Halving on Binary Edwards Curves ⋆

Qiping Lin and Fangguo Zhang

School of Information Science and Technology
Sun Yat-sen University, Guangzhou 510275, P.R.China
isszhfg@mail.sysu.edu.cn

Abstract. Edwards curves have attracted great interest for their efficient addition and doubling formulas. Furthermore, the addition formulas are strongly unified or even complete, i.e., work without change for all inputs. In this paper, we propose the first halving algorithm on binary Edwards curves, which can be used for scalar multiplication. We present a point halving algorithm on binary Edwards curves in case of $d_1 \neq d_2$. The halving algorithm costs about $3I+5M+4S$, which is slower than the doubling one. We also give a theorem to prove that the binary Edwards curves have no minimal two-torsion in case of $d_1 = d_2$, and we briefly explain how to achieve the point halving algorithm using an improved algorithm in this case. Finally, we apply our halving algorithm in scalar multiplication with ω-coordinate using Montgomery ladder.

Keywords: Point halving, Binary Edwards curves, ω-coordinate, Montgomery ladder.

1 Introduction

Edwards [8] proposed a new normal form for elliptic curves and gave an addition law over a non-binary field k. Later, Bernstein and Lange presented fast explicit formulas for addition and doubling on Edwards curve [5], and showed that those formulas can save time in elliptic curve cryptography. In [4], Bernstein *et al.* generalized the Edwards addition law to cover more curves. The addition formulas of Edwards curves are "unified" or even "complete". "Complete" is stronger than "unified": it means that the addition formulas work for all pairs of input points. There are no troublesome points at infinity. For these reasons and the addition and doubling formulas are very efficient, Edwards curves have attracted great interest in recent years.

Unfortunately, these coordinates are not elliptic over fields k with char(k)=2. Bernstein, Lange and Farashahi [6] introduced a new method of carrying out computations on binary elliptic curves, i.e., elliptic curves over fields k with char(k)=2. And they presented their addition and doubling formulas in two ways. The first one used traditional coordinates (X, Y, Z) and the second one

⋆ This work is supported by the the National Natural Science Foundation of China (No. 60773202, 60633030) and 973 Program (2006CB303104)

used ω-coordinate. It was shown that if $n \geq 3$ then every ordinary elliptic curve over F_{2^n} is birationally equivalent to a so-called complete binary Edwards curve.

It is well known that the double-and-add algorithm is essential to the computation of cryptosystems of elliptic curves. And the algorithm is based on two group operations: the addition of two distinct group elements and the computation of the double of an element. From 1999, there has an equivalence algorithm to realize the function of double-and-add in binary fields. The algorithm is called half-and-add by Knudsen [10] and Schroeppel [13]. The halving algorithm relies on the computation of the "half" of a group element. And halving algorithm is more efficient than doubling one over fields k with char(k)=2, because a square root, the traces and half-traces can be computed very efficiently in these fields. In other fields k with char(k) $\neq 2$, the halving algorithm cannot do better than the doubling one.

A point halving algorithm is one of the effective algorithms on ECC and the algorithm tries to find a point P such that $2P = Q$ for a given point Q. Knudsen and Schroeppel independently proposed a point halving algorithm for ECC over binary fields F_{2^n} ([10], [13]). Their algorithm is faster than a doubling algorithm. The primary focus of Knudsen's work was with curves with cofactor of 2. And later in [12], King and Rubin extended the point halving algorithm of Knudsen to do with elliptic curves with cofactor of 4 or 2^k where $k \geq 2$. And there has been growing consideration of the point halving algorithm, such as [9]. There was also an application for Koblitz curve [2] and some extensions to HECC ([11], [3], [7]).

Our contributions. In this paper, we propose a halving algorithm on binary Edwards curves in case of $d_1 \neq d_2$. Then we prove that the binary Edwards curves have no minimal two-torsion in the case of $d_1 = d_2$. So in this case we have to use the improved version of the elliptic curve for curves with cofactor of 4, i.e., minimal four-torsion in [12]. Using Montgomery ladder, we apply our halving algorithm in scalar multiplication with ω-coordinate. Our algorithm is the first one to try to speed up the scalar multiplication using point halving on Edwards curves.

Organization. The remainder of this paper is organized as follows: in section 2 we recall the binary Edwards curves, the addition and doubling formulas based on both (x, y)-coordinate and ω-coordinate. In section 3 we show how we developed the halving formulas, and also present the halving algorithm in case of $d_1 \neq d_2$. In section 4 we give a theorem to prove that the binary Edwards curves have no minimal two-torsion in the case of $d_1 = d_2$, and briefly explain how to use the improved halving algorithm of [12] to achieve the algorithm in this case. In section 5 we apply our halving algorithm in scalar multiplication with ω-coordinate. The conclusion of this paper is given in section 6.

2 Binary Edwards Curves

Definition 1. (Binary Edwards Curves [6]) *Let k be a field with char(k)=2. Let d_1, d_2 be elements of k with $d_1 \neq 0$ and $d_2 \neq d_1^2 + d_1$. The binary Edwards* *curve with coefficients d_1 and d_2 is the affine curve*

$$E_{B,d_1,d_2} : d_1(x + y) + d_2(x^2 + y^2) = xy + xy(x + y) + x^2y^2$$

The sum of two points (x_1, y_1), (x_2, y_2) on E_{B,d_1,d_2} is the point (x_3, y_3) defined as follows:

$$x_3 = \frac{d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)},$$

$$y_3 = \frac{d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}.$$

When (x_1, y_1)=(x_2, y_2), then the doubling formulas are

$$\begin{aligned} x_3 &= 1 + \frac{d_1 + d_2(x_1^2 + y_1^2) + y_1^2 + y_1^4}{d_1 + x_1^2 + y_1^2 + (d_2/d_1)(x_1^4 + y_1^4)}, \\ y_3 &= 1 + \frac{d_1 + d_2(x_1^2 + y_1^2) + x_1^2 + x_1^4}{d_1 + x_1^2 + y_1^2 + (d_2/d_1)(x_1^4 + y_1^4)}. \end{aligned} \tag{1}$$

It takes $1I + 2M + 4S$. In particular, if $d_1 = d_2$, then the doubling formulas are

$$\begin{aligned} x_3 &= \frac{d_1(x_1^2 + y_1^2) + x_1^2 + x_1^4}{d_1 + x_1^2 + y_1^2 + x_1^4 + y_1^4}, \\ y_3 &= x_3 + 1 + \frac{d_1}{d_1 + x_1^2 + y_1^2 + x_1^4 + y_1^4}. \end{aligned} \tag{2}$$

2.1 Differential addition

Bernstein *et al.* [6] used the idea of Montgomery ladder to present fast explicit formulas for ω-coordinate differential addition on binary Edwards curves. The Montgomery ladder is one of the most popular scalar-multiplication methods. It has several attractive features: it is fast, its double-and-add structure is uniform and it can fit into extremely small hardware.

"Differential addition" means computing $(2m+1)P$ given $(m+1)P, mP, P$, or computing $2mP$ given $mP, mP, 0P$.

Let (x_2, y_2) be a point on the binary Edwards curve E_{B,d_1,d_2}. Assumed that the sum $(x_2, y_2) + (x_2, y_2)$ is defined. Write $(x_4, y_4) = (x_2, y_2) + (x_2, y_2)$, and write $\omega_i = x_i + y_i$. Then $d_1^2 + d_1\omega_2^2 + d_2\omega_2^4 \neq 0$ and

$$\omega_4 = \frac{d_1\omega_2^2 + d_1\omega_2^4}{d_1^2 + d_1\omega_2^2 + d_2\omega_2^4} = \frac{\omega_2^2 + \omega_2^4}{d_1 + \omega_2^2 + (d_2/d_1)\omega_2^4}. \tag{3}$$

Assumed that $(x_1, y_1), P = (x_2, y_2), Q = (x_3, y_3), (x_5, y_5)$ are points on E_{B,d_1,d_2} satisfying $(x_1, y_1) = Q - P$ and $(x_5, y_5) = P + Q$, and write $\omega_i = x_i + y_i$. The addition is

$$\omega_5 = \frac{\omega_2\omega_3(1 + \omega_2 + \omega_3) + (\omega_2\omega_3)^2}{d_1 + \omega_2\omega_3(1 + \omega_2 + \omega_3) + (d_2/d_1)(\omega_2\omega_3)^2} + \omega_1$$

One can recover $2P$ from $Q-P, \omega(P), \omega(Q)$. If $\omega_1^2+\omega_1 \neq 0$, let $A=\omega_1+\omega_2$, $B=\omega_1\omega_2$ then

$$x_2^2+x_2 = \frac{\omega_3\left(d_1+B(A+1)+\frac{d_2}{d_1}B^2\right)+d_1A+(y_1^2+y_1)(\omega_2^2+\omega_2)}{\omega_1^2+\omega_1}.$$

We can use this formula to compute $2(x_2, y_2)$ given $x_1, y_1, \omega_2, \omega_3$. In particular, one can recover $2mP$ given $P, \omega(mP), \omega((m+1)P)$.

3 Point Halving in Case of $d_1 \neq d_2$

In this section, we describe how we developed the halving formulas and give the halving algorithm in case of $d_1 \neq d_2$. We assume the group order of E_{B,d_1,d_2} is $2r$, where r is odd. $E_{B,d_1,d_2}(F_{2^d}) = G \times E[2]$ where d is odd and $Q=(x_4, y_4)$ in G.

Let $Q=(x_4,y_4)$, $P=(x_2,y_2)$, $Q=2P$, the halving formulas try to get $P=(x_2,y_2)$ by given $Q=(x_4,y_4)$. From the doubling formulas (1), let

$$\lambda = \frac{1}{d_1+(x_2+y_2)^2+(d_2/d_1)(x_2+y_2)^4}$$

we have

$$\begin{aligned} x_4 &= 1+\lambda(d_1+d_2(x_2+y_2)^2+y_2^2+y_2^4) \\ y_4 &= 1+\lambda(d_1+d_2(x_2+y_2)^2+x_2^2+x_2^4). \end{aligned} \tag{4}$$

Then add these two functions to get

$$\lambda(x_2+y_2)^4+\lambda(x_2+y_2)^2 = x_4+y_4.$$

Substitute λ into the above function and let $X_2=x_2+y_2$, $X_4=x_4+y_4$, we can get

$$(1+\frac{d_2}{d_1}X_4)X_2^4+(1+X_4)X_2^2 = d_1X_4. \tag{5}$$

3.1 Halving formulas

When $X_4+1=0$, the function (5) equals to

$$X_2^4 = \frac{d_1}{1+d_2/d_1}.$$

And we can get X_2 directly by computing the square root twice.

And when $1+\frac{d_2}{d_1}X_4=0$, i.e. $X_4=\frac{d_1}{d_2}$, the function (5) equals to

$$X_2^2 = \frac{d_1}{1+d_2/d_1}.$$

And we can get X_2 directly by computing the square root. Therefore, we assume that $X_4 \neq 1, d_1/d_2$ in the following. Now, given (x_4, y_4), we show how to compute (x_2, y_2) using our halving formulas.

Firstly, let $M = \frac{1}{(1+X_4)(d_1/d_2+X_4)} = \frac{1}{d_1/d_2+(1+d_1/d_2)X_4+X_4^2}$ and $T = X_2^2$. We can rewrite the function (5) as:

$$\frac{d_1 + d_2X_4}{d_1(1+X_4)}T^2 + T = \frac{X_4 d_1}{1+X_4}$$

Let $T_1 = \frac{d_1+d_2X_4}{d_1(1+X_4)}T = \frac{1+\frac{d_2}{d_1}X_4}{1+X_4}T$, then we get the function

$$\begin{aligned}
T_1^2 + T_1 &= \frac{d_1 + d_2X_4}{d_1(1+X_4)}\frac{d_1X_4}{1+X_4} \\
&= d_2(1 + \frac{1}{1+X_4})(1 + (1 + \frac{d_1}{d_2})\frac{1}{1+X_4}) \\
&= d_2(1 + M(\frac{d_1}{d_2} + X_4))(1 + (1 + \frac{d_1}{d_2})M(\frac{d_1}{d_2} + X_4)) \\
&= (1 + \frac{d_1}{d_2}M + MX_4)(d_2 + (d_1 + d_2)(\frac{d_1}{d_2}M + MX_4)) \\
&= d_2 + d_1(\frac{d_1}{d_2}M + MX_4) + (d_1 + d_2)(\frac{d_1}{d_2}M + MX_4)^2. \qquad (6)
\end{aligned}$$

And now we will solve the quadratic equation and check which solution is the right one.

Lemma 1. *An equation $ax^2 + bx + c = 0, (a, b \neq 0)$ in F_{2^d} can be simplified to be*

$$T^2 + T = c',$$

which has roots if and only if $Tr(c') = 0$. If d is odd, then one root of $T^2+T = c'$ is given by using half trace

$$t = \sum_{i=0}^{(d-3)/2} c'^{2^{2i+1}}.$$

When t is one root of $T^2 + T = c'$, then $t + 1$ is the other one.

□

We know the function (6) has a solution, for the function (6) comes from the function (5). This implies $\mathrm{Tr}(d_2+d_1(\frac{d_1}{d_2}M+MX_4)+(d_1+d_2)(\frac{d_1}{d_2}M+MX_4)^2)=0$. And there exist two solutions t_1 and $t_1 + 1$ which can be computed using half-trace. We will figure out which solution is the right one in the following.

Using the first one of the two solutions t_1, we can get the value of t from

$$\begin{aligned}
t &= \frac{d_1}{d_2}\left[1 + (1 + \frac{d_1}{d_2})\frac{1}{d_1/d_2 + X_4}\right] t_1 \\
&= \frac{d_1}{d_2}\left[1 + (1 + \frac{d_1}{d_2})(M + MX_4)\right] t_1,
\end{aligned}$$

Now, we want to check whether X_2 is the correct one. And we can just check whether

$$\begin{aligned}\mathrm{Tr}(d_2 + d_1\frac{1}{1+X_2} + (d_1+d_2)(\frac{1}{1+X_2})^2) = 0 \\ \Leftrightarrow \mathrm{Tr}(d_2(1+\frac{1}{1+X_2})) = 0 \\ \Leftrightarrow \mathrm{Tr}(d_2(1+\frac{1}{1+t})) = 0.\end{aligned}$$

The computations of the trace can be simplified, because the trace is a linear map and $\mathrm{Tr}(a^2)$=$\mathrm{Tr}(a)$ for any $a \in F_{2^d}$; see [1] for detail. If the above $\mathrm{Tr}(d_2(1+\frac{1}{1+t})) = 0$ holds, then the first solution t_1 is the right one. And we compute the correct $X_2 = \sqrt{t}$. If the trace is not zero, then the other solution t_1+1 is the right one. And we can compute the correct X_2 by

$$t = \frac{d_1}{d_2}\left[1 + (1+\frac{d_1}{d_2})(M + MX_4)\right](t_1+1),$$

$$X_2 = \sqrt{t}.$$

From the functions (4) we have

$$\begin{aligned}x_2^4 + x_2^2 + d_1 + d_2X_2^2 = \frac{y_4+1}{\lambda}, \\ \Rightarrow x_2^4 + x_2^2 = (y_4+1)(d_1 + X_2^2 + \frac{d_2}{d_1}X_2^4) + d_1 + d_2X_2^2.\end{aligned}$$

Let $x_2' = x_2^2$, and substitute it into the function above

$$x_2'^2 + x_2' = (y_4+1)(d_1 + X_2^2 + \frac{d_2}{d_1}X_2^4) + d_1 + d_2X_2^2$$

we get two solutions x_2' and $x_2'+1$ which can be computed using half-trace. Hence, $x_2 = \sqrt{x_2'}$ or $x_2 = \sqrt{x_2'+1}$, and $P = (\sqrt{x_2'}, \sqrt{x_2'} + X_2)$ or $P = (\sqrt{x_2'+1}, \sqrt{x_2'+1}+X_2)$. We can get the right point P with the following lemma.

Lemma 2. *Assumed the group order of E_{B,d_1,d_2} is $2r$, where r is odd. $E_{B,d_1,d_2}(F_{2^d})$ $= G \times E[2]$ where d is odd. Then a point $P = (x,y)$ is in G if and only if*

$$Tr(d_1^2 + d_2 + \frac{d_1(d_1^2+d_1+d_2)(x+y)}{xy + d_1(x+y)}) = 0.$$

Proof. A binary Edwards curve

$$E_{B,d_1,d_2}\colon d_1(x+y) + d_2(x^2+y^2) = xy + xy(x+y) + x^2y^2$$

is birational equivalence to the elliptic curves [6]:

$$E\colon v^2 + uv = u^3 + (d_1^2+d_2)u^2 + d_1^4(d_1^4+d_1^2+d_2^2).$$

And in [10], a point $Q = (u, v)$ on Weierstrass curves with character 2 is in G if and only if:

$$\exists\lambda \in F_{2^d} : \lambda^2 + \lambda = a + u$$

i.e. $\mathrm{Tr}(a + u) = 0$, where $a = d_1^2 + d_2$ and $u = \frac{d_1(d_1^2+d_1+d_2)(x+y)}{xy+d_1(x+y)}$.

$\square$

3.2 Halving algorithm

Let $Q = (x_4, y_4)$, $P = (x_2, y_2)$, $Q = 2P$, $X_2 = x_2 + y_2$, $X_4 = x_4 + y_4$, $M = \frac{1}{(1+X_4)(d_1/d_2+X_4)}$.

Algorithm 1. halving on binary Edwards curves in case of $d_1 \neq d_2$

INPUT: (x_4, y_4)
OUTPUT: (x_2, y_2)

1. $M \leftarrow \frac{1}{d_1/d_2+(1+d_1/d_2)X_4+X_4^2}$
2. $c_0 \leftarrow d_2 + d_1(\frac{d_1}{d_2}M + MX_4) + (d_1 + d_2)(\frac{d_1}{d_2}M + MX_4)^2$
3. $t_1 \leftarrow \sum_{i=0}^{(d-3)/2} c_0^{2^{(2i+1)}}$
4. $a_0 \leftarrow \frac{d_1}{d_2}[1 + (1 + d_1/d_2)(M + MX_4)]$
5. $t \leftarrow a_0 t_1$
6. $b_0 \leftarrow d_2 + d_2\frac{1}{1+t}$
7. if $\mathrm{Trace}(b_0) = 0$ then $X_2 \leftarrow \sqrt{t}$ else $X_2 \leftarrow \sqrt{t + a_0}$
8. $f_0 \leftarrow (y_4 + 1)(d_1 + X_2^2 + \frac{d_2}{d_1}X_2^4) + d_1 + d_2X_2^2$
9. $x_2' \leftarrow \sum_{i=0}^{(d-3)/2} f_0^{2^{(2i+1)}}$
10. $e_0 = \sqrt{x_2'}$
11. $g_0 \leftarrow d_1^2 + d_2 + \frac{d_1(d_1^2+d_1+d_2)X_2}{x_2'+(e_0+d_1)X_2}$
12. if $\mathrm{Trace}(g_0) = 0$ then return $P = (e_0, e_0 + X_2)$
    else return $P = (\sqrt{x_2' + 1}, \sqrt{x_2' + 1} + X_2)$

In case of $d_1 \neq d_2$, one halving algorithm takes $3I+5M+4S+2H+2SR+2T$, here we neglect the multiplications by the curves parameters. If we have a normal basis representation, the computations of squarings, square roots, half-traces and traces are trivial. Hence the cost of a halving algorithm can be simplified as $3I + 5M$. Note that a field multiplication by M for short and other field operations are expressed as follows: a squaring (S), an inversion (I), a square root (SR), a half trace (H), and a trace (T).

In case of $d_1 = d_2$, if we can do the same way as above, the halving algorithm may be efficient. Unfortunately, we can not half on these curves in the same way as the case of $d_1 \neq d_2$. We have the following theorem.

4 Point Halving in Case of $d_1 = d_2$

We can see in Appendix A of [10], a non-supersingular curve E is defined by an equation

$$v^2 + uv = u^3 + a_2u^2 + a_6, \quad a_2, a_6 \in F_{2^n},\ a_6 \neq 0.$$

We also use the notation T_{2^k} for a point of order 2^k as in [10]. We say that the curve has minimal two-torsion when $k = 1$ in $E(F_{2^n}) = G \times E[2^k]$, where G is a group of odd order. And we have

$$T_4, [3]T_4 \in E(F_{2^n}) \Leftrightarrow \exists \lambda \in F_{2^n} : \lambda^2 + \lambda = a_2.$$

Let F denote the linear operator $F(\lambda) = \lambda^2 + \lambda$ with domain F_{2^n}, a result is

$$E \text{ has minimal two-torsion } \Leftrightarrow a_2 \overline{\in} Im(F)$$

Theorem 1. *The binary Edwards curves have no minimal two-torsion in the case of $d_1 = d_2$.*

Proof. When $d_1 = d_2$, the binary Edwards curves are

$$\begin{aligned} E_{B,d_1,d_2}\colon d_1(x+y) + d_1(x^2+y^2) &= xy + xy(x+y) + x^2y^2 \\ \Leftrightarrow d_1(x + x^2 + y + y^2) &= (x + x^2)(y + y^2) \end{aligned}$$

And it is a birational equivalence from E_{B,d_1,d_2} to the elliptic curves [6]:

$$E\colon v^2 + uv = u^3 + (d_1^2 + d_1)u^2 + d_1^8.$$

Therefore, if we want to get a binary Edwards curve with $d_1 = d_2$, we have to let $a_2 = d_1^2 + d_1$. It is obviously from above that there always exists $\lambda = d_1 \in F_{2^n} : a_2 = d_1^2 + d_1 = \lambda^2 + \lambda$, so that $a_2 \in Im(F)$, E has four-torsion, i.e. $E(F_{2^n}) = G \times E[2^k]$ where $k \geq 2$. And so E has no minimal two-torsion, hence E_{B,d_1,d_2} has no minimal two-torsion. □

4.1 Using the algorithm of [12] to half in case of $d_1 = d_2$

King and Rubin [12] improved the algorithm of Knudsen [10] to half a point where the elliptic curve has a cofactor of 4, i.e., minimal four-torsion. When $d_1 = d_2$, the binary Edwards curves are the elliptic curves with cofactor of 4.

Now, let $Q = (x_4, y_4)$, $P = (x_2, y_2)$, $Q = 2P$, given Q, we explain how to compute P. Let $X_2 = x_2 + y_2$, $X_4 = x_4 + y_4$ and $T = X_2^2$, we can compute T using (2) as

$$T^2 + T = d_1 + \frac{d_1}{X_4 + 1}$$

Where $X_4 + 1 \neq 0$, otherwise we can see from function (2) $d_1 = 0$, and this is contradiction with the definition of binary Edwards curves.

We note that X_2 is equal to either $\frac{1}{2}X_4$ or $\frac{1}{2}X_4 + T_2$. Unfortunately, we cannot distinguish between these two values. The remaining algorithm in this case is the same as the algorithm in [12]. After getting the correct X_2, we can use a half-trace and lemma 2 to reveal the answer from the function (2):

$$x_2^4 + x_2^2 = x_4(d_1 + X_2^2 + X_2^4) + d_1X_2^2.$$

5 Performance Analysis and Application in Scalar Multiplication

The halving algorithm on binary Edwards curves in case of $d_1 = d_2$ has to half twice to distinguish the right answer. It is too expensive to be interested. Hence, we only analyze the halving algorithm in case of $d_1 \neq d_2$.

If we have a normal basis representation, we can compute the square of a field element simply by shifting the representing vector. Computing a square root works the same way but shifting to the opposite direction. And the traces and half-traces can also be computed very efficiently, because they are sums of powers of squares.

We can see that in our halving algorithm we have to figure out the right point $P = (x, y)$, and it takes $1I+3M+2S+1H+1T+1SR$, here we neglect the multiplications by the curves parameters. However, if we apply the half-and-add structure in ω-coordinate differential addition in scalar multiplication, we can just recover the x and y in the final phase.

Note : Our purpose is to propose a point halving algorithm on binary Edwards curves. Heretofore, it is more effective to use the point halving algorithm on Weierstrass curves than on Edwards ones in characteristic 2. Hence, another way to employ the point halving algorithm is to turn the binary Edwards curves into Weierstrass forms using the birational equivalence, and then apply the point halving algorithm on Weierstrass curves.

5.1 Application in scalar multiplication

Using the idea differential addition in [6], we can simplify the halving algorithm in scalar multiplication. We can replace the double-and-add structure into half-and-add.

Let $\omega_i = X_i$. We only need to compute the halving algorithm from step 1 to step 7, and at step 7 we can get the correct ω_i. Using half-and-add structure in differential addition, we can compute any scalar multiplication mP. And finally, let $A = \omega_1 + \omega_2$, $B = \omega_1\omega_2$ then we get

$$x_2^2 + x_2 = \frac{\omega_3\left(d_1 + B(A+1) + \frac{d_2}{d_1}B^2\right) + d_1 A + (y_1^2 + y_1)(\omega_2^2 + \omega_2)}{\omega_1^2 + \omega_1}.$$

This formula comes from $(x_1, y_1) = (x_3, y_3) - (x_2, y_2)$, i.e., the sum of two point $(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$, where $\omega_1 = x_1 + y_1, \omega_2 = x_2 + y_2$ and $\omega_3 = x_3 + y_3$:

$$x_3 = \frac{d_1(x_1 + x_2) + d_2(x_1 + y_1)(x_2 + y_2) + (x_1 + x_1^2)(x_2(y_1 + y_2 + 1) + y_1 y_2)}{d_1 + (x_1 + x_1^2)(x_2 + y_2)},$$

$$y_3 = \frac{d_1(y_1 + y_2) + d_2(x_1 + y_1)(x_2 + y_2) + (y_1 + y_1^2)(y_2(x_1 + x_2 + 1) + x_1 x_2)}{d_1 + (y_1 + y_1^2)(x_2 + y_2)}.$$

Given $(x_1, y_1), (x_3, y_3)$, there always has a solution (x_2, y_2) of the formula except the easily recognizable case [6], $\omega_1^2+\omega_1 = 0$. The formula produces $x_2^2+x_2$; a half-trace computation reveals either x_2 or x_2+1, and therefore either $mP = (x_2, y_2)$ or $mP = (x_2 + 1, y_2 + 1)$.

At the end of the computation, there are two answers, and we cannot distinguish which is the correct one. If we assume the group order of E_{B,d_1,d_2} is $2r$, where r is odd. $E_{B,d_1,d_2}(F_{2^d}) = G \times E[2]$ where d is odd and P in G, then there is only one correct answer mP which can be distinguished by lemma 2.

6 Conclusion

In this paper, we analyzed the point halving algorithm on binary Edwards curves. We presented a halving algorithm on binary Edwards curves in case of $d_1 \neq d_2$. Then we gave a theorem to prove that the binary Edwards curves have no minimal two-torsion. We also described how to use the improved version of the elliptic curve for curves with cofactor of 4 to achieve the halving formula in this case. Finally, we applied our halving algorithm in scalar multiplication using the idea of Montgomery ladder. This is the first halving algorithm on binary Edwards curves which tries to speed up the scalar multiplication on Edwards curves. However, our halving algorithm is slower than the doubling one. Even if we apply our half-and-add structure in scalar multiplication with ω-coordinate, our algorithm is still slightly slower than the double-and-add one. How to improve the halving algorithm on binary Edwards curves is our future work.

References

1. R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, F. Vercauteren, “Handbook of elliptic and hyperelliptic curve cryptography”, CRC Press, Boca Raton, 2005.
2. R. Avanzi, M. Ciet, and F. Sica, “Faster scalar multiplication on Koblitz curves combining point halving with the Frobenius endomorphism”, PKC 2004, LNCS 2947, pp. 28-40, 2004.
3. P. Birkner, “Efficient divisor class halving on genus two curves”, SAC 2006, LNCS 4356, pp. 317-326, 2007.
4. D.J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, “Twisted Edwards curves”, AFRICACRYPT 2008, LNCS 5023, pp. 389-405, 2008.
5. D.J. Bernstein, T. Lange, “Faster addition and doubling on elliptic curves”, ASIACRYPT 2007, LNCS, vol. 4833, pp. 29-50, 2007.
6. D.J. Bernstein, T. Lange, and R.R. Farashahi, “Binary Edwards curves”, CHES 2008, LNCS 5154, pp. 244-265, 2008.
7. P. Birkner and N. Thériault, “Faster halvings in genus 2”, SAC 2008, LNCS 5381, pp. 1-17, 2009.
8. H.M. Edwards, “A normal form for elliptic curves”, Bulletin of the American Mathematical Society 44, 393-422, 2007.
9. K. Fong, D. Hankerson, J. López, and A. Menezes, “Field inversion and point halving revisited”, IEEE TRANSACTIONS ON COMPUTERS, vol. 53, no. 8, August 2004.

10. E. Knudsen, "Elliptic scalar multiplication using point halving", ASIACRYPT 1999, LNCS, vol. 1716, pp. 135-149, 1999.
11. I. Kitamura, M. Katagi, T. Takagi, "A complete divisor class halving algorithm for hyperelliptic curve cryptosystems of genus two", ACISP 2005, LNCS, vol. 3574, pp. 146-157, 2005.
12. B. King and B. Rubin, "Improvements to the point halving algorithm", ACISP 2004, LNCS 3108, pp. 262-276, 2004.
13. R. Schroeppel, "Elliptic curve point halving wins big", 2nd Midwest Arithmetical Geometry in Cryptography Workshop, Urbana, November 2000.