://cr.yp.to/newelliptic/newelliptic.html`

Many practical improvements have been made to the original work of Harold Edwards: the original Edwards form has been simplified, the efficiency of the addition algorithm has been improved, the number of elliptic curves that we can model in Edwards form has been increased, and so on. We refer to the previous website for a complete survey of the domain, for the technical details, and for pointers to the main papers. We just recall here some classification results (see for example [BBJ+08]) from which we have derived our choices for our cryptographic toolbox.

4.2. **Global processing.** Recall that our base field is the prime finite field $\mathbb{F}_p$ where p is a large prime (*i.e.* having a size in the order of 256, 384 or 512 bits). In the original work of Edwards, the studied curves are

$$x^2 + y^2 = c^2(1 + x^2y^2).$$

Bernstein and Lange showed that it is possible to obtain more curves by taking the following equations

$$x^2 + y^2 = c^2(1 + dx^2y^2),$$

and that in fact these curves are isomorphic to curves of the form

$$x^2 + y^2 = 1 + dx^2y^2. \tag{3}$$

From now on, curves in the form of Equation 3 will be named *Edwards curves*. The group of $\mathbb{F}_p$-rational points on such a curve always has a point of order 4, and so the order of an Edward curve's group is always a multiple of 4. In particular, it is never prime, unlike the case of Weierstrass curves. A *twisted Edwards curve* is a curve of the form

$$E_{E,a,d}: \quad ax^2 + y^2 = 1 + dx^2y^2.$$

Even more elliptic curves admit a model as a twisted Edwards curve.

Let us recall that a Montgomery curve is an elliptic curve of the form

$$E_{M,a,b}: \quad by^2 = x^3 + ax^2 + x$$

where $b \neq 0$. The following known results give us a curve classification.

Theorem 4.1. *Let k be a non-binary field (i.e. a field whose characteristic is not 2). Then every twisted Edwards curve over k is birationally equivalent over k to a Montgomery curve. More precisely, given a twisted Edward curve $E_{E,a,d}$, it is birationally equivalent to the Montgomery curve $E_{M,A,B}$ where*

$$A = \frac{2(a+d)}{a-d} \quad \textit{and} \quad B = \frac{4}{a-d},$$

and conversely given a Montgomery curve $E_{M,A,B}$, it is birationally equivalent to the Edwards curve $E_{E,a,d}$ where

$$a = \frac{A+2}{B} \quad \textit{and} \quad d = \frac{A-2}{B}.$$

Theorem 4.2. *Let k be a non-binary field. Let E be an elliptic curve defined over k. The group $E(k)$ of the k-rational points of E has an element of order 4 if and only if E is birationally equivalent over k to an Edwards curve.*

Theorem 4.3. *Let k be a finite field such that $\#k \equiv 3 \pmod 4$. Then every Montgomery curve over k (and hence every twisted Edwards curve over k) is birationally equivalent over k to an Edwards curve.*

For our database we made the following choices of parameters:

- ◇ p a large prime (256, 384 or 512 bits);
- ◇ $p \equiv 3 \pmod 4$;
- ◇ equation $x^2 + y^2 = 1 + dx^2y^2$ (Edwards curve);
- ◇ $d \neq 0$ and $d \neq 1$ so that the curve is irreducible of genus one;
- ◇ d is not a square in $\mathbb{F}_p$; and
- ◇ the number n of $\mathbb{F}_p$-rational points is four times a prime.

The second last condition is required to obtain a complete set of addition laws on the curve, namely an addition formula with no exceptional points. In this case, the equations for addition take the particularly elegant form

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3),$$

where

$$x_3 = \frac{(x_1y_2 + x_2y_1)}{(1 + dx_1x_2y_1y_2)} \quad \text{and} \quad y_3 = \frac{(y_1y_2 - x_1x_2)}{(1 - dx_1x_2y_1y_2)}.$$

4.3. **Details.** Similar to the case of Weierstrass curves, we wrote two PARI scripts to generate the database of Edwards curves. The first generates Edwards curves defined over a prime finite field p whose coefficient d is not a square in $\mathbb{F}_p$ and whose group of $\mathbb{F}_p$-rational points has order four times a prime. To prove that the curve is not specially chosen, we do not select d directly, but rather generate a number r_d which is hashed to give d. To compute the number of rational points we use the SEA algorithm after transforming the Edwards curve into Weierstrass form (not the short Weierstrass form). More precisely, the Edwards curve

$$x^2 + y^2 = 1 + dx^2y^2$$

is birationally equivalent to the smooth Weierstrass curve

$$y^2 = x^3 + a_2 x^2 + a_4 x$$

where

$$a_2 = 2\frac{(1+d)}{(1-d)^2} \quad \text{and} \quad a_4 = \frac{1}{(1-d)^2}.$$

We apply SEA to this last curve and we test if the number of rational points is of the right form. Unfortunately, the standard early-abort procedure cannot be used here because the order of group of rational points of an Edwards curve is a multiple of 4. On the other hand, for efficiency reasons we cannot deactivate the early-abort capability. To circumvent this problem we applied a patch to the PARI implementation of SEA in order to disable the early-abort when a factor of 2 is found.

As in the case Weierstrass curves, when a candidate curve is found we test if the curve is of a type known to be insecure. We also compute a point on the curve which is not of order 1, 2 or 4. To do that we draw a random number r, compute the hash g_x of r, and test if $z = (1 - g_x^2)/(1 - dg_x^2)$ is a square. If z is square, we compute g_y to be a square root of z, otherwise we repeat the process with a new r.

The second script verifies whether a curve is correctly generated and computes the exact order of the chosen point on the curve. We know that the point $(0, 1)$ is always the neutral element on an Edwards curve. The opposite of (x, y) is $(-x, y)$. There is an element of order two: $(0, -1)$ and two elements of order four: $(1, 0)$, $(-1, 0)$. So it is very easy to detect elements of order 1, 2 and 4. If we exclude these four elements, the other elements are of order $n = 4u$, $n/2 = 2u$ or $n/4 = u$ where u is prime. For each accepted curve, we choose a point (g_x, g_y) of order u, $2u$ or $4u$. We fill in a field t in the descriptor file of the curve such that $t = 0$ if the order is n, $t = 1$ if the order is $n/2$, and $t = 2$ if the order is $n/4$. In fact, knowing an element (g_x, g_y) on the curve and its t-value, we can deduce elements of any order. More precisely, if g has order n, then $2g$ has order $n/2$ and $4g$ has order $n/4$; if g has order $n/2$, $2g$ has order $n/4$ and $g + (0, -1)$ has order n; and if g has order $n/4$, $g + (0, -1)$ has order $n/2$ and $g + (1, 0)$ has order n.

The first script generates files giving the description of a curve in the following form

```
p = 17788785049862795200150516910406025137
    463828480015848539718291306993861084899
```

```
p = 17788785049862795200150516910406025137
    463828480015848539718291306993861084899
n = 17788785049862795200150516910406025137
    363578126680481424741935402610840792044
d = 37969516109524189464148380139464025406
    59352227509671351658573117542984656493
rd= 86918087186841376244437356659969366922
    40583232324910500040371199339620074813
gx= 19866051186693892783831850190823171157
    67420240937840666424031679646376737334
gy= 13522141226273509754871071682844347818
    526232922984052207011535368467814622472
r = 11437956621720228291212199612953420381
    679188428091051450834331532002067513477
t = 2
```

where p, n, d, r_d, g_x, g_y, r, t have been previously defined.

Given such a file the test script verifies that

- ◇ p is a prime;
- ◇ $p \equiv 3 \pmod 4$;
- ◇ n is four times a prime;
- ◇ d is not a square;
- ◇ $d \equiv \text{Hash}(r_d) \pmod p$;
- ◇ $g_x \equiv \text{Hash}(r) \pmod p$;
- ◇ n is the number of rational points of the curve;
- ◇ the curve is not known to be insecure;
- ◇ the point $g = (g_x, g_y)$ is on the curve; and
- ◇ the point g has order $n/2^t$.

4.4. **Database of elliptic curves in Edwards form.** Using the scripts described in the previous section we have constructed a database of elliptic curves in Edwards form. In the folders "ce*xxx*", where *xxx*=256, 384 or 512, one can find the files "e*xxx*-001.gp", . . . , "e*xxx*-018.gp". These files describe Edwards curves with *xxx* bits in the format given in the previous section. This database can also be found on the website

`http://galg.acrypta.com`

5. Conclusion

Using PARI we have built two databases containing data for elliptic curves in Weierstrass and Edwards forms respectively. These curves are of cryptographic strength and thus greatly increase the number of curves available for use in public key cryptography. Not only do the databases provide alternatives to the few curves provided by NIST, they also include Edwards curves, whose simple, unified addition formulae permit particularly fast arithmetic.

References

[BBJ+08] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. Cryptology ePrint Archive, Report 2008/013, 2008. http://eprint.iacr.org/.

[NIS09] NIST. NIST-FIPS 186-3 (website), 2009. http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf.

[NSA09] NSA. NSA Suite B Cryptography (website), 2009. http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml.

Institut de Mathématiques de Luminy, case 930, F13288 Marseille cedex 9, France
E-mail address: hlaw@iml.univ-mrs.fr

Institut de Mathématiques de Luminy, Campus de Luminy, Case 907, 13288 MARSEILLE Cedex 9
E-mail address: robert.rolland@acrypta.fr