

Efficient Montgomery-like formulas for general Huff's and Huff's elliptic curves and their applications to the isogeny-based cryptography

Robert Dryło¹, Tomasz Kijko¹, and Michał Wroński¹

Military University of Technology, Kaliskiego Str. 2, Warsaw, Poland
{robert.drylo,tomasz.kijko,michal.wronski}@wat.edu.pl

Abstract. Montgomery's formulas for doubling and differential addition in x -coordinates for elliptic curves $By^2 = x^3 + Ax^2 + x$ are among the most efficient formulas for point multiplication after compression. In general, if E is an elliptic curve over a field K , then a degree 2 function $f : E \rightarrow K$ such that $f(P) = f(-P)$ for $P \in E$ can be used as a compression and there exist analogous formulas for doubling and differential addition of values f which can be used in the Montgomery ladder algorithm to compute multiplication $[n]f(P) = f([n]P)$ for $n \in \mathbb{N}$. In this paper we give formulas for doubling and differential addition of the same or similar efficiency as Montgomery's for Huff's and general Huff's curves of odd characteristic and degree 2 compression, which seems to be new for these models of elliptic curves. Additionally, we give formulas for point recovery after compression. We also found efficient formulas for general odd-isogeny computation on Huff's curves and we showed how to apply obtained formulas, especially, to the isogeny based cryptography. Moreover, it was showed how to apply obtained by us formulas using compression to the ECM algorithm. In appendix, we present examples of convenient cryptographic Huff's curves, where presented compression techniques can be used.

Keywords: general Huff's curves · Huff's curves · compression on elliptic curves · isogeny-based cryptography · ECM method

1 Introduction

The x -coordinate arithmetic is an approach of point compression on elliptic curves, which is widely used, for example for reduction of keys sizes and protection of implementations against side-channel attacks. The clear presentations of results on x -coordinate are presented for example in [2], [5]. We shortly recall the description of x -coordinate arithmetic from [5]. Let E be an elliptic curve over a field K , with group law $+$, identity point O and negation map $-$. Scalar multiplication endomorphism on E is therefore defined by

$$[k] : P \rightarrow \underbrace{P + \dots + P}_{k \text{ times}} \quad (1)$$

for every $k \in \mathbb{Z}$ (with $[-k]P = [k](-P)$). In this case, the negation map is an automorphism of E and the quotient of E by $\langle - \rangle$ is the 2-to-1 mapping

$$f : E \rightarrow \mathbb{P}^1 \cong E/\langle - \rangle \quad (2)$$

such that $f(P) = f(Q)$ iff $P = Q$ or $P = -Q$. If E is defined by short Weierstrass equation $y^2 = x^3 + Ax + B$, then f is just $P \rightarrow x(P)$, where $x(P)$ denotes the x -coordinate of point P . In other models (or coordinate systems), f may be more complicated. \mathbb{P}^1 is called the x -line of E . \mathbb{P}^1 does not inherit any group structure from E . But since negation map commutes with $[k]$, we get an induced pseudomultiplication

$$f(P) \rightarrow f([k]P) \quad (3)$$

on \mathbb{P}^1 for every $k \in \mathbb{Z}$.

Function f is often called the compression function. There exist formulas for doubling $D(x) \in K(x)$ and differential addition $A(x, y, z) \in K(x, y, z)$ such that $D(f(P)) = f([2]P)$ and $f(P + Q) = A(f(P), f(Q), f(Q - P))$ for generic $P, Q \in E$, which can be used in the Montgomery ladder algorithm to compute $[n]f(P)$ for $n \in \mathbb{N}$. In practice one usually obtains a function A from the functions $A_1 \in K(x, y)$ or $A_2 \in K(x, y)$ such that $f(P + Q)f(Q - P) = A_1(f(P), f(Q))$ and $f(P + Q) + f(Q - P) = A_2(f(P), f(Q))$ for generic $P, Q \in E$ (e.g., $A(x, y, z) = A_1(x, y)/z$).

Since introducing other models of elliptic curves with efficient and useful for cryptographic application group law a natural task was to extend formulas for doubling and differential addition to these curves. This was done for, e.g., curves given by the equation of Weierstrass [3], Edwards [8, 13], Hessian form [9], Jacobi quartic [4], [6] and twisted Hessian form [6]. We did not find any differential addition and doubling formulas nor application of such formulas for isogeny based cryptography for Huff's curves over fields of characteristic greater than 2, instead of compression functions, differential addition and doubling formulas in [6] for Huff's curves, but in this paper we give more efficient formulas.

However, formulas similar to the Montgomery formulas, were presented for differential addition on extended Huff's model (Appendix B of [16]). Extended Huff's curve is given by equation:

$$EH_{a,c,d} : y(1 + ax^2) = cx(1 + dy^2). \quad (4)$$

The compression function $f(P), P \in EH_{a,c,d}(K)$ is given in this case by $f(P) = x$, and differential addition formula is given by

$$f(P + Q)f(P - Q) = \frac{f(P)^2 - f(Q)^2}{1 - a^2f(P)^2f(Q)^2} \quad (5)$$

Moreover, there are also formulas for differential addition and doubling using point compression on binary Huff's curves [7].

In this paper we give as efficient formulas as Montgomery's for Huff's curves $ax(y^2 - 1) = by(x^2 - 1)$ for a degree 2 compression function (Theorem 1) and

general Huff's curves $\bar{x}(\bar{a}\bar{y}^2 - 1) = \bar{b}\bar{y}(\bar{x}^2 - 1)$ for a degree 2 compression function (Theorem 3).

If $f : E \rightarrow K$ is degree 2 compression function and $f(P) = f(-P)$ for $P \in E$, then one can recover a point after compression, i.e., given $P \in E$ and $n \in \mathbb{N}$ the Montgomery ladder algorithm returns $[n]f(P)$ and $[n+1]f(P)$, which can be used to recover the point $[n]P$ on E , because there exists a rational map B such that $Q = B(P, f(Q), f(P+Q))$ for generic $P, Q \in E$. Formulas for point recovery are known for curves given by the equation of Weierstrass [3, 18], Edwards [13], generalized and twisted Hessian form, Huff's and Jacobi quartic [6]. In Theorem 1 and Theorem 3 we also give formula for point recovery on Huff's and general Huff's curves, respectively.

Thus Huff's and general Huff's curves can be competitive with other models, since they have efficient and unified addition law [12, 19, 17] and as efficient arithmetic after compression as Montgomery's curves. Moreover, we present application of computed formulas for obtaining efficient formulas for computation of general odd-degree isogeny and applications to the ECM method.

2 Montgomery-like formulas for Huff's and general Huff's curves

In this section we give formulas for doubling, differential addition and point recovery for both Huff's and general Huff's curves using a degree 2 compression.

From now we assume that K is a field with $\text{char}(K) \geq 5$.

2.1 Huff's curves

Joye, Tibouchi and Vergnaud in [12] described the group law and pairing computation on Huff's elliptic curves, which are useful in cryptography. Huff's curve over K is given by the equation

$$H_{a,b} : ax(y^2 - 1) = by(x^2 - 1), \quad (6)$$

where $a^2 \neq b^2$ and $a, b \neq 0$. The point $O = (0, 0)$ is the neutral element, and the opposite point $-(x, y) = (-x, -y)$. The addition $(x_P, y_P) + (x_Q, y_Q) = (x_R, y_R)$ is given by

$$\begin{cases} x_R = \frac{(x_P + x_Q)(1 + y_P y_Q)}{(1 + x_P x_Q)(1 - y_P y_Q)}, \\ y_R = \frac{(y_P + y_Q)(1 + x_P x_Q)}{(1 - x_P x_Q)(1 + y_P y_Q)}. \end{cases} \quad (7)$$

Let's note that every Huff's curve has three points at infinity (which are also points of order 2) and therefore these points cannot be represented in affine coordinates. It means that compression function $f(P) = xy$, for $P = (x, y)$, cannot be applied to these three points. We have obtained the following formulas for this compression function.

Theorem 1. *On Huff's curves (6) with a degree 2 compression function $f(x, y) = xy$ we have the following formulas for doubling and differential addition*

$$f([2]P) = \frac{4f(P)(f(P)^2 + (\frac{b}{a} + \frac{a}{b})f(P) + 1)}{(f(P)^2 - 1)^2}, \quad (8)$$

$$f(P + Q)f(P - Q) = \left(\frac{f(P) - f(Q)}{f(P)f(Q) - 1} \right)^2. \quad (9)$$

Theorem 2. *There holds the following point recovery formula, for $Q = (x_Q, y_Q)$*

$$x_Q = \frac{f(P)f(Q)f(P+Q) + 2\frac{a}{b}f(P)f(Q) + f(P) + f(Q) - f(P+Q)}{x_P(f(P)f(Q)f(P+Q) - f(P) - f(Q) - f(P+Q) - 2\frac{a}{b})}, \quad (10)$$

and $y_Q = f(Q)/x_Q$.

Let $R \in H_{a,b}$ and let $(X_{f(R)} : Z_{f(R)})$ be projective representation of $f(R)$. Writing as Montgomery's formulas in projective representation [14] we have for doubling

$$\begin{cases} 4X_{f(P)}Z_{f(P)} = (X_{f(P)} + Z_{f(P)})^2 - (X_{f(P)} - Z_{f(P)})^2, \\ X_{f([2]P)} = 4X_{f(P)}Z_{f(P)}((X_{f(P)} - Z_{f(P)})^2 + AX_{f(P)}Z_{f(P)}), \\ Z_{f([2]P)} = (X_{f(P)} + Z_{f(P)})^2(X_{f(P)} - Z_{f(P)})^2, \end{cases} \quad (11)$$

where $A = \frac{a}{b} + \frac{b}{a} + 2$. The cost of these formulas is equal to $3M + 2S + c$, where M, S, c are the costs of multiplication, squaring and multiplication by a constant in K , respectively. The cost c can be made small if coefficients a, b are chosen such that A is small. Moreover, if $A_4 = \frac{(a+b)^2}{4ab}$, then we can rewrite $X_{f([2]P)}$ in equation (11) as

$$X_{f([2]P)} = 4X_{f(P)}Z_{f(P)}((X_{f(P)} - Z_{f(P)})^2 + A_4X_{f(P)}Z_{f(P)}) \quad (12)$$

and doubling requires $2M + 2S + c$. Similarly, the differential addition in projective representation is given by

$$\begin{cases} X_{f(P+Q)} = Z_{f(P-Q)}((X_{f(P)} - Z_{f(P)})(X_{f(Q)} + Z_{f(Q)}) - (X_{f(P)} + Z_{f(P)})(X_{f(Q)} - Z_{f(Q)}))^2, \\ Z_{f(P+Q)} = X_{f(P-Q)}((X_{f(P)} - Z_{f(P)})(X_{f(Q)} + Z_{f(Q)}) + (X_{f(P)} + Z_{f(P)})(X_{f(Q)} - Z_{f(Q)}))^2, \end{cases} \quad (13)$$

and has cost $4M + 2S$. Formula for point recovery (10) takes $4M + 2I$, where I is the cost of inversion in K .

2.2 Verification

In this subsection it will be showed how to verify that formulas (8), (9) and (10) are proper.

At first, the formula (7) for point addition will be used, to compute $R = (x_R, y_R) = P + Q$ and $S = (x_S, y_S) = P - Q$, where $P = (x_P, y_P), Q = (x_Q, y_Q)$.

Then

$$\begin{cases} x_R = \frac{(x_P+x_Q)(1+y_P y_Q)}{(1+x_P x_Q)(1-y_P y_Q)}, \\ y_R = \frac{(y_P+y_Q)(1+x_P x_Q)}{(1-x_P x_Q)(1+y_P y_Q)}, \\ x_S = \frac{(x_P-x_Q)(1-y_P y_Q)}{(1-x_P x_Q)(1+y_P y_Q)}, \\ y_S = \frac{(y_P-y_Q)(1-x_P x_Q)}{(1+x_P x_Q)(1-y_P y_Q)}. \end{cases} \quad (14)$$

Let's note, that $f(P+Q)f(P-Q) = f(R)f(S) = x_R y_R x_S y_S = \frac{L}{M}$, where:

$$L = x_Q^2 y_Q^2 - x_P^2 y_Q^2 - x_Q^2 y_P^2 + x_P^2 y_P^2 = f(Q)^2 - (x_P^2 y_Q^2 + x_Q^2 y_P^2) + f(P)^2 \quad (15)$$

and

$$M = x_P^2 x_Q^2 y_P^2 y_Q^2 - y_P^2 y_Q^2 - x_P^2 x_Q^2 + 1 = f(P)^2 f(Q)^2 - (y_P^2 y_Q^2 + x_P^2 x_Q^2) + 1. \quad (16)$$

Now, we will prove the following lemma.

Lemma 1. *Having compression $f(P)$ of point $P \in H_{a,b}(K)$, where $f(P) = x_P y_P$, one can obtain exact value of x_P^2 and y_P^2 , both given by rational functions of $f(P)$.*

Proof. At first, having $f(P)$ for the point $P \in H_{a,b}$, where $f(P) \neq 0$ we will find x_P^2 as the rational function of $f(P)$. Because $x_P y_P = f(P)$, then $y_P = \frac{f(P)}{x_P}$. Substituting y_P in the Huff's curve $H_{a,b}$ equation one obtains

$$a x_P \left(\frac{f(P)^2}{x_P^2} - 1 \right) = b \frac{f(P)}{x_P} (x_P^2 - 1) \quad (17)$$

and after some transformations, one obtains

$$x_P^2 (b f(P) + a) = a f(P)^2 + b f(P) \quad (18)$$

and finally

$$x_P^2 = \frac{a f(P)^2 + b f(P)}{b f(P) + a} = \frac{f(P)(a f(P) + b)}{b f(P) + a} \quad (19)$$

Then of course also holds

$$y_P^2 = \frac{f(P)^2}{x_P^2} = \frac{f(P)^2 (b f(P) + a)}{f(P)(a f(P) + b)} = \frac{f(P)(b f(P) + a)}{a f(P) + b}. \quad (20)$$

Using Lemma 1 and equations (19) and (20), one can find, that $f(P+Q)f(P-Q) = \frac{L}{M}$ is equal to:

$$f(P+Q)f(P-Q) = \frac{f(Q)^2 - 2f(P)f(Q) + f(P)^2}{f(P)^2 f(Q)^2 - 2f(P)f(Q) + 1}, \quad (21)$$

which is equal to the formula (9).

Similarly, one can verify formula (8). Again, formula (7) should be used, to obtain $R = [2]P = (x_R, y_R)$, where

$$\begin{cases} x_R = \frac{2x_P(y_P^2+1)}{(x_P^2+1)(1-y_P^2)}, \\ y_R = 2(x_P^2+1)y_P(1-x_P^2)(y_P^2+1). \end{cases} \quad (22)$$

and then $f([2]P) = f(R) = \frac{L}{M}$, where

$$L = 4x_P y_P = 4f(P)$$

and

$$M = x_P^2 y_P^2 - y_P^2 - x_P^2 + 1 = f(P)^2 - (x_P^2 + y_P^2) + 1.$$

Finally, one can again use formulas (19) and (20) to obtain

$$f([2]P) = \frac{L}{M} = \frac{4f(P)(f(P)^2 + (\frac{b}{a} + \frac{a}{b})f(P) + 1)}{(f(P)^2 - 1)^2}, \quad (23)$$

which is equal to the formula (8).

Unfortunately, we were not able to verify the point recovery formula, (10) using elementary methods. This formula was found using the method from [6] implemented in Magma. In the Appendix A we give a Magma code for checking correctness of this formula using Gröbner basis.

2.3 General Huff's curves

In [19] Wu and Feng introduced general Huff's curves. General Huff's curves are given by the equation

$$G_{\bar{a}, \bar{b}} : \bar{x}(\bar{a}\bar{y}^2 - 1) = \bar{y}(\bar{b}\bar{x}^2 - 1) \quad (24)$$

where $\bar{a} \neq \bar{b}$ and $\bar{a}, \bar{b} \neq 0$. Similarly as for Huff's curve the point $O = (0, 0)$ is the neutral element, and the opposite point $-(\bar{x}, \bar{y}) = (-\bar{x}, -\bar{y})$. The addition $(\bar{x}_P, \bar{y}_P) + (\bar{x}_Q, \bar{y}_Q) = (\bar{x}_R, \bar{y}_R)$ is given by

$$\begin{cases} \bar{x}_R = \frac{(\bar{x}_P + \bar{x}_Q)(\bar{a}\bar{y}_P\bar{y}_Q + 1)}{(\bar{b}\bar{x}_P\bar{x}_Q + 1)(\bar{a}\bar{y}_P\bar{y}_Q - 1)}, \\ \bar{y}_R = \frac{(\bar{y}_P + \bar{y}_Q)(\bar{b}\bar{x}_P\bar{x}_Q + 1)}{(\bar{b}\bar{x}_P\bar{x}_Q - 1)(\bar{a}\bar{y}_P\bar{y}_Q + 1)}. \end{cases} \quad (25)$$

Lemma 2. *Every Huff's curve over a field K given by the equation (6) is also a general Huff's curve.*

Proof. By the substitutions:

$$\bar{x} = ax, \quad \bar{y} = by, \quad \bar{a} = \frac{1}{b^2} \text{ and } \bar{b} = \frac{1}{a^2} \quad (26)$$

we can transform equation (6) into the following general Huff's curve equation

$$G_{\bar{a}, \bar{b}} : \bar{x}(\bar{a}\bar{y}^2 - 1) = \bar{y}(\bar{b}\bar{x}^2 - 1). \quad (27)$$

If \bar{a} and \bar{b} are squares in K we can transform the general Huff's curve with equation (27) into the Huff's curve (6) by substitutions

$$x = \bar{x}\sqrt{\bar{b}}, \quad y = \bar{y}\sqrt{\bar{a}}, \quad a = \frac{1}{\sqrt{\bar{b}}} \text{ and } b = \frac{1}{\sqrt{\bar{a}}}. \quad (28)$$

Theorem 3. On general Huff's curves (24) with a degree 2 compression function $f(\bar{x}, \bar{y}) = \bar{x}\bar{y}$ we have the following formulas for doubling and differential addition

$$f([2]\bar{P}) = \frac{4f(\bar{P})(\bar{a}\bar{b}f(\bar{P})^2 + (\bar{a} + \bar{b})f(\bar{P}) + 1)}{(\bar{a}\bar{b}f(\bar{P})^2 - 1)^2}, \quad (29)$$

$$f(\bar{P} + \bar{Q})f(\bar{P} - \bar{Q}) = \left(\frac{f(\bar{P}) - f(\bar{Q})}{\bar{a}\bar{b}f(\bar{P})f(\bar{Q}) - 1} \right)^2. \quad (30)$$

The point recovery formula is as follows

$$\bar{x}_{\bar{Q}} = \frac{\bar{x}_P(\bar{a}\bar{b}f(\bar{P})f(\bar{Q})f(\bar{P} + \bar{Q}) - f(\bar{P}) + 2\bar{a}f(\bar{Q})f(\bar{P} + \bar{Q}) + f(\bar{Q}) + f(\bar{P} + \bar{Q}))}{\bar{a}\bar{b}f(\bar{P})f(\bar{Q})f(\bar{P} + \bar{Q}) + 2\bar{a}f(\bar{P})f(\bar{P} + \bar{Q}) + f(\bar{P}) - f(\bar{Q}) + f(\bar{P} + \bar{Q})} \quad (31)$$

and $\bar{y}_{\bar{Q}} = f(\bar{Q})/\bar{x}_{\bar{Q}}$.

Proof. Formula (29) can be mechanically obtained from (8) by substitutions (26). Similarly we can derive the doubling formula (30) from (9) and the point recovery formula (31) from (10).

3 Applications to the isogeny-based cryptography and ECM method

In this section we present the applications of formulas obtained in the previous sections.

3.1 General Huff's isogenies computation using compression techniques

In [15] Moody and Shumow presented formulas for computation of general Huff's curves isogenies. Because points of order 2 cannot be presented by compression function f , we present formulas for computation of odd degree isogenies using point compression.

Let $\bar{F} = \{(0, 0), (\bar{\alpha}_i, \bar{\beta}_i), (-\bar{\alpha}_i, -\bar{\beta}_i) : i = 1 \dots s\}$, where $(-\bar{\alpha}_i, -\bar{\beta}_i) = (-\bar{\alpha}_i, -\bar{\beta}_i)$, is the kernel of an isogeny $\bar{\psi}$ of degree ℓ , where $\ell = 2s + 1$. Let $\bar{A} = \prod_{i=1}^s \bar{\alpha}_i$ and $\bar{B} = \prod_{i=1}^s \bar{\beta}_i$.

Theorem 4. ([15], Theorem 5.) Define

$$\bar{\psi}(\bar{P}) = \left(\bar{x}_P \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{-\bar{x}_{\bar{P} + \bar{Q}}}{\bar{x}_{\bar{Q}}}, \bar{y}_P \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{-\bar{y}_{\bar{P} + \bar{Q}}}{\bar{y}_{\bar{Q}}} \right). \quad (32)$$

Then $\bar{\psi}$ is an ℓ -isogeny with kernel \bar{F} from the curve $G_{\bar{a}, \bar{b}}$ to the curve $G_{\bar{a}', \bar{b}'}$, where $\bar{a}' = \bar{a}^\ell \bar{B}^4$ and $\bar{b}' = \bar{b}^\ell \bar{A}^4$.

It is possible to compute isogeny $f(\bar{\psi})$ using point compression.

Let $\bar{R} \in G_{\bar{a}, \bar{b}}$ and let $(X_{f(\bar{R})} : Z_{f(\bar{R})})$ be projective representation of $f(\bar{R})$, where \bar{R} is the point defining kernel \bar{F} of the isogeny $\bar{\psi}$. Let $Ord(\bar{R})$ be the odd number. Let's note that $f(\bar{\psi}(\bar{P}))$ is given by

$$f(\bar{\psi}(\bar{P})) = \left(\bar{x}_{\bar{P}} \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{-\bar{x}_{\bar{P}+\bar{Q}}}{\bar{x}_{\bar{Q}}} \cdot \bar{y}_{\bar{P}} \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{-\bar{y}_{\bar{P}+\bar{Q}}}{\bar{y}_{\bar{Q}}} \right), \quad (33)$$

which is equal to

$$f(\bar{\psi}(\bar{P})) = \left(\bar{x}_{\bar{P}} \bar{y}_{\bar{P}} \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{\bar{x}_{\bar{P}+\bar{Q}} \bar{y}_{\bar{P}+\bar{Q}}}{\bar{x}_{\bar{Q}} \bar{y}_{\bar{Q}}} \right) = \left(f(\bar{P}) \prod_{\bar{Q} \in \bar{F}^+} \frac{f(\bar{P}+\bar{Q})f(\bar{P}-\bar{Q})}{f(\bar{Q})^2} \right), \quad (34)$$

where \bar{F}^+ is the set $\{(\bar{\alpha}_i, \bar{\beta}_i) : i = 1 \dots s\}$. Having generator \bar{R} of the kernel of the isogeny $\bar{\psi}$, given by projective compression $(\bar{X}_{f(\bar{R})} : \bar{Z}_{f(\bar{R})})$, it is easy to obtain others elements of the \bar{F}^+ , using for example ladder method. Let \bar{J} be the set of compressions in projective representation of \bar{F}^+ , so $\bar{J} = \{(\bar{X}_{f(\bar{P}_i)} : \bar{Z}_{f(\bar{P}_i)}) : i = 1 \dots s\}$. The value of $f(\bar{\psi})$ using point compression may be given by

$$f(\bar{\psi}(\bar{P})) = \left(\frac{\bar{X}_{f(\bar{P})}}{\bar{Z}_{f(\bar{P})}} \prod_{i=1}^s \frac{\bar{X}_{f(\bar{P}+\bar{Q}_i)} \bar{X}_{f(\bar{P}-\bar{Q}_i)} \bar{Z}_{f(\bar{Q}_i)}^2}{\bar{Z}_{f(\bar{P}+\bar{Q}_i)} \bar{Z}_{f(\bar{P}-\bar{Q}_i)} \bar{X}_{f(\bar{Q}_i)}^2} \right). \quad (35)$$

Having compression $f(\bar{P})$ of point \bar{P} , given in projective compression representation by $(\bar{X}_{f(\bar{P})} : \bar{Z}_{f(\bar{P})})$ and the set \bar{J} one can compute $\frac{\bar{X}_{f(\bar{P}+\bar{Q})} \bar{X}_{f(\bar{P}-\bar{Q})}}{\bar{Z}_{f(\bar{P}+\bar{Q})} \bar{Z}_{f(\bar{P}-\bar{Q})}}$ using identities

$$\begin{cases} \bar{X}_{f(\bar{P}+\bar{Q})} \bar{X}_{f(\bar{P}-\bar{Q})} = \left((\bar{X}_{f(\bar{P})} - \bar{Z}_{f(\bar{P})})(\bar{X}_{f(\bar{Q})} + \bar{Z}_{f(\bar{Q})}) - (\bar{X}_{f(\bar{P})} + \bar{Z}_{f(\bar{P})})(\bar{X}_{f(\bar{Q})} - \bar{Z}_{f(\bar{Q})}) \right)^2, \\ \bar{Z}_{f(\bar{P}+\bar{Q})} \bar{Z}_{f(\bar{P}-\bar{Q})} = \left((\bar{X}_{f(\bar{P})} - \bar{Z}_{f(\bar{P})})(\bar{X}_{f(\bar{Q})} + \bar{Z}_{f(\bar{Q})}) + (\bar{X}_{f(\bar{P})} + \bar{Z}_{f(\bar{P})})(\bar{X}_{f(\bar{Q})} - \bar{Z}_{f(\bar{Q})}) \right)^2 \end{cases} \quad (36)$$

and therefore one can obtain $f(\bar{\psi}(\bar{P}))$.

To find the coefficients \bar{a}' and \bar{b}' of general Huff's curve $G_{\bar{a}', \bar{b}'}$, one can use similar transformations as for formulas (19) and (20) and obtain

$$\begin{aligned} \bar{x}_{\bar{P}}^2 &= \frac{\bar{X}_{f(\bar{P})} (a \bar{X}_{f(\bar{P})} + \bar{Z}_{f(\bar{P})})}{\bar{Z}_{f(\bar{P})} (b \bar{X}_{f(\bar{P})} + \bar{Z}_{f(\bar{P})})}, \\ \bar{y}_{\bar{P}}^2 &= \frac{\bar{X}_{f(\bar{P})} (b \bar{X}_{f(\bar{P})} + \bar{Z}_{f(\bar{P})})}{\bar{Z}_{f(\bar{P})} (a \bar{X}_{f(\bar{P})} + \bar{Z}_{f(\bar{P})})}. \end{aligned} \quad (37)$$

Finally

$$\begin{aligned} \bar{a}' &= \bar{a}^\ell \bar{B}^4 = \bar{a}^\ell \prod_{i=1}^s \bar{y}_{\bar{P}_i}^4 = \bar{a}^\ell \prod_{i=1}^s \left(\frac{\bar{X}_{f(\bar{P}_i)} (b \bar{X}_{f(\bar{P}_i)} + \bar{Z}_{f(\bar{P}_i)})}{\bar{Z}_{f(\bar{P}_i)} (a \bar{X}_{f(\bar{P}_i)} + \bar{Z}_{f(\bar{P}_i)})} \right)^2 \\ \bar{b}' &= \bar{b}^\ell \bar{A}^4 = \bar{b}^\ell \prod_{i=1}^s \bar{x}_{\bar{P}_i}^4 = \bar{b}^\ell \prod_{i=1}^s \left(\frac{\bar{X}_{f(\bar{P}_i)} (a \bar{X}_{f(\bar{P}_i)} + \bar{Z}_{f(\bar{P}_i)})}{\bar{Z}_{f(\bar{P}_i)} (b \bar{X}_{f(\bar{P}_i)} + \bar{Z}_{f(\bar{P}_i)})} \right)^2. \end{aligned} \quad (38)$$

3.2 Huff's isogenies computation using compression techniques

In this subsection it will be showed how to obtain formulas for computation of isogeny on Huff's curves using Theorem 4 and sequence of isomorphisms and isogenies between Huff's and general Huff's curves.

Theorem 5. *Let $F = \{(0, 0), (\alpha_i, \beta_i), (-\alpha_i, -\beta_i) : i = 1 \dots s\}$, where $(-\alpha_i, -\beta_i) = (-\alpha_i, -\beta_i)$, be the kernel of an isogeny $\bar{\psi}$. Let $A = \prod_{i=1}^s \alpha_i$ and $B = \prod_{i=1}^s \beta_i$. Lets Define*

$$\psi(P) = \left(-x_P \prod_{Q \neq (0,0) \in F} x_{P+Q}, -y_P \prod_{Q \neq (0,0) \in F} y_{P+Q} \right). \quad (39)$$

Then ψ is a ℓ -isogeny with kernel F , from the curve $H_{a,b}$, to the curve $H_{a',b'}$, where $a' = \frac{a}{A^2} = \frac{a}{\prod_{i=1}^s x_{Q_i}^2}$ and $b' = \frac{b}{B^2} = \frac{b}{\prod_{i=1}^s y_{Q_i}^2}$

Proof. To prove the Theorem 5 we will use the following composition $\tau \circ \bar{\psi} \circ \xi$, where:

- ξ is an isomorphism from Huff's curve $H_{a,b}$ to general Huff's curve $G_{\bar{a},\bar{b}}$, where $\bar{a} = \frac{1}{b^2}, \bar{b} = \frac{1}{a^2}$ and where for $P = (x, y)$ the isomorphism ξ using lemma 2 has form $\bar{P} = \xi(P) = (ax, by) = (\bar{x}, \bar{y})$,
- $\bar{\psi}$ is an isogeny from general Huff curve $G_{\bar{a},\bar{b}}$ to general Huff curve $G_{\bar{a}',\bar{b}'}$, where the kernel $\bar{F} = \{(0, 0), \xi(\alpha_i, \beta_i), \xi(-\alpha_i, -\beta_i)\} = \{(0, 0), (\bar{\alpha}_i, \bar{\beta}_i), (-\bar{\alpha}_i, -\bar{\beta}_i)\}$ and for $\bar{P} = (\bar{x}, \bar{y})$ the isogeny $\bar{\psi}$ has form

$$\begin{aligned} \bar{P}' = \bar{\psi}(\bar{P}) &= \left(\bar{x}_{\bar{P}} \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{-\bar{x}_{\bar{P}+\bar{Q}}}{\bar{x}_{\bar{Q}}}, \bar{y}_{\bar{P}} \prod_{\bar{Q} \neq (0,0) \in \bar{F}} \frac{-\bar{y}_{\bar{P}+\bar{Q}}}{\bar{y}_{\bar{Q}}} \right) = \\ &= \left(ax_P \prod_{Q \neq (0,0) \in F} \frac{-ax_{P+Q}}{ax_Q}, by_P \prod_{Q \neq (0,0) \in F} \frac{-by_{P+Q}}{by_Q} \right) = \\ &= \left(ax_P \prod_{Q \neq (0,0) \in F} \frac{-x_{P+Q}}{x_Q}, by_P \prod_{Q \neq (0,0) \in F} \frac{-y_{P+Q}}{y_Q} \right) \end{aligned} \quad (40)$$

where

$$\begin{aligned} \bar{a}' &= \bar{a}^\ell \bar{B}^4 = \bar{a}^\ell \left(\prod_{i=1}^s \bar{\beta}_i \right)^4, \\ \bar{b}' &= \bar{b}^\ell \bar{A}^4 = \bar{b}^\ell \left(\prod_{i=1}^s \bar{\alpha}_i \right)^4 \end{aligned} \quad (41)$$

- τ is an isomorphism from general Huff curve $G_{\bar{a}',\bar{b}'}$ to the Huff curve $H_{a',b'}$, where

$$\begin{aligned} a' &= \frac{1}{\sqrt{\bar{b}'}} = \frac{1}{\sqrt{\frac{1}{\bar{a}'^\ell} \left(\prod_{i=1}^s ax_{Q_i} \right)^2}} = \frac{1}{\frac{\bar{a}'^{2s}}{\bar{a}'^\ell} \left(\prod_{i=1}^s x_{Q_i} \right)^2} = \frac{\bar{a}'^\ell}{\left(\prod_{i=1}^s x_{Q_i} \right)^2}, \\ b' &= \frac{1}{\sqrt{\bar{a}'}} = \frac{1}{\sqrt{\frac{1}{\bar{b}'^\ell} \left(\prod_{i=1}^s by_{Q_i} \right)^2}} = \frac{\bar{b}'^{2s}}{\bar{b}'^\ell} \frac{1}{\left(\prod_{i=1}^s y_{Q_i} \right)^2} = \frac{\bar{b}'^\ell}{\left(\prod_{i=1}^s y_{Q_i} \right)^2} \end{aligned} \quad (42)$$

and

$$\begin{aligned} P' = \tau(\bar{P}') &= \left(\frac{\bar{a}'^\ell}{a'} x_P \prod_{Q \neq (0,0) \in F} \frac{-x_{P+Q}}{x_Q}, \frac{\bar{b}'^\ell}{b'} y_P \prod_{Q \neq (0,0) \in F} \frac{-y_{P+Q}}{y_Q} \right) = \\ &= \left(x_P \left(\prod_{i=1}^s x_{Q_i} \right)^2 \prod_{Q \neq (0,0) \in F} \frac{-x_{P+Q}}{x_Q}, y_P \left(\prod_{i=1}^s y_{Q_i} \right)^2 \prod_{Q \neq (0,0) \in F} \frac{-y_{P+Q}}{y_Q} \right) = \\ &= \left(-x_P \prod_{Q \neq (0,0) \in F} x_{P+Q}, -y_P \prod_{Q \neq (0,0) \in F} y_{P+Q} \right) \end{aligned} \quad (43)$$

Corollary 1 *Let $R \in H_{a,b}$ and let $(X_{f(R)} : Z_{f(R)})$ be projective representation of $f(R)$, where R is the point defining the kernel F of the isogeny ψ . Let $\text{Ord}(R)$ be the odd number. Let's note that $f(\psi(P))$ is given by*

$$f(\psi(P)) = \left(x_P \prod_{Q \neq (0,0) \in F} x_{P+Q} \cdot y_P \prod_{Q \neq (0,0) \in F} y_{P+Q} \right), \quad (44)$$

which is equal to

$$f(\psi(P)) = \left(x_P y_P \prod_{Q \neq (0,0) \in F} x_{P+Q} y_{P+Q} \right) = \left(f(P) \prod_{Q \in F^+} f(P+Q) f(P-Q) \right), \quad (45)$$

where F^+ is the set $\{(\alpha_i, \beta_i) : i = 1, \dots, s\}$. Having generator R of the kernel of the isogeny $\bar{\psi}$, given by projective compression representation $(X_{f(R)} : Z_{f(R)})$, it is easy to obtain others elements of the F^+ , using for example ladder method. Let J be the set of projective representations of F^+ , so $J = \{(X_{f(Q_i)} : Z_{f(Q_i)}) : i = 1, \dots, s\}$. In projective representation $f(\bar{\psi})$ using point compression may be given by

$$f(\psi(P)) = \left(\frac{X_{f(P)}}{Z_{f(P)}} \prod_{i=1}^s \frac{X_{f(P+Q_i)} X_{f(P-Q_i)}}{Z_{f(P+Q_i)} Z_{f(P-Q_i)}} \right). \quad (46)$$

To find the coefficients a' and b' of Huff's curve $H_{a',b'}$, if $f(P) = \frac{X_{f(P)}}{Z_{f(P)}}$, one can use formula (47)

$$\begin{aligned} x_P^2 &= \frac{X_{f(P)}(aX_{f(P)} + bZ_{f(P)})}{Z_{f(P)}(bX_{f(P)} + aZ_{f(P)})}, \\ y_P^2 &= \frac{X_{f(P)}(bX_{f(P)} + aZ_{f(P)})}{Z_{f(P)}(aX_{f(P)} + bZ_{f(P)})}, \end{aligned} \quad (47)$$

and finally

$$\begin{aligned} a' &= \frac{a}{\left(\prod_{i=1}^s x_{Q_i}\right)^2} = \frac{a}{\prod_{i=1}^s \left(\frac{X_{f(Q_i)}(aX_{f(Q_i)} + bZ_{f(Q_i)})}{Z_{f(Q_i)}(bX_{f(Q_i)} + aZ_{f(Q_i)})} \right)} = \frac{a \prod_{i=1}^s Z_{f(Q_i)}(bX_{f(Q_i)} + aZ_{f(Q_i)})}{\prod_{i=1}^s X_{f(Q_i)}(aX_{f(Q_i)} + bZ_{f(Q_i)})}, \\ b' &= \frac{b}{\left(\prod_{i=1}^s x_{Q_i}\right)^2} = \frac{b}{\prod_{i=1}^s \left(\frac{X_{f(Q_i)}(bX_{f(Q_i)} + aZ_{f(Q_i)})}{Z_{f(Q_i)}(aX_{f(Q_i)} + bZ_{f(Q_i)})} \right)} = \frac{b \prod_{i=1}^s Z_{f(Q_i)}(aX_{f(Q_i)} + bZ_{f(Q_i)})}{\prod_{i=1}^s X_{f(Q_i)}(bX_{f(Q_i)} + aZ_{f(Q_i)})}. \end{aligned} \quad (48)$$

Obtained formulas may be used for example in the isogeny based cryptography, like in SIDH algorithm, and may be reasonable alternative for Montgomery curves arithmetic.

It is worth to note that Huff's curve arithmetic (including obtaining and evaluation of an isogeny of given degree) is more efficient than such arithmetic on general Huff's curves. Thus, the computational cost of presented formulas for Huff's curves using compression function $f(P) = x_P y_P$ is presented in Table 1.

3.3 ECM algorithm using Huff's and general Huff's curves

In this subsection we will show how to generate Huff's and general Huff's curves convenient to the use in ECM algorithm, where presented in this paper compression techniques may be used.

In [11] was proved the Theorem 6.

Operation	Computational cost
$f(P+Q)f(P-Q)$ (F)	$2M+2S$
Differential addition $f(P+Q)$	$4M+2S$
Doubling $f([2]P)$ (D)	$4M+2S$
Doubling ($\frac{(a+b)^2}{4ab}$ is constant)	$2M+2S$
3-isogenous curve	$10M$
5-isogenous curve the full kernel is not given	$26M+2S$
5-isogenous curve the full kernel is given	$22M$
ℓ -isogenous curve the full kernel is not given	$6(\ell-1)M-2M+[\ell \geq 5]D+[\ell \geq 7](\frac{\ell-5}{2}F)$
ℓ -isogenous curve the full kernel is given	$6(\ell-1)M-2M$
3-isogeny evaluation	$4M$
5-isogeny evaluation	$8M$
ℓ -isogeny evaluation (the full kernel is given)	$2(\ell-1)M$

Table 1. Computational costs of Huff's curve operations using compression function $f(x, y) = xy$, where costs of operations in field K are denoted as: M for multiplication, S for squaring. $[C]$ denotes logical operation and is equal to 1 if sentence C is true, and 0 otherwise.

Theorem 6. ([11], Theorem 4.10.) Let $K = \mathbb{Q}(\sqrt{-1}, \sqrt{t^4 - 6t^2 + 1})$ with $t \in \mathbb{Q}$ and $t \neq 0, \pm 1$ and let E be an elliptic curve defined by equation

$$E : \check{y}^2 + \check{x}\check{y} - \left(v^2 - \frac{1}{16}\right)\check{y} = \check{x}^3 - \left(v^2 - \frac{1}{16}\right)\check{x}^2, \quad (49)$$

where $v = \frac{t^4 - 6t^2 + 1}{4(t^2 + 1)^2}$ and $t \neq 0, \pm 1$. Then the torsion subgroup of E over K is equal to $\mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/8\mathbb{Z}$ for almost all t .

We will show how to find Huff's curve $H_{a,b}$ isomorphic to the curve E .

At first, the isomorphic short Weierstrass curve E_1 to the curve E is equal to

$$E_1 : \check{y}^2 = \check{x}^3 + (-432s^2 - 432s - 27)\check{x} + (-3456s^3 + 6480s^2 + 1296s + 54), \quad (50)$$

where $s = (v^2 - \frac{1}{16})$. Now it is necessary to find the x -coordinate of three points of order 2, which are the roots of $f(u) = u^3 + (-432s^2 - 432s - 27)u + (-3456s^3 + 6480s^2 + 1296s + 54)$. They are equal to

$$\begin{cases} r_0 = \frac{3t^8 - 12t^6 + 66t^4 - 12t^2 + 3}{t^8 + 4t^6 + 6t^4 + 4t^2 + 1}, \\ r_1 = -\frac{6t^8 - 24t^6 - 12t^4 - 24t^2 + 6}{t^8 + 4t^6 + 6t^4 + 4t^2 + 1}, \\ r_2 = \frac{3t^8 - 12t^6 - 78t^4 - 12t^2 + 3}{t^8 + 4t^6 + 6t^4 + 4t^2 + 1}. \end{cases} \quad (51)$$

Substituting

$$R_0 = 0, R_1 = r_1 - r_0, R_2 = r_2 - r_1,$$

one obtains isomorphic elliptic curve

$$E_2 : \hat{y}^2 = \hat{x}^3 - (R_1 + R_2)\hat{x}^2 + R_1R_2\hat{x}. \quad (52)$$

The roots R_0, R_1, R_2 are equal to:

$$\begin{cases} R_0 = 0, \\ R_1 = -\frac{9(t-1)^4(t+1)^4}{(t^2+1)^4} = -\left(\frac{3(t-1)^2(t+1)^2}{(t^2+1)^2}\right)^2, \\ R_2 = -\frac{144t^4}{(t^2+1)^4} = -\left(\frac{12t^2}{(t^2+1)^2}\right)^2. \end{cases} \quad (53)$$

Using isomorphism between Weierstrass and Huff's curve given in [12]

$$H_{a,b} : ax(y^2 - 1) = by(x^2 - 1) \cong E_2 : \hat{y}^2 = \hat{x}(\hat{x} + a^2)(\hat{x} + b^2) \quad (54)$$

and isomorphism between general Huff's and Weierstrass curve [19]

$$G_{\bar{a},\bar{b}} : \bar{x}(\bar{a}\bar{y}^2 - 1) = \bar{y}(\bar{b}\bar{x}^2 - 1) \cong E_3 : \hat{y}^2 = \hat{x}(\hat{x} + \bar{a})(\hat{x} + \bar{b}), \quad (55)$$

one can find the coefficients of the isomorphic Huff's curve whose are therefore equal to

$$a = \frac{3(t-1)^2(t+1)^2}{(t^2+1)^2}, \quad b = \frac{12t^2}{(t^2+1)^2}. \quad (56)$$

and the coefficients of the isomorphic general Huff's curve whose are therefore equal to

$$\bar{a} = \frac{3(t-1)^4(t+1)^4}{(t^2+1)^4}, \quad \bar{b} = \frac{12t^4}{(t^2+1)^4}. \quad (57)$$

4 Conclusion

In this paper were given formulas for doubling and differential addition of the same efficiency as Montgomery's for Huff's curves and degree 2 compression. For general Huff's curves and degree 2 compression, we obtained formulas for doubling and differential addition of similar efficiency as Montgomery's. Additionally, for both models of elliptic curves were also presented formulas for point recovery after point compression. Recently formulas as efficient as Montgomery's were given by Farashahi [8] for twisted Edwards curves, who used compression function $E \rightarrow K$ of degree 8.

The applications of obtained formulas for Huff's and general Huff's curves to the isogeny-based cryptography and ECM method have been showed. We hope that presented formulas will be practically used as a reasonable alternative to the Montgomery's curve arithmetic using x -line.

References

1. Bernstein, D., Birkner, P., Joye, M., Lange, T., Peters, Ch.: Twisted Edwards curves, In: *Progress in Cryptology-AFRICACRYPT 2008*, pp. 389-405. Springer (2008)
2. Bernstein, D.-J., Lange, T.: Montgomery curves and the Montgomery ladder. Report 293/2017, Cryptology ePrint Archive, (2017)
3. Brier, E., Joye, M.: Weierstraß elliptic curves and side-channel attacks. In D. Naccache and P. Paillier, editors, *Public Key Cryptography*, number 2274 in *Lecture Notes in Computer Science*, pages 183-194, Berlin, Heidelberg, Springer-Verlag (2002)
4. Castryck, W., Vercauteren, F.: Toric forms of elliptic curves and their arithmetic, *Journal of Symbolic Computation*, vol. 46, issue 8, pp. 943-966, (2011)
5. Costello, C., Smith, B.: Montgomery curves and their arithmetic: The case of large characteristic fields. Report 212/2017, Cryptology ePrint Archive, (2017)
6. Dryło, R., Kijko, T., Wroński, M.: Determining Formulas Related to Point Compression on Alternative Models of Elliptic Curves, *Fundamenta Informaticae*, vol. 169, no. 4, pp. 285-294 (2019)
7. Farashahi, R.-R., Hosseini, S.: Differential Addition on Binary Elliptic Curves, In: *Arithmetic of Finite Fields*, *Lecture Notes in Computer Science*, vol. 10064, pp. 21-35. Springer (2016)
8. Farashahi, R.-R., Hosseini, S.: Differential Addition on Twisted Edwards Curves, In: *ACISP 2017: Information Security and Privacy*, *Lecture Notes in Computer Science*, vol. 10343, pp. 366-378. Springer (2017)
9. Farashahi, R.-R., Joye, M.: Efficient arithmetic on Hessian curves. In *International Workshop on Public Key Cryptography* (pp. 243-260). Springer, Berlin, Heidelberg, (2010).
10. Hisil, H., Renes, J.: On Kummer Lines With Full Rational 2-torsion and Their Usage in Cryptography. Report 839/2018, Cryptology ePrint Archive, (2018)
11. Jeon, D., Kim, C. H., Lee, Y.: Families of elliptic curves over quartic number fields with prescribed torsion subgroups. *Mathematics of Computation*, 80(276), 2395-2410. (2011)
12. Joye, M., Tibouchi, M., Vergnaud, D.: Huff's Model for Elliptic Curves, In: *Lecture Notes in Computer Science*, vol. 6197. pp 234-250. Springer (2008)
13. Justus, B., Loebenberger, D.: Differential addition in generalized Edwards coordinates. *Advances in Information and Computer Security*, 316-325 (2010)
14. Montgomery, P.: Speeding the Pollard and elliptic curve methods of factorization, In: *Mathematics of Computation*, vol. 48, issue 177, pp. 243-264 (1987)
15. Moody, D., Shumow, D.: Analogues of Vélu's formulas for isogenies on alternate models of elliptic curves. *Mathematics of Computation*, 85(300), 1929-1951. (2016)
16. Oliveira, T., Lopez, J., Hisil, H., Faz-Hernandez, A., Rodriguez-Henriquez, F.: How to (pre-)compute a ladder. Improving the performance of X25519 and X448. Report 264/2017, Cryptology ePrint Archive, (2017)
17. Orhon, N.-G., Hisil, H.: Speeding up Huff form of elliptic curves. *Designs, Codes and Cryptography*, 86(12), 2807-2823 (2018)
18. Okeya, K., Sakurai, K.: Efficient elliptic curve cryptosystems from a scalar multiplication algorithm with recovery of the y-coordinate on a Montgomery-form elliptic curve. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 126-141). Springer, Berlin, Heidelberg (2001)
19. Wu, H., Feng, R.: Elliptic curves in Huff's model. *Wuhan University Journal of Natural Sciences*, 17(6), 473-480 (2012)

Appendix A Magma code for checking the point recovery function

Below we give a Magma code which can be used to compute point recovery formula for Huff's curve presented in subsection 2.1 (which can be performed in Magma calculator <http://magma.maths.usyd.edu.au/calc/>).

Given formula for point recovery this program computes the normal form with respect to a Gröbner basis according to the method described in [6]. For correct formulas the normal form Nor should be equal to 0. Let's note, that in the code below $F1 = f(P), F2 = f(Q), F3 = f(P + Q), Z1 = x_P$ and a, b are coefficients of Huff's curve $H_{a,b}$.

```

Q:=Rationals();
Z:=Integers();
F<a,b>:=FunctionField(Q,2);
pF2<x,y>:=PolynomialRing(F,2);
pF4<x1,y1,x2,y2>:=PolynomialRing(F,4);
pF44<F1,F2,F3,Z1>:=PolynomialRing(F,4);

//Curve equation
E:= a*x*(y^2-1) - b*y*(x^2-1);
//Definition of compression function f
f:=x*y;
//Addition formula
x3:=(x1+x2)*(1+y1*y2)/((1+x1*x2)*(1-y1*y2));
y3:=(y1+y2)*(1+x1*x2)/((1-x1*x2)*(1+y1*y2));

//Evaluation of compression function for given points
f1:=Evaluate(f,[x1,y1]);
f2:=Evaluate(f,[x2,y2]);
f3:=Evaluate(f,[x3,y3]);
//Point recovery formula
G:=[F1*F2*F3 + 2*a/b*F1*F2 + F1 + F2 - F3,
    F1*F2*F3*Z1 - F1*Z1 - F2*Z1 - F3*Z1 - 2*a/b*Z1];

v:=[f1,f2,f3,x1];

I:=ideal<pF4|[ Evaluate(E,[x1,y1]), Evaluate(E,[x2,y2]) ]>;
Nor:=NormalForm(Numerator(x2 - Evaluate(G[1]/G[2],v)), I);
Nor;

```

Similar computations may be performed for general Huff's curves, by exchanging in the code above the curve equation, addition formula and point recovery formula with:

```
//Curve equation
```

```

E:=x*(a*y^2-1)-y*(b*x^2-1);
//Addition formula
x3:=(x1+x2)*(a*y1*y2+1)/((b*x1*x2+1)*(a*y1*y2-1));
y3:=(y1+y2)*(b*x1*x2+1)/((b*x1*x2-1)*(a*y1*y2+1));
//Point recovery formula
G:=[Z1*(a*b*F1*F2*F3 - F1 + 2*a*F2*F3 + F2 + F3),
    a*b*F1*F2*F3 + 2*a*F1*F3 + F1 - F2 + F3];

```

Appendix B Example of Huff's curves sufficient to the SIDH (SIKE) algorithm

In this appendix we present examples of Huff's curves sufficient to the SIDH (SIKE) algorithm.

B.1 Supersingular Huff curves with $\#E(\mathbb{F}_{p^2}) = (p + 1)^2$

All Huff's curves are given in form $H/\mathbb{F}_{p^2} : ax(y^2 - 1) = by(x^2 - 1)$, where the field characteristic p is $p = 2^k 3^m 5^n - 1$ and the defining polynomial f of the field \mathbb{F}_{p^2} has a form $f(t) = t^2 + 1$. The coefficients a and b are $a = t$, $b = -1$. The binary length of p is denoted as l ($l = \lfloor \log_2 p \rfloor + 1$). Additionally we give binary lengths $lm = \lfloor \log_2 3^m \rfloor + 1$ and $ln = \lfloor \log_2 5^n \rfloor + 1$. The group generator of order $r = 3^m 5^n$ is a point $P = (x1 \cdot t + x0, y1 \cdot t + y0)$.

Huff curve with $l = 438$

```

k = 2
m = 137, lm = 218
n = 94, ln = 219
l = 438
x1 = 0x0
x0 = 0xd46de8889b08d5638b90ca837d4224980558052dfd27f25bd030ff04898
    ee10f03fd1d298b17f08d7668a9a17b724f363a4f127ce1a77
y1 = 0x22e494735b8efc23d7dbd425873bc958af2a1fad5fbddf5fb5d45584098
    39fa3a6dbaa15d6ed1c7748a766bc8fc08d42ae679b2f13ef04
y0 = 0x0
r = 0xa906b869ab471b86ce23a2ae536f7efe269e3423d803cd9afd80ffefe1e
    d712236ad0d0581896b3d9e5bc6148decbf665f667e90ce9db

```

Huff curve with $l = 606$

```

k = 2
m = 193, lm = 306
n = 128, ln = 298
l = 606

```

```

x1 = 0x0
x0 = 0x54b3c452e59da9f32b03e8fdb83d79999a7e8ba59d0d04ee3b2080e051d
    4d539025b8fba6961cc0224cc4bc7b2eb681cfeba3c89bbb5427ecd2e5248
    31325aeb286c9240b939f9803c3b5da
y1 = 0x13f769b16767b2d90e361989a147bb2272c66540fe68cd392f6e476fb77
    a7112582c3934ce64edce6eb2c2e00cd6bc7c6a40e1d63d65058b4a8f4909
    fe129ddb8b04860cd94c1f2eeace648
y0 = 0x0
r = 0x899f0620bca00d6ab2977e1e513f3c719f9b5303590ecb7b133228b193e
    6ca793a3b04c16f18e9189ab232802a7208bd8d8d4c0d25281278f872bf09
    1d43325b2c30357d717ddd37afb5f03

```

Huff curve with $l = 778$

```

k = 2
m = 243, lm = 386
n = 168, ln = 391
l = 778
x1 = 0x0
x0 = 0x708a9cca6751c88d0da3e92b63bf18518629d0a7a8af61f479df636f615
    20a5c26137191f5ed0f8996bca16c022657c55c42bf5c6f73aca487d307f5
    4d3567ab040b78bb56dcfbfc52b0d1354385a50cd1f899e150baf5968aa64c
    d5fec3144bad4
y1 = 0x1f3387f6e2f61e4f1b04364b2cb3c636a24df98852a0826fa64cd82830d
    fe75fddf100a7d78d15280ede4fb55a56a704a98d3c064cb8f0a0b9727f62
    1eceb7e39f831580f4b4a6a77dd7b4a7fb5fb85eb58da7dbe054874ac5fc8
    a2830b1fbaca6d
y0 = 0x0
r = 0x961a54c3fd38fb39272b2b4fa8b8d16657b11c52c4b90de3cb469cc1087
    7f15819f17297b0f7ff9f9043aeae443c7349ff5ebe4d01095047a7c2fd05
    fc76ebeb76364d7ef80c93b60226769f564fd837a0c139bc96170b779dd46
    cc558b5513b7b

```

Huff curve with $l = 984$

```

k = 2
m = 310, lm = 492
n = 211, ln = 490
l = 984
x1 = 0x0
x0 = 0x51713c7837710f62243b9fe3abecb7b73f0e1bcaa59d9c16522062b545c
    17ed3aa573bd66cad50855dc20512ab2007a4c6e1f0b91441c32eeef1c53
    8d15ca379ceb7651fa4ba2d233ba42a1ea7360eed760af020eae58d2cc590
    d97a6b80c5fbf868896e09225ca06ebfd276689f4f103a7fe8dc54b54a965
    d6a
y1 = 0x46616d22015e1dca61ac0d50f1348ce2f88b1b9a9ab25dfba05c3dd1006

```



```

0de954712699a880f865e081d59dd878c6b9e94689daed31ccd782c83c2bf
32f3fab2ded444cb56991a0514814e67a3b431b432c1d80b8e3e1595550d9
c405e2f105328aa41e9e5b6cb35a097bb1adb0081c03ee7d01685cafbb1a6b
cc9a
y0 = 0x0
r = 0x2675307c42f76ca189217941ff01465670f9596c46902ed930566d6c7b6
d1feacdeb08fa70a88770cf3e5d3be80510f9ff5a19ee3793587bee4cf2e4
3e24c2552f45eac8f6150a0a3da09dda9abf496670a8dd6a7c5342fffecc0
84689d36b91479ff037819bb22885fd463f4bdf7b69849684a3b639a24b08
1c75

```

Huff curve with $l = 437$

```

k = 3
m = 137, lm = 218
n = 93, ln = 216
l = 437
x1 = 0x0
x0 = 0xf556024b224f26858c587863917321731a98e0671105d72073af29df6e
cbca2350afdbdc5cfdc58879def2dc3547c57df0ea86e0987c
y1 = 0x667931d08a99f7a02f3e2e11da2685c1325df3430de8f13ad6f3863295d
f69f150e9fdd8f67f22c7a4d2a64743e41945fbb98c892919b
y0 = 0x0
r = 0x21ce24e1ef0e38b48fa0ba22dd7cb2ffa152d73a5e67291eff80332ff9f
c49d3a489029ab381e23f86125ad0e92f597adfe14c835c85f

```

Huff curve with $l = 495$

```

k = 3
m = 155, lm = 246
n = 106, ln = 247
l = 495
x1 = 0x0
x0 = 0x14be109c0dadebc16931f7f48d36d10c14b3ae9bf5431859d096906d335
eccf2df8055cae7fc757f7ef3dd4608e3f58b9d5a37d223faa91606b7f21d
297f
y1 = 0x5214ab5207fcd92f2f07fdcae005cfdaff882a734642d6cb9bd0b61abb2
c124f892259b630f020dd5468bff85ed9e43ba1d218f9b9a2903214b3f87a
3a2e
y0 = 0x0
r = 0xddde84a3e7944b68f8c1d88cbe1cbc236b178d3f96fac3dd946973cbf
1adb3d898ffbd3e3bfa53cc1a577defab5d32fdc8d65fee632c9ccef47473
c63

```

Huff curve with $l = 829$

```

k = 3
m = 260, lm = 413
n = 178, ln = 414
l = 829
x1 = 0x0
x0 = 0x1101b3829732b77eb7bf32b78860c6daa32b7ccec442f4d569dcf2c7f1b
    08df6212b803cca13e1514b2c176a7a0e77e88e4fb02e2cce2786af7332e3
    73a74dd3d7751428a481488f914a6215bc09f91e4c33ec9b09c3dcfb7ef7a
    35e40492615eae42482f6f7767a
y1 = 0x118c03e2c8d3023c42c0e0c127892867f8cd5d116c85a0d579109fa83d8
    8529a37c247dd24125e7fdab9b1bdb4be23a7d39e1e800a9bd24645b8fa24
    708c31cb3e68e05debf9d26506c6070785d431f06907aeab37ca1a880f32e
    07976f039b42fc8ec358e4d6bf2
y0 = 0x0
r = 0x2a0873bec233634e591f4fbd3d78d2de3aabfd9a2478ebc2269aea248af
    722c70ce9248620051fd7e32577128888f57e2e52deb2a3fda4270b0cde39
    c97ada3f79637a6cd25377b6641ca1230086adefa8454beb27bc017762c8e
    17a522815b05f521e4e6899b29

```

Huff curve with $l = 802$

```

k = 3
m = 256, lm = 406
n = 169, ln = 393
l = 802
x1 = 0x0
x0 = 0xc475424d1acf4960fb616016318b6b12e2cf83f0b71151c11358d2d4fdc
    6cb25bda8d991187ad1920b4f3368e1a181e3fdbcfcf8015dfe1e3771230f
    22238ef0a305bc6fc9e6eb4189d999320204fed7a53c785d20c3e7506846b
    30c65fc8423f9343285
y1 = 0xe76c9477bbc4dc6c9373698e9228f8796ebe3fe5bece6f371cb3f7b263
    4926864f190b9ddd8eb955ca63817c7314bf714463c11ab6e0dd08b59bfe1
    ff4597f833415043f627edcfe95ac9a3451a04a2760f867f5c13373e94e93
    56834720311ec0bc4b9
y0 = 0x0
r = 0x47521701cbbde5afdf5343b374b66e1514f7f0c69a1cbf36ebcd63aa98e
    47577d5954f09e61b442835bd2b20c95d1f3eb3c75732099063001ca3a80f
    8bc7aecce85d7284ee7841dc99ca96d17f9cfd7ca1956bc94c024b44a1266
    1c741c11495cad884e5

```

Huff curve with $l = 768$

```

k = 3
m = 248, lm = 394
n = 160, ln = 372
l = 768

```

```

x1 = 0x0
x0 = 0x19e958151bee30f7310e8ba7252ab6d9821502ac3a15186b555da162b4b
    ac8ffe197a11c06a15c34399bd054a59f1c8cd9aeb7f464fe5df0a3f12977
    8d98d574609d0547ff5d1eeb027c7b769f4f09c09ab617d5457120100e23d
    0721a52ba7f
y1 = 0x5dbf665640e0b1a7a3574a186a11ce3ca1ae4366d368746b116d025162e
    33813388af95c50ef8f5bc7bb8e8337c3152402d59775feb4de07de56feb5
    da6c23a2d5f737aca61b95c357e20571876198d3da362ab12b5a9a5c0c00e
    9971f043bb
y0 = 0x0
r = 0x17e77d10229c90d8e08916437db4c940d76ae5772af9442c089a847249f
    611ee5a002e9a46bce43d14b0c6a9af06a39ad99be53eb7a00e05a7ea1813
    ba62b7d889a26d14ecb49c077dfa6cb1e49105ab881d013a3d26cc988bca9
    46deea3bbe1

```

Huff curve with $l = 982$

```

k = 3
m = 310, lm = 492
n = 210, ln = 488
l = 982
x1 = 0x0
x0 = 0x1ffddb1e76a07d1bac2c799ce36eb35c92da896c171c8d85a191078e6d5
    f749dc716b06dea18baae87adcc2b4e3c99cbf0e456455bd6b6bd90a4a2ab
    f9b1bed30e797c9421c4dff75912f0b6055c4013be5b2e5a7997f0c55756c
    91d76512853ed1d354017d3f6dedb846cf5c9560536cef114bc27c4096c6c
    403f
y1 = 0x1fa017bfccddc5ef6f6f39fac0d2d85343339388b2c049efc9486d9bf63
    f6b8eb5fd4c01c2c467a66ed42f5a603919a337611ad0cb93d1ab2ed5ef3c
    057a0bb23071639fc7f9a4964ef580b054ec86c47591a3d57bcfc1939035a
    a0f2be77393042f8c63bffa0674936c896a13005bfd83bcf2fcccf10ba70a
    40b
y0 = 0x0
r = 0x7b109b273cb15b9e839e50d330041447cfeab7c0e1cd62b70114915b249
    0662292f01cbb021b4b02972df72619a9cfecab9ec93e50ab4bfc75ca2da
    6075a11097462283137686872b9b92bb88ca847b021c5e218dd73cccc8ece
    7481f7158374b99671805256d4e7990e0ca8c64be1e7514dba57a52075680
    5b1

```

Huff curve with $l = 429$

```

k = 4
m = 133, lm = 211
n = 92, ln = 214
l = 429
x1 = 0x0

```

```
x0 = 0xa07c7dc83f95999fc189b467e5d645f1d6e3334a9f7ff466aadcefb248e
    41ad7af8513e4a451e2b506b189fbb22d7707a29d4e3b8e2
y1 = 0x178356537157dbf278ac4763841656b32e31b7b4c5d8e05286598252eb1
    673d80b27ef3ddd434c21c6764a250d360acbf8ece716c17
y0 = 0x0
r = 0x155e4616b6b62f389c8676ebb00806f3c1a073d1753167bda0dea00a194
    6586261ae77376f91f0fc89d7c5193a451ea4e60215d0d23
```

Huff curve with $l = 518$

```
k = 4
m = 164, lm = 260
n = 109, ln = 254
l = 518
x1 = 0x0
x0 = 0x19bbf19fe3e3fc84fbbfa3213b64a3461d89203b35ca7d73c07192ce788
    8108541f931b42e16b77430a9abfda4f37f37f6b48e79eb4b71908144625d
    4b1bb20906
y1 = 0x6bf8c87e78c95a0ebfba11ddf877864a99c67eaa99127b2eb437cd9d88e
    c8fffb18bb1f8a410430542cf40a5323c38b8429bbeff6931d91a83c414aaf
    64a025295
y0 = 0x0
r = 0x20897d751238ec545ddf45b5e036d658ab93e10d228918b249366456397
    61eaa9a22ea58b4ca26c76ce835427d01bb218770a8631935681e00e8966a
    cfdb01525
```

Huff curve with $l = 604$

```
k = 4
m = 189, lm = 300
n = 129, ln = 300
l = 604
x1 = 0x0
x0 = 0x2eaa540d453ff088919990eccefedf7291d52d27b5183b5f118ded546a4
    d2f7c9227911a645dafa1eb2a202117d828715422d4827afb291093f4a61d
    711784a1c5bd3f0ff844c9339af8e94
y1 = 0x3e5431ea00eee5f9c069cfbd201e82ebd1a0ac129a2fdb8b1901054ce35
    90227dbc1ae11a33b749e225121d234b3c19b72570a6c6cb8c0fcee18bd4c
    12cbc17996c6732b64c0f45424f1510
y0 = 0x0
r = 0x87ec12b1b722d133a6e7cb92e1a06e43f94a64f3810b73a2a4577734204
    c3ce98233f80e0ee93211fac2f5d3eab6283ccb03a6c44a9fad1bd29d9397
    39587d9618ae08952db85c1112ad5f
```

Huff curve with $l = 788$

```

k = 4
m = 247, lm = 392
n = 169, ln = 393
l = 788
x1 = 0x0
x0 = 0x59d52abcadc0213415f22ad0a7627b46eea7c0fbbd8a78baaa8823ac562
    3f40a45922fb75c562297b15ba4504117a2da71ae300ed06ac55660842f9e
    bd5610dd074a33badf30ce9c25823140a0ebdfa5a6a116242511ed05965d6
    231ec4015fb76cb9
y1 = 0x84c8895e5bc807686f0a9f6fc8e5d685beb815b1740a085705c553fd1dd
    121579c7ab1ffbddc247344f07b5697617133f9b28e6f266311d24ee7a001
    31d0ea197cdec090e3762cd5d55c93130d2d3f98a51a565544ac5044b233f
    2b1f2b151fc6fbec
y0 = 0x0
r = 0xed77a81a0f9b25716af74b8505ec6346e8bb31cef138c2f96096b5fd626
    5c0d0610afa49faf857676f3b1359b1f3a2641100e30fd2a3bc015c677a4a
    7868233b8203e899de6be5aef166d9aa138c51100151b057597275243cb11
    0143556d9831997

```

Huff curve with $l = 799$

```

k = 4
m = 251, lm = 398
n = 171, ln = 398
l = 799
x1 = 0x0
x0 = 0x6143ecbad5dd0820ed3e745ef5aaf57b252a08a7a2dc45ddb3179a852a6
    58ba4c867e2fd7a8a9fc1464dc61d75aecb7140d599828577e911a45bb25e
    80598956041f271b10b2093d4ae0bd7c88ad484a0cf1b6a6e7ef060d6a6ee
    7af9e20f757488f3233
y1 = 0x676accd4d0da6c850114e72700f17e9665b9d787080f3c2b446359dfb74
    b76459cbd42766492aa8da34e398b268a629622810d3f624d8eda056df31c
    9f361c3b4d745130fed49fec1c4c5ed01b65c927c6d95495843a1d7a0b1ba
    4dc9e0fef44cba40b3
y0 = 0x0
r = 0x7566780b625723b2e271e245f3bdadd4be6f0bcfdf219fe469d0825914f
    56e2304f9fd5d332366b612eda4c1080a02f898a80041829312a26c3ee865
    7126fbeb1b770eae920637bd4d3868fbc254a0ed379a6f2be2f28a609ba414
    982fd9ebfe8e056b6f

```

B.2 Supersingular Huff curves with $\#E(\mathbb{F}_{p^2}) = (p - 1)^2$

All Huff's curves are given in form $H/\mathbb{F}_{p^2} : ax(y^2 - 1) = by(x^2 - 1)$, where the field characteristic p is $p = 2^k 3^m 5^n + 1$ and the defining polynomial f of the field \mathbb{F}_{p^2} has a form $f(t) = t^2 + c$. The coefficients a and b have a form $a = a_1 \cdot t + a_0$, $b = b_1 \cdot t + b_0$. The binary length of p is denoted as l ($l = \lfloor \log_2 p \rfloor + 1$). Additionally

we give binary lengths $lm = \lfloor \log_2 3^m \rfloor + 1$ and $ln = \lfloor \log_2 5^n \rfloor + 1$. The group generator of order $r = 3^m 5^n$ is a point $P = (x1 \cdot t + x0, y1 \cdot t + y0)$.

Huff curve with $l = 523$

```

k = 2
m = 166, lm = 264
n = 111, ln = 258
l = 523
c = 2
a1 = 0x58061a9599eaa399283efc26845558943549831e531a477436f1f79554e
    4b6fa78a34d9c236a538ba5005cdddc85d6346d8623dcc3660ec1129b981a
    281fe4c987b
a0 = 0x4675259bcb33db8c1fa26e8cd75873170bfb1144fbe103ae183c0f5f1e5
    bfe4619113a5d596e5bc8941a83e8d485ab2e215c0c01202b334245235037
    c49b2d42af6
b1 = 0x605c14e219dade557f8ebec51f0a20687853dbbca096cd511527941cf62
    01ce1d4b5eaec59fa156a300cb664242552a007e0ccf727db00c132de3b5
    40bfdf57343
b0 = 0x65c60204ea02ecbc1a36a4fb34f5e78bbdf46ad53dfaa5939f2fc3506dd
    71a583a7db8469b280c8d8f437d2346f5b85eaf25fccbcaa635a3b9e3d0c0
    d9dcf055306
x1 = 0x5647094c932293944381b6ba24f231c3264b29ed7c8d41a8105c372f7b9
    d08efa6dc287c43655b761df3a18955b7176573414ef65cb40b8a4efe008f
    40845691b3a
x0 = 0x46a0628d70699e7596bc864ba273b7fc79087abb339ecd0dd2589dc5999
    5b2ac7a873404aaa015d115adb78beb63900d9d8721613d9a86059b753738
    5c2a990273d
y1 = 0x585e501a89d83cada9763c003711517cf72133dbf50c1dd1e0c04aa1d1b
    602d49e923b39267bb898b173392b682e0e79b1ef0a669ec2abe348239fab
    57a7156b7f7
y0 = 0x2e37f8acee7c6808c525f38ef81e35553188d308daf80fa29055cbe52f9
    8238b60f4af36d6f54942a0a27ddb9d50d9b9a4f10e258322a20dd4446a
    a21febcc3b3
r = 0x1c98d743e50407b626813c44da103263eccc8cc8b5a7eb4b258ce2fc88
    0d0f3f178aff7f6e5ac1546b816cf6fde8578780a03ff1927f0825ecc6c33
    e0af7c29585

```

Huff curve with $l = 547$

```

k = 2
m = 172, lm = 273
n = 117, ln = 272
l = 547
c = 2
a1 = 0x2c02ee06ff8d2b542a952492699f49d3e4dcc02a8c49bd0cb5477aea424

```

```

dd63d50cf9cf08b5d5105936f459c99f3490436087a9d956201cc1a16827e
309dc3b8f6125edd1
a0 = 0x4af11afa137d47ffe2887b3d581919242e1cf48ea97eb7f0524101dace6
018275198e9cc99d27760b9abb0c287e07021273455deb10b63751509ba15
211ad6c240f380323
b1 = 0x2a3c0ed65f5e66a0b988e9362342a8aea026c663cdeedb6ff01d3294514
38ad02dd2055e1dd6bb0f2a36968f27baf0cfae0cf67acf6d9d277cc128e0
864d804116a0a3c89
b0 = 0x32446f016f074e359fc1466893c97d6e2ec752ad86c0f266d9c75c5148b
4eaabde2041ba6d7b8bc6bd0ef2ed0b2011e7e59c036598aa87ee463b9fe6
9111c575d10175c88
x1 = 0x3bb598658eac57fb08aaf0d3987ac6262aa0042bb5f0fb76f43be81d8f5
caf296e67f1cbb9e0b3225b040e0592eb2ffa70dd71a9b18b7f449080560c
5590c7e7e547646fd
x0 = 0x29c0ad3f199440305876e72bfe3ba5c3e904ee9d19d40dcb0e4c06ac6fe
d82b6c4dea4fcff43d3b453e6e34dd9d92f2826a05c800df1faf63dbb126d
a0e3a4590547915ae
y1 = 0x1966bf92d14d3af8077048d8ba9fe88e0ae2c06b06384652e19465ff7a8
8d59bc49926f00a4d542845c6cdfd0a252cba93b7afd0a7b1d10185d79c06
18d364123d92f8e8b
y0 = 0x25f742b2404db42f22a88205eed117ed748ae654145edf194a4886a433b
45fae44f0eae81f55e7890ebcf8d574e306a540086b7806d8ebb32c3c6347
ba556285ec6fcfc2f
r = 0x136a5e7239c7b57bd571dff04dabb83a447b8641573cee15fb9d2af60e9
e71f601e448e5120cd4614b22b1064fddd9cc2002040e7c9aa7dc13a6b3c
b0a2046c370920ea5

```

Huff curve with $l = 758$

```

k = 2
m = 238, lm = 378
n = 163, ln = 379
l = 758
c = 2
a1 = 0x12c3b9888c699077491bd6f0e001856db43e2ae3d410ff41bfa68d0c62a
a530917c7d002d914a04fa4c67e9386a8bcd27cd45373f5b919fc56206ed
102c3753800afdb476a1dc0c121218e0f091887695e7ce902225f174dc413
a25042944
a0 = 0x236718f4bca662f6b96a5ab5a2eaf1cb9b3eaf8682816b10fa80a3d3163
a9b30fe7215cf6879bddefeb556671d86507e5b8913cef709f80aa7fc901
e0e4d7764ecabb4848f49467ceba5cedc940d73499908cf9012b0271b7aa1
45b444996
b1 = 0x26f6a88845f2d848818dee259c08964bc931bbbf1daa193351bd098dbe6
10ca9c9b7c5a939b2245369995356130deda928662f3ea62c9f81909f6a9f
429d8e0ce17e0e17716c12f4251d88639e1876b3cadf32aa60c5d37243dcb
47ebaecb4

```

```

b0 = 0x31c51d938f4ca0fbcb0226e6312e291813648830e8540f9c71331ad8f7
    d92936cba5b91da93fa175b8272354c168e7bfb4e5caa84a7d9028a6c5284
    684660d2a8f337441b1f09cf3d08917703ce89c148b111611235277714c67
    2df5328e9
x1 = 0x2469d774777d67fa638c3e062f42fc19fef15182035b3b4cf5dff1d7fea
    8881a9f8860a6f8ca58bdded9a354cfd0e2cf13dc730746e25116584df688
    4a60acb917e7b01b73d10fc50d287ec6146dae91e92fcb10fd944a0fa4ae6
    d7fe774d6
x0 = 0x1876661c4066f24ef7abe094bc5cc1cb9eeef7d029e3df54e190df9d41a
    68537dfb9f3d0ede6a316ce222a1c161727bccb72d0bb393f421a930d105a
    d12a94166e3eea82f373ca056252e8f58b62ca4f2e8f30fe2f340026603ce
    91c9d7b
y1 = 0x5ff8ad852727551e23c8a38f048d3b5bed9dee71fb482bdfce478c4c514
    c2979d822d00eb7b6f7e828a7c8593a63424b765cb83b5a049c4e59fd7055
    03e302cd01721fb94b56ae0fe96e878e6322c7457929eae59ffa4ab758382
    8723b01e
y0 = 0xf9a4e74b04e55b88a69a4021abcf07cb2423a8a9a7e9053510e3ac9713d
    994d9172355c839bde0d3cf921ce0425f479cd74ba112fab36f51a6d6e9e
    3cf41f6ee40d433ad9899d1612695bd74a2c562ee39a79ceb2e809524f71d
    58f84359
r = 0xcf44a58d18c5eacf67ce6dcfe51ae05520011ce64fcb5efbe41807a6c4b
    4f1203663a2d8cc43bfe04e8b5d8d3d6254de519773c012ab85ea9bc6262b
    5412a0471d4c5e5ecd8db4cac68071f57d58322b608e4fbd9b5ac7c3c5fc9
    ba70e915

```

Huff curve with $l = 780$

```

k = 2
m = 243, lm = 386
n = 169, ln = 393
l = 780
c = 2
a1 = 0x450dd8d469680cea90f9ffad7c5e9a8ed0b5dcb79fb04799a71038c4759
    4f1382820989f3a3694ea28333d9976f837908233d0479b4ffd7dc52e79d0
    ce2fd9feb8e7df31d9724c8c73313e9ac430f156202fd940e791ee1e79908
    e824af2cef1336
a0 = 0xb06eab4e2aa0be70c6d3ed16bb10aba23617e670be2870b93b5e416a76b
    4c7cfe508697497fd3d31b01aabe69d5e49929cc59adeb110c330dff4e2cf
    dde22742084989d54c212b2b06268e53f0a9550514f4fa89afa049b2b4959
    b1f60b89bf7289
b1 = 0x2ed49a9696908b6acf6549ce2562978ccb157cdf2415a9744c9b64437ed
    919b56540eeb0d0333f96fd03d53383e261ac3cc99a7a3e85c967550d77c4
    66e5b66b156c946f3e37e3da4908f769c4cb4d78a11eae5fd341d1752986
    6745b25cc7ef19
b0 = 0x8e2b1548e1776f44237701f3bc1a6730ad814365b82ce322d2e526ae869
    3fb8c39a1ff1270b364be67c957bc7c1c4b4390431a6d0dac022aa1617c1a

```



```

55dd9fe66297960f39e63319f2e82dd3dee637c8a5580012ad1d367af0212
3ac695bd4b1f0e
x1 = 0x5f01754e4d9a872a8c9e5e1440a02da24741dddde7ef7ff0bb985d08c9c0
a0945c89cf174b291939a28f6c35aa032277c35b35074ae411712da5d80f3
42ed0c486e125e6c78c80a5aa0d28f47500e64c3d8ca74de51865e516068e
2fce554c2df2c3
x0 = 0xa296819ad30368ade85794a81bb3fa1360f6de710a54ac608dc5f448d79
e11457533049ebe355a98c83b1c6ff736fac782b3203c79f9db1b878907cf
8d462f4ad0aae88d38bf9255e0f663011ee7256b92f442e637377bb7e3caa
2344c5f3870298
y1 = 0x8e5caa867913f350cf4082490d0563aa88050599a91f7f66eed4e7f7feb
9307b1b73f40638c1125feabec3fc8a61828564df1efb314b363ed680f2e7
4190930702dbba2ae2997e90e67ee09a9f4fc248a4f1936afe38f8bffaadd
804826b2001dc0
y0 = 0x92438846376cb80c032af57b8439e77dc8e3c98ebd6e5b99e98b2f48ff2
38778db060e3e8b0632dc75a7588ba2da1e1dad4bcf90b6b3503ebd8cab17
6a158132c5853d6cde1d8761490f850b2651d8c8f34aba0e59166a6647f80
bb8333a0b2c391
r = 0x2ee83a7d3f21ce81dc3d7d88e4b9c16ffb6758d9dd79d4572f8610fc52a
57b6b881b73cf674d7fe1dd1526967552e4071fcd9b781052e916646cef11
dee529b994f0f837ad83ee28e0ac0511caf8f391623c620aeee7339561526
1fdabb8a962967

```

Huff curve with $l = 832$

```

k = 2
m = 261, lm = 414
n = 179, ln = 416
l = 832
c = 2
a1 = 0x5c83e9c10a2e00682f29f9b110db31a747f924dc47c322e980aabb58dc8
203e5d08036138262549c5e4802475f9fe21380963597bd3fd061f521ecb2
6e7c3b45ad6b9d3a57be2e16f8241089f4fc45361ec071dd17ff297949ae2
fef7ac413f9f826f7a26de3caa6
a0 = 0x468bc40a9151ba7b2e34dd600fd73c97f0ec9a8b2d0692bb5d39b9a8a5f
106eb238fb4c8484fe9c223a80a26efea171c5efb48a77d8e637b69eac466
6a65284f4eb0634dc91b0785ffbfee52e801a905932c5c04cd3c4f096e72f
fdffc87cee6398d264cfe6d19cc
b1 = 0x2a522da3883e93ef7c21e27849fbd0b1b5962f5416115a4987a276da6bd
26b846af70ea32ecf9b21439ffb97acfe0c5a63cac2bc084717a151fdb987
02caeb0e45484fbe9cfc4fb0544dbb0c76f6e7208b070ff0c0ff4727bf6c4
e79abf5c4a268daa9604c9a6d2e
b0 = 0x881eb5041c085e2c2547425fe884542f422c3616d9e9e49fafc0e242eac
0078f1525e069133fd40e0e7da7a07e934be897b4ddedba81a9cedb555b34
f686cee8906236f14799a5a5b14b2267326af39f18bc8db6226b4f5f26b6b
73235fa5405e7f9d6ad38b9af4b

```

```

x1 = 0x3b0132e1c7866314491df3c23af85dd9c81324bc02956755133511f5d7d
703b16fbdda1f1f27786ed403f50087c6521c9739c452232780a6b42d6315
8c2d472cf0c755f781d2b7f463ae9374a1f5f6e322df9373407e6c326bfaf
cdecfb13c80387f1c6080bb1a5
x0 = 0x4d413c66e47f04ee630f0dbf2caf3cb21af2e3a896586ad840fa285363e
e9410add6237e9563ead427b96ff4b829e674ef711c0eb09a88751e4794c9
d7d36dce7c0464431fa1c9cbb619391ff2ecea9d0c88685ef52bf95dae347
63850b7005e207e601a871cd9c6
y1 = 0x300d6a034d289aea5553cb2a032fbb4cbc5947cf1396e4fd5955790698d
d91f6f09a256d4e50a0dc134939f1fb664cfc22b6bbd6814c1b836eae1aa
45c87172229cde6b620df4e66905d9deab25e9540d92a1ebce6d3c33b02c3
5f58380adba2588e0695040bb9b
y0 = 0x8178d91d6940233a7fb9df117363ce14d7898f6f2dc5e421bcde28db206
3fc38bf76753519ed7e45072ea0eed20a9c2208434178966781d193fd4837
821e8ea97ccb4d50d810300a0640212dce3c29725b30527df781bffc587bf
0d0a61d7ba4f545dc009542b0f6
r = 0x2767ec82d6102d19738d5ac169a145b057013dc082315d0604313b82424
7b09a9c1a923dbe04cdda64f31fa1600066264b6db0c779bdc9e49a5c1056
2ce32c9b81cd42c6052e403afddad710d07e4310adc0f72c7540415fec9c5
362ad059455595cfc6982101767

```

Huff curve with $l = 972$

```

k = 2
m = 304, lm = 482
n = 210, ln = 488
l = 972
c = 2
a1 = 0x640e07dc1dcc0b7e698ed728761dee6d30d59ca07d7da9ddf0be88955de
ed6a97150146611d940c16d632e810479ff818904236d4e7f9d242a2e4cd7
c5efe03017fdf3062c4ebfa8c6fdb3253a03ff2f270b983301da023d5984
ff774e7432362e3aa975f7bef114fb725438a34c0d80b082927c7b032479c
e
a0 = 0x88e07aad7f062622973b62e0146f2dc8dda2870ccd8dee7ae3b765b6021
bdeb0d82833f4d3d68843bff6c070afdd33c47b936f8415f15d0b8b33e5d5
6c07a52ca67b45d8494d2ad0ae1bcad83c705efeb9b68363d2ad74e47f438
58a5a41e743fe062f8574418d91c5910302d47c282aede246405434666208
1
b1 = 0xa36c28995de00b28d4312835a8f0de1b04a962cb0046712907d0b4749db
1e24993c0562b93e58ce2f0b7314abea4d76709a041e193934f8524108cf7
dd072a68be750cba599953efaedda16a3a0cd5acaca477e19745242fac1ef
e7332697750bf34a97e1a9b58e6b2931801593ddfee18c6a1aa9a5a7543fe
8
b0 = 0x88fa4a5462344cd23b0430cc56d690c77cce3aefbdb39e91584abea613e
9d97cd1c76965087e4460794f5b8b2adff22835f88e24b5523f6f4d05406a
7be2732be7d800f1856ee05b9bd4909a82e2fc4bb2055587cb534d69ecf17

```

```
dd976f822c9f5fdc4f601a39ded93d2a97f02819cc8b31238a4ad8464c36d
d
x1 = 0x296696151ea34aeb8043de652214b6586aed37f9886e143c10d65ba1cc7
89b4b7bc1e79d4f1d2069f98548f6c470f09e52684188a814fdc67506e67e
96e69c679e81708d729a31a31b91ebe628c31f6a52512119a4a5ba1cc3b37
d386fd071e8cc6332d896b39faaa5d70d1603de32f1eae38eee8a13d461b2
f
x0 = 0x528ca5dea792755ac405e17612d2f273cf1a4fbd53c34425df0f99f3a0a
bab71a5d5af371563b81e94271325dcc613cc19ab9a8adffcfef7531c0a4c2
c42b88723d9186f6d77547614320f06f85a8cd5b6be2826bcb3ade096e581
5e8c2872d4d4b973ac6723041c035c7723a1703770f66d2330eb8e148c794
1
y1 = 0x47bdc124aa45ed4d7423a178ae7b4b7150a7abbb2417b1f75de1e64cd9b
3eaebacb83b9350da69de4fe379c3bf2a9b345f998484ab1c42f44fcf1746
4cedcf96ea3c9571d219f303c2e9908530392f1d16eac3ad699869be2549c
006ad51f2cac69aa2abff1e9ee4e9563db0c3c1b3e1cd45b5371ff3daf765
9
y0 = 0x3fefb528e90ce62c6de5d402c742a77cd83804024507dfb5c0418a0509e
74c0423d4ac03162dda1d9679de9fe77b77da2ee660959865227531c69e1
e5c1502d27e937a6104a3434566b174c60a5961edd798045fa5f4ea5f2199
1a9b8b98d9af8b0ed47fe9aa9c91ca0bfbd20ed07126a7233e66b683502bf
d
r = 0x2b37588c32cfaa2e88d21bae3e835d285170f9f18f8eb573e2e1a5e0181
3ce2e8e334f44d4d811e9bb95ef8c2682915b3ae65696f6e26619e89954ee
390cdce7fa205ac7d7137fa89e96d9faeb8f1b35614fd8035ae56510c7c98
90dbd1530e34edf21beb0b015875d9bdf87788dd46beb6348cfc3fec1a29
9
```