Complete and Unified Group Laws are not Enough for Elliptic Curve Cryptography

Graham Enos[1]

[1]Dept. of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223, genos@uncc.edu

Abstract

We analyze four recently proposed normal forms for elliptic curves. Though these forms are mathematically appealing and exhibit some cryptographically desirable properties, they nonetheless fall short of cryptographic viability, especially when compared to various types of Edwards Curves. In this paper, we present these forms and demonstrate why they fail to measure up to the standards set by Edwards Curves.

1 Introduction

In several recent publications—[6], [9], [4], and [8]—normal forms for elliptic curves are proposed that have some advantages over the typical Weierstrass versions, like faster, *complete*, or *unified* group laws. Though all of these constructions are very interesting from a purely mathematical point of view, they fall short of suitability for cryptographic application if judged against Edwards Curves.

As we shall see, these constructions exhibit weaknesses that fall into one of two categories: either their group law is not symmetric, so commutativity is hard to see (though of course still present), or their atypical choice of neutral point obfuscates the result of adding a point and the neutral element. In both cases, one has to resort to working modulo the curve equation (or more precisely, modulo the ideal generated by the curve equation in the appropriate polynomial ring) to see that these computations behave the way one expects. This means that elementary operations, the results of which should be immediately apparent, cannot be implemented programmatically in a simple way; even simple work must involve unnecessary checks and reductions. This extra work will at best slow down a cryptosystem, and at worst could leak enough side-channel information to severely weaken the system.

In what follows, we first discuss the nature of the two weaknesses exhibited by these normal forms and how curves in the Edwards family avoid them. We then devote a section to each of the normal forms and demonstrate how they each fall prey to one of these weaknesses.

2 Two Weaknesses & How Edwards Curves Avoid Them

Edwards Curves have generated a lot of excitement in the cryptographic community because their group laws are *unified* and *complete*. That is, the group law can just as easily be used for doubling a point as

adding two different points, and any two points can be added with a single formula (there are no exceptional points). These properties make them attractive for implementation, since stronger immunity to side-channel and exceptional operation analysis and attacks are "baked in" to the elliptic curve implementation from the very start, from the mathematical foundation, as opposed to tacked on later (if at all).

Recall that Edwards Curves, originally presented by Edwards in [5] and expanded upon by Bernstein and Lange in [2], are elliptic curves over a field of characteristic not equal to two of the form

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

with some restrictions on c and d and have the (affine) group law

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{c(1 + dx_1x_2y_1y_2)}, \frac{y_1y_2 - x_1x_2}{c(1 - dx_1x_2y_1y_2)} \right) \tag{1}$$

Binary Edwards Curves ([3]) are elliptic curves over a field of characteristic two of the form

$$d_1(x + y) + d_2(x^2 + y^2) = (x + x^2)(y + y^2)$$

whose group law is (the slightly more complicated but still symmetric) $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$ such that

$$\begin{aligned} x_3 &= \frac{d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)} \\ y_3 &= \frac{d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)} \end{aligned} \tag{2}$$

Finally, Twisted Edwards Curves are elliptic curves over a non-binary field of the form

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

with group law

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \right) \tag{3}$$

All of four of the normal forms we examine have group laws that are purported to be unified and complete (at least, on a specified subgroup). They fail to live up to the Edwards Standard in other ways, however. A few of these normal forms have group laws that are asymmetric; that is, the equations for adding two points P and Q involve their coordinates in such a fashion that it's not obvious that $P + Q$ is the same as $Q + P$, even though it is true (because addition of two rational points on an elliptic curve is commutative). None of the three major Edwards Curve types—the original one put forward in [5] and [2], binary curves presented in [3], or twisted curves from [1]—exhibit this flaw. All three of the Edwards group laws (1), (2), and (3) are symmetric with respect to their inputs; one can clearly see that $(x_1, y_1) + (x_2, y_2)$ is the same as $(x_2, y_2) + (x_1, y_1)$ without any extra work, simply because of the commutativity of field addition and multiplication. This means that any implementation of these laws in computer code will be much less complex than they otherwise could be if extra work were needed to demonstrate this simple fact.

The other weakness exhibited by some of the normal forms we examine is the atypical choice of neutral

element. For some of the normal forms we present below, the neutral element is such that it's not immediately obvious that $\mathcal{O} + P = P + \mathcal{O} = P$. For Edwards Curves, the neutral element is $(0, 1)$. It's clear that this can be substituted into group law (1) in either position and the result will always be the other point; that is, it's immediately clear that $(0, 1)$ is indeed the neutral element for (1). Similarly, Binary Edwards Curves have neutral point $(0, 0)$, while Twisted Edwards Curves have neutral point $(0, 1)$. Substituting these into either position in their group laws—(2) and (3) respectively—clearly demonstrates that they are the correct neutral elements. For some of the normal forms presented below, it is not so apparent that the stated neutral element is correct.

3 Farashahi & Joye's Construction

The first curve we'll consider is Farashahi and Joye's Generalized Hessian curve presented in [6]. This curve has the form

$$H_{c,d} : x^3 + y^3 + c = dxy$$

or, in projective coordinates,

$$\mathbf{H}_{c,d} : X^3 + Y^3 + cZ^3 = dXYZ$$

The group of rational points on this curve has neutral element $\mathcal{O} = (1 : -1 : 0)$.

The authors present some unified addition formulas for $\mathbf{H}_{c,d}$ (equations (9) and (10) in [6]). If we let $P = (X_1 : Y_1 : Z_1)$ and $Q = (X_2 : Y_2 : Z_2)$ be two points on $\mathbf{H}_{c,d}$, then according to their first equation we have $P + Q = (X_3 : Y_3 : Z_3)$ where

$$\begin{aligned} X_3 &= cY_2Z_1^2Z_2 - X_1X_2^2Y_1 \\ Y_3 &= X_2Y_1^2Y_2 - cX_1Z_1Z_2^2 \\ Z_3 &= X_1^2X_2Z_2 - Y_1Y_2^2Z_1 \end{aligned}$$

Using these formulas, we have $\mathcal{O} + P = (X_1^2 : X_1Y_1 : X_1Z_1)$; while at first blush this may not seem to be the same as P, projective points are really equivalence classes, so this is of course the same point as we would get dividing all three positions by X_1,[1] viz. $(X_1 : Y_1 : Z_1) = P$. Similarly, $P + \mathcal{O} = (-X_1Y_1 : -Y_1^2 : -Y_1Z_1) \equiv (X_1 : Y_1 : Z_1) = P$.

The real trouble with this construction, however, comes from comparing $P + Q$ with $Q + P$. Let $(X_4 : Y_4 : Z_4) = Q + P$, so

$$\begin{aligned} X_4 &= cY_1Z_1Z_2^2 - X_1^2X_2Y_2 \\ Y_4 &= X_1Y_1Y_2^2 - cX_2Z_1^2Z_2 \\ Z_4 &= X_1X_2^2Z_1 - Y_1^2Y_2Z_2 \end{aligned}$$

Since we need point addition to be commutative, this should be equal (or at least equivalent) to $P + Q$. Suppose that all of $P, Q, P + Q$, and $Q + P$ are finite points, so their Z coordinate is nonzero. Then we need

[1] X_1 cannot be zero, or else $\mathcal{O} + P$ would be a singular point on $\mathbf{H}_{c,d}$, something which the authors show is impossible.

the following:

$$\frac{X_3}{Z_3} = \frac{cY_2Z_1^2Z_2 - X_1X_2^2Y_1}{X_1^2X_2Z_2 - Y_1Y_2^2Z_1} = \frac{cY_1Z_1Z_2^2 - X_1^2X_2Y_2}{X_1X_2^2Z_1 - Y_1^2Y_2Z_2} = \frac{X_4}{Z_4}$$

This is true if and only if $X_3Z_4 - X_4Z_3 = 0$; i.e. if and only if

$$-X_1X_2\left(cX_1Y_1Z_1Z_2^3 - cX_2Y_2Z_2Z_1^3 - X_1^3X_2Y_2Z_2 + X_1Y_1Z_1X_2^3 + X_1Y_1Z_1Y_2^3 - X_2Y_2Z_2^3\right) = 0 \qquad (4)$$

Suppose furthermore that $X_1X_2 \neq 0$; then we need the larger factor of (4) to be zero, which isn't immediately apparent. Factoring and simplifying, the larger factor in (4) becomes

$$(X_1Y_1Z_1)(cZ_2^3) - (X_2Y_2Z_2)(cZ_1^3) - (X_2Y_2Z_2)(X_1^3) + (X_1Y_1Z_1)(X_2^3) + (X_1Y_1Z_1)(Y_2^3) - (X_2Y_2Z_2)(Y_1^3)$$

or

$$(X_1Y_1Z_1)(X_2^3 + Y_2^3 + cZ_2^3) - (X_2Y_2Z_2)(X_1^3 + Y_2^3 + cZ_1^3)$$

Working modulo the curve equation, we know $X^3 + Y^3 + cZ^3 = dXYZ$, which implies our work simplifies to

$$(X_1Y_1Z_1)(dX_2Y_2Z_2) - (X_2Y_2Z_2)(dX_1Y_1Z_1)$$

which is, at last, zero.

Similarly,

$$\begin{aligned} Y_3Z_4 - Y_4Z_3 &= (X_1^2X_2Z_3 - Y_1Y_2^2Z_1)(cX_2Z_1^2Z_2 - X_1Y_1Y_2^2) - (X_1X_2^2Z_1 - Y_1^2Y_2Z_2)(cX_1Z_1Z_2^2 - X_2Y_1^2Y_2) \\ &= Y_1Y_2(cX_1Y_1Z_1Z_2^3 - cX_2Y_2Z_1^3Z_2 + X_1X_2^3Y_1Z_1 - X_1^3X_2Y_2Z_2 + X_1Y_1Y_2^3Z_1 - X_2Y_1^3Y_2Z_2) \\ &= Y_1Y_2\left[(X_1Y_1Z_1)(X_2^3 + Y_2^3 + cZ_2^3) - (X_2Y_2Z_2)(X_1^3 + Y_1^3 + cZ_1^3)\right] \end{aligned}$$

If $Y_1Y_2 \neq 0$, then this can only be zero if we resort to the curve equation, getting

$$Y_1Y_2\left[(X_1Y_1Z_1)(dX_2Y_2Z_2) - (X_2Y_2Z_2)(dX_1Y_1Z_1)\right]$$

Thus $P + Q$ does indeed equal $Q + P$; note, however, that in order to reach this conclusion, we had to perform substitutions using $\mathbf{H}_{c,d}$'s equation. That is, this equality was not apparent from the outset, but rather required working modulo the ideal generated by the curve equation in the appropriate polynomial ring. Thus, this addition is true, and even mathematically pleasing, but not cryptographically viable. Such reductions would complicate any computer code implementation of this group—at best leading to slow execution speed, and at worst causing side-channel leaks that could potentially lead to a break in the cryptosystem. This elliptic curve is not as safe as Edwards Curves when it comes to the concerns of cryptographic implementation.

4 Wang, Tang, & Yang's Construction

In [8], the authors explore the curve

$$M_d : x^2y + xy^2 + dxy + 1 = 0$$

and its homogeneous projective version

$$\widetilde{M_d} : X^2Y + XY^2 + dXYZ + Z^3 = 0$$

over a field of characteristic greater than 3. The neutral element of the group of rational points on this curve is $(1 : -1 : 0)$ Though their affine group law seems to have little trouble in the symmetry department, the projective group law $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$ where

$$\begin{aligned} X_3 &= X_1X_2(Y_1Z_2 - Y_2Z_1)^2 \\ Y_3 &= Y_1Y_2(X_1Z_2 - X_2Z_1)^2 \\ Z_3 &= (X_1Z_2 - X_2Z_1)(Y_1Z_2 - Y_2Z_1)(X_2Y_2Z_1^2 - X_1Y_1Z_2^2) \end{aligned}$$

is problematic with regards to the neutral element. Suppose we wished to add the point $P = (X : Y : Z)$ (a finite point, so $Z \neq 0$) and the neutral element $(1 : -1 : 0)$; then we'd have

$$\begin{aligned} X_3 &= X \cdot 1(Y \cdot 0 - (-1) \cdot Z)^2 \\ Y_3 &= Y \cdot (-1)(X \cdot 0 - 1 \cdot Z)^2 \\ Z_3 &= (X \cdot 0 - 1 \cdot Z)(Y \cdot 0 - (-1) \cdot Z)(1 \cdot (-1) \cdot Z^2 - X \cdot Y \cdot 0^2) \end{aligned}$$

which simplifies to $(XZ^2 : -YZ^2 : Z^4) \equiv (X : -Y : Z^2)$. Except in very special circumstances, this is of course not equal to $(X : Y : Z)$; moreover, it's not apparent how resorting to the curve equation will even help here.

There are even more problems here, though. For example, $\mathcal{O} + P = (XZ^2 : -YZ^2 : -Z^4) \equiv (X : -Y : -Z^2)$, $\mathcal{O} + \mathcal{O} = (0 : 0 : 0)$, and $P + P = (0 : 0 : 0)$, so this law is not unified (contrary to the claims of [8]). These problems can be seen by running the following Sage [7] script:

```
var('d x y z')
R.<d, x, y, z> = GF(17^17, 'a')[]
S = R.quotient([x^2 * y + x * y^2 + d * x * y * z + z^3])

def add((x1, y1, z1), (x2, y2, z2)):
     x3 = S(x1 * x2 * (y1 * z2 - y2 * z1)^2)
     y3 = S(y1 * y2 * (x1 * z2 - x2 * z1)^2)
     z3 = S((x1 * z2 - x2 * z1) *
            (y1 * z2 - y2 * z1) *
            (x2 * y2 * z1^2 - x1 * y1 * z2^2))
     return (x3, y3, z3)
```

```
o, p = (1, -1, 0), (x, y, z)

for pair in cartesian_product_iterator([(o, p)] * 2):
    print "add{0}\t=\t{1}".format(pair, add(*pair))
```

Hence this curve is not a suitable candidate for cryptographic implementation.

5 Wu, Tang, & Feng's Construction

In [9], presented at Indocrypt 2012, Wu, Tang, & Feng introduce the curve

$$S_t : x^2y + xy^2 + txy + x + y = 0$$

and its projective version

$$X^2Y + XY^2 + tXYZ + XZ^2 + YZ^2 = 0$$

and study its properties over a binary field. In their paper, they define the projective point $\mathcal{O} = (1 : 1 : 0)$ as the neutral element.

Suppose that we wish to add the finite projective point $(X : Y : 1)$ to $\mathcal{O}$ using the formulas given in [9] to obtain the point $(X_3 : Y_3 : Z_3)$; moreover, suppose that $X \neq Y$ and both are nonzero. Then

$$\begin{aligned}
X_3 &= (Y \cdot 1 + 1 \cdot 0)\left[(X \cdot 1 + Y \cdot 1)(Y \cdot 0 + 1 \cdot 1) + t \cdot Y \cdot 1 \cdot (1 \cdot 0 + X \cdot 1)\right] \\
&= X\left[(X + Y) + tXY\right] \\
&= X(X + Y + tXY) \\
Y_3 &= (Y \cdot 1 + 1 \cdot 0)\left[(X \cdot 1 + Y \cdot 1)(X \cdot 0 + 1 \cdot 1) + t \cdot X \cdot 1(1 \cdot 0 + Y \cdot 1)\right] \\
&= Y\left[(X + Y) + tXY\right] \\
&= Y(X + Y + tXY) \\
Z_3 &= (X \cdot 1 + Y \cdot 1)(X \cdot 1 + 1 \cdot 0)(Y \cdot 1 + 1 \cdot 0) \\
&= XY(X + Y)
\end{aligned}$$

Therefore $(X_3 : Y_3 : Z_3)$ is equivalent to

$$\left(\frac{X(X + Y + tXY)}{XY(X + Y)} : \frac{Y(X + Y + tXY)}{XY(X + Y)} : 1\right) = \left(\frac{X + Y + tXY}{Y(X + Y)} : \frac{X + Y + tXY}{X(X + Y)} : 1\right)$$

From the curve equation, we know that $X + Y + tXY = X^2Y + XY^2 = XY(X + Y)$, so $(X_3 : Y_3 : Z_3)$ is indeed equal to $(X : Y : 1)$. Note, however, that this result only occurs if we take into account the curve equation. For something as simple as adding a point to the neutral element, having to modulo the curve equation to show that $(X : Y : 1) + \mathcal{O} = (X : Y : 1)$ is unnecessarily complicated.

6 Diao & Fouotsa's Construction

In [4], presented at "Journes C2: Codage et Cryptographie" in September 2012, Diao & Fouotsa introduce the curve

$$\mathcal{E}_\lambda : 1 + x^2 + y^2 + x^2y^2 = \lambda xy$$

which is valid over a field of any characteristic. Their paper is very detailed, and the construction involves some interesting work with Theta functions. Unfortunately, due to the asymmetry of the group law they present, this construction also falls short of the cryptographic applicability of Edwards Curves.

Suppose we wished to add two points (x_1, y_1) and (x_2, y_2); it shouldn't matter in which order we add them, because the group law should be commutative. By the work in [4], we have

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 + y_1x_2y_2}{y_2 + x_1y_1x_2}, \frac{x_1x_2 + y_1y_2}{1 + x_1x_2y_1y_2}\right)$$

while

$$(x_2, y_2) + (x_1, y_1) = \left(\frac{x_2 + x_1y_1y_2}{y_1 + x_1x_2y_2}, \frac{x_1x_2 + y_1y_2}{1 + x_1x_2y_1y_2}\right)$$

The second coordinates of these points are obviously equal to each other, but we also need the first ones to be equal. This is the case if and only if

$$\begin{aligned}\frac{x_1 + y_1x_2y_2}{y_2 + x_1y_1x_2} = \frac{x_2 + x_1y_1y_2}{y_1 + x_1x_2y_2} &\iff (x_1 + y_1x_2y_2)(y_1 + x_1x_2y_2) = (x_2 + x_1y_1y_2)(y_2 + x_1x_2y_1) \\ &\iff x_1y_1 + x_2(x_1^2 + y_1^2 + x_1y_1x_2y_2) = x_2y_2 + x_1y_1(x_2^2 + y_2^2 + x_1x_2y_1y_2)\end{aligned}$$

Using the curve equation:

$$\begin{aligned}&\iff x_1y_1 + x_2y_2(1 + x_1^2y_1^2 + x_1x_2y_1y_2) = x_2y_2 + x_1y_1(1 + x_2^2y_2^2 + x_1x_2y_1y_2) \\ &\iff x_1y_1 + x_2y_2 + x_1^2x_2y_1^2y_2 + x_1x_2^2y_1y_2^2 = x_2y_2 + x_1y_1 + x_1x_2^2y_1y_2^2 + x_1^2x_2y_1^2y_2\end{aligned}$$

So we have $(x_1, y_1) + (x_2, y_2) = (x_2, y_2) + (x_1, y_1)$ as we needed. Note that proving this simple fact required resorting to working modulo the curve equation again (i.e. modulo the ideal generated by the curve equation in the polynomial ring $\mathbb{F}_2^n[x_1, x_2, y_1, y_2]$).

7 Conclusion

Following the excitement regarding the various types of Edwards Curves, normal forms for elliptic curves have been presented and explored with an eye to improving upon one characteristic or another of Edwards Curves while maintaining the same safety and security afforded by their complete and unified group laws. It turns out that there is more to being as safe as Edwards Curves than just being complete (on a subgroup or over the whole group) and unified, however. As we have demonstrated, four recently proposed normal forms exhibit weaknesses that don't show up in Edwards Curves: either their group laws are not symmetric or they use an unusual choice of neutral element.[2] Both of these weaknesses mean that we must reduce modulo

[2] In fact, one normal form's troubles extend even deeper.

their curve equations to demonstrate even elementary facts, like $\mathcal{O} + P = P$ or $P + Q = Q + P$. This extra work will complicate any computer implementation, leading to slower execution speed and perhaps leakage of information through side channels. The main advantage Edwards Curves have for implementation is their incorporating safety and security from the ground up; these newer normal forms do not measure up when it comes to suitability for cryptographic implementation.

References

[1] D. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, *Twisted edwards curves*, Progress in Cryptology–AFRICACRYPT 2008 (2008), 389–405.

[2] D. Bernstein and T. Lange, *Faster addition and doubling on elliptic curves*, Advances in cryptology–ASIACRYPT 2007 (2007), 29–50.

[3] D. Bernstein, T. Lange, and R. Rezaeian Farashahi, *Binary edwards curves*, Cryptographic Hardware and Embedded Systems–CHES 2008 (2008), 244–265.

[4] O. Diao and E. Fouotsa, *Edwards model of elliptic curves defined over any fields*, Tech. report, Cryptology ePrint Archive: Report 2012/346. http://eprint.iacr.org/2012/346, 2012.

[5] H.M. Edwards, *A normal form for elliptic curves*, Bulletin of the American Mathematical Society **44** (2007), no. 3, 393–422.

[6] R. Farashahi and M. Joye, *Efficient arithmetic on hessian curves*, Public Key Cryptography–PKC 2010 (2010), 243–260.

[7] W. A. Stein et al., *Sage Mathematics Software (Version 5.0)*, The Sage Development Team, 2012, `http://www.sagemath.org`.

[8] B. WANG, C. TANG, and Y. YANG, *A new model of elliptic curves with effective and fast arithmetic*, Journal of Computational Information Systems **8** (2012), no. 10, 4061–4067.

[9] H. Wu, C. Tang, and R. Feng, *A new model of binary elliptic curves with fast arithmetic*, Tech. report, Cryptology ePrint Archive: Report 2010/608. http://eprint.iacr.org/2010/608, 2010.