Application of Velusqrt algorithm to Huff's and general Huff's curves

Michał Wroński

Military University of Technology, Kaliskiego Str. 2, Warsaw, Poland
michal.wronski@wat.edu.pl

Abstract. In 2020 Bernstein, De Feo, Leroux, and Smith presented a new odd-degree ℓ-isogeny computation method called Velusqrt. This method has complexity $\tilde{O}(\sqrt{\ell})$, compared to the complexity of $\tilde{O}(\ell)$ of the classical Vélu method. In this paper application of the Velusqrt method to Huff's and general Huff's curves is presented. It is showed how to compute odd-degree isogeny on Huff's and general Huff's curves using Velusqrt algorithm and x-line arithmetic for different compression functions.

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

1 Introduction

In [1] Bernstein, De Feo, Leroux, and Smith presented an odd-degree isogeny computation method called Velusqrt. They modified the algorithm for the evaluation of polynomials whose roots are powers $h_S(\alpha) = \prod_{s \in S}(\alpha - \zeta^s)$, with complexity $\tilde{O}(\sqrt{\#S})$, to use a similar technique with x-line arithmetic for points on an elliptic curve to evaluate $h_S(\alpha) = \prod_{s \in S}(\alpha - f([s]P))$, where $f : E \to \mathbb{F}_q$ is compression function (in the case of Weierstrass and Montgomery curve $f(P) = x$, where $P = (x, y)$, is compression function of degree 2). Such an algorithm has complexity $\tilde{O}(\sqrt{\ell})$, where ℓ is the degree of the isogeny.

The core algorithm problem is contained in a more general framework, which is an efficient evaluation of polynomial and rational functions over $\mathbb{F}_q$ whose roots are values of a function from a cyclic group to $\mathbb{F}_q$. In such case, one has to fix a cyclic group G, a generator P of G and a function $f : G \to \mathbb{F}_q$. For each finite subset S of $\mathbb{Z}$, one then defines polynomial $h_S(X) = \prod_{s \in S}(X - f([s]P))$, where $[s]P$ is the sum of s copies of P (group G is written additively).

So, given f and S, one wants then to evaluate $h_S(X)$ at point α, for any $\alpha \in \mathbb{F}_q$. The standard way of computation of $h_S(\alpha)$ requires $O(\#S)$ operations in $\mathbb{F}_q$. Even though, if S has enough additive structure and f is sufficiently compatible with the group structure on G, then one can compute $h_S(\alpha)$ in $\tilde{O}(\sqrt{\#S})$ operations in $\mathbb{F}_q$. This idea is applied, e.g., in Pollard's and Strassen's algorithms of factorization.

We now define an index system.

Definition 1 *[1, Definition 4.6] Let I and J be finite sets of integers.*

1. *We say that (I, J) is an index system if the maps $I \times J \to Z$ defined by $(i, j) \to i + j$ and $(i, j) \to i - j$ are both injective and have disjoint images.*
2. *If S is a finite subset of $\mathbb{Z}$, then we say that an index system (I, J) is an index system for S if $I + J$ and $I - J$ are both contained in S.*

If (I, J) is an index system, then the sets $I + J$ and $I - J$ are both in bijection with $I \times J$. We write $I \pm J$ for the union of $I + J$ and $I - J$.

The main result of [1] is an adaptation of Pollard's idea to evaluate $h_{I\pm J}(\alpha)$, where $h_{I\pm J}(\alpha)$ is the kernel polynomial. The biggest problem which had to be solved is that $f([i + j]P)$ cannot be represented only by $f([i]P)$ and $f([j]P)$.

Even though it is possible to do the following trick. If $f(P)$ is a compression function (whose degree is coprime with the degree of the isogeny) on elliptic curve E, then exist rational functions F_0, F_1 and F_2 such that $\frac{F_1(f(P),f(Q))}{F_0(f(P),f(Q))} = -(f(P + Q) + f(P - Q))$ and $\frac{F_2(f(P),f(Q))}{F_0(f(P),f(Q))} = f(P + Q)f(P - Q)$. Then $(X - f(P+Q))(X - f(P-Q)) = X^2 + \frac{F_1(f(P),f(Q))}{F_0(f(P),f(Q))}X + \frac{F_2(f(P),f(Q))}{F_0(f(P),f(Q))}$. This property then leads to the following equations

$$h_{I\pm J}(X) = \prod_{(i,j)\in I\times J} (X - f([i + j]P))(X - f([i - j]P))$$

$$= \prod_{i\in I} \prod_{j\in J} (X^2 - A_2(f([i]P), f([j]P))X + A_1(f([i]P), f([j]P))).$$

It means that most of S cannot be decomposed as $I + J$, but such decomposition involves both $I + J$ and $I - J$. By using these observations, it is possible to construct Algorithm 1.

Algorithm 1: Computing $h_S(\alpha) = \prod_{s\in S} (\alpha - f([s]P))$, based on [1, Algorithm 2]

```
Data: a prime power $q$, an elliptic curve $E/\mathbb{F}_q, P \in E(\mathbb{F}_q)$, a finite
      subset $S \subset \mathbb{Z}$, an index system $(I, J)$ for $S$ such that
      $S \cap n\mathbb{Z} = I \cap n\mathbb{Z} = J \cap n\mathbb{Z} = \{\}$, where $n$ is the order of $P$
Input: $\alpha \in \mathbb{F}_q$
Output: $h_S(\alpha)$, where $h_S(X) = \prod_{s\in S} (X - f([s]P))$
 1. $h_I = \prod_{i\in I} (Z - f([i]P)) \in \mathbb{F}_q[Z]$
 2. $D_J = \prod_{j\in J} F_0(Z, f([j]P)) \in \mathbb{F}_q[Z]$
 3. $Res_Z (h_I, D_J) \in \mathbb{F}_q$
 4. $E_J = \prod_{j\in J} F_0(Z, f([j]P))\alpha^2 + F_1(Z, f([j]P))\alpha + F_2(Z, f([j]P)) \in \mathbb{F}_q[Z]$
 5. $h_K = \prod_{k\in S\setminus(I\pm J)} (\alpha - f([k]P)) \in \mathbb{F}_q$

return $\frac{h_K \cdot R}{\Delta_{I,J}}$
```

Example 1. We use the Example [1, Example 4.12]. Let us suppose that we want for Weierstrass curve, with compression $f(P) = x$ to evaluate $h_S(X) =$

$\prod_{s\in S}(X - x([s]P))$, where $S = \{1, 3, \ldots, \ell - 2\}$. Let us note that set S can be replaced by any set of representatives of $((\mathbb{Z}/\ell\mathbb{Z}) \setminus \{0\}) / \langle \pm 1 \rangle$.

Let $I = \{2b(2i + 1) | 0 \leq i \leq b'\}$ and $J = \{1, 3, \ldots, 2b - 1\}$ with $b = \left\lfloor \frac{\sqrt{\ell-1}}{2} \right\rfloor$ and (for $b > 0$) $b' = \left\lfloor \frac{\ell-1}{4b} \right\rfloor$. Then (I, J) is an index system for S. What is more $S \setminus (I \pm J) = K$, were $K = \{4bb' + 1, \ldots, \ell - 4, \ell - 2\}$. Algorithm 1 computes $h_S(\alpha)$ for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$.

As was shown in [1], the Velusqrt algorithm can be applied to the practical implementations of CSIDH and CSURF, obtaining faster solutions for $\ell \gtrapprox 110$ (it depends on many factors). The presented algorithm gives a 16% speedup for CSIDH-1024. In other presented situations, the speedup is less significant. Because the presented algorithm has much better asymptotic complexity than the method of Vélu for isogeny evaluation, for isogeny-based protocols with a higher level of security (e.g., CSIDH-2048, CSIDH-4096), the speedup should be much more significant.

What is more Chavez-Saab, Chi-Dominguez, Jaques and Rodríguez-Henriquez in [2] considered constant-time implementation of CSIDH using Velusqrt method.

In the next sections, basing on these ideas, will be showed how to adapt the Velusqrt algorithm to the Huff's and general Huff's models of elliptic curves.

2 Compression functions

The following section is mainly based on [3].

Compression on elliptic curves (often called x-line arithmetic) is mainly used to reduce key sizes and protect solutions against side-channel attacks. If E is an elliptic curve over a field K and $f : E \to K$ is a rational function, for which holds that $f(P) = f(-P)$ for all $P \in E$, then f is a compression function and for any $k \in \mathbb{Z}$ holds that $[k]f(P) = f([k]P)$. For example, on Weierstrass and Montgomery curves $f(x, y) = x$ is a compression function. Moreover, for compression function $f : E \to K$ there exist rational functions for doubling $D(x) \in K(x)$ and differential additions $\frac{F_1}{F_0}, \frac{F_2}{F_0} \in K(x, y)$ such that

$$f([2]P) = D(f(P)), \tag{1}$$

$$f(P + Q) + f(Q - P) = \frac{F_1(f(P), f(Q))}{F_0(f(P), f(Q))}, \tag{2}$$

$$f(P + Q)f(Q - P) = \frac{F_2(f(P), f(Q))}{F_0(f(P), f(Q))} \tag{3}$$

for any points $P, Q \in E$. After functions D and F_0 and F_1 or F_1 and F_2 are found, one can compute $[k]f(P)$ using values of f and the Montgomery ladder algorithm. There also exists a rational map $B : E \times K \times K \to E$ such that

$$Q = B(P, f(Q), f(P + Q)) \tag{4}$$

for generic points $P, Q \in E$, which we call the point recovery formula. Such formula allows for $P \in E$ to compute $[k]f(P)$ using the Montgomery ladder algorithm, which also gives $[k+1]f(P)$, and to recover point $[k]P$ on E given $P, [k]f(P), [k+1]f(P)$ substituting $Q=[k]P$ to the formula (4).

Presented above functions can be found using elementary methods (for example, in the case of Huff's and general Huff's curves for the compression function $f_2(x,y)=xy$, see [3]), but in many cases, it is tough to find suitable formulas using only elementary methods. That is why the Gröbner basis mechanism is often used in such cases, where searching for convenient functions can be automatized. Description of such method is presented in [4] for the compression function of degree 2 and in [5] for compression functions of high-degree, where program from [4] was a little modified.

In this paper, when necessary, the method described in [5] for searching for suitable functions will be used. The correctness of the formulas presented in the paper can be checked using programs *Huff_Correctness_x_square* and *General-Huff_Correctness_x_square* from [6], which are analogous to the programs used for checking the correctness of the formulas presented in [3].

2.1 Huff's curves

Huff's curve over K is provided by the equation [7]

$$H_{a,b} \ : \ ax(y^2-1)=by(x^2-1), \tag{5}$$

where $a^2 \neq b^2$ and $a,b \neq 0$. The neutral element is the point $O=(0,0)$ and for any point $P=(x_P,y_P)$ the opposite point is equal to $-P=-(x_P,y_P)=(-x_P,-y_P)$. The addition law for two points $P=(x_P,y_P)$, $Q=(x_Q,y_Q)$ on $H_{a,b}$ is given by

$$\begin{cases} x_R = \frac{(x_P+x_Q)(1+y_Py_Q)}{(1+x_Px_Q)(1-y_Py_Q)}, \\ y_R = \frac{(y_P+y_Q)(1+x_Px_Q)}{(1-x_Px_Q)(1+y_Py_Q)}, \end{cases} \tag{6}$$

where $P+Q=(x_R,y_R)$.

Doubling and differential addition on Huff's curve using a compression function $f_2(x,y)=xy$ are given by [3]

$$f_2([2]P)=\frac{4f_2(P)(f_2(P)^2+\left(\frac{b}{a}+\frac{a}{b}\right)f_2(P)+1)}{(f_2(P)^2-1)^2}, \tag{7}$$

$$f_2(P+Q)f_2(P-Q)=\left(\frac{f_2(P)-f_2(Q)}{f_2(P)f_2(Q)-1}\right)^2. \tag{8}$$

One can also find

$$\begin{aligned} &f_2(P+Q)+f_2(P-Q) \\ &=\tfrac{2(f_2(P)f_2(Q)^2+f_2(P)^2f_2(Q)+2\frac{b}{a}f_2(P)f_2(Q)+2\frac{a}{b}f_2(P)f_2(Q)+f_2(Q)+f_2(P))}{(f_2(P)f_2(Q)-1)^2} \end{aligned} \tag{9}$$

using the method described in [4]. The correctness of the formulas presented above can be checked using the program *Huff_diff_add_doub_rec_correctness_checking* from [6].

Moreover, one can also find the formula for point recovery. For generic points $P = (x_P, y_P), Q = (x_Q, y_Q)$ on $H_{a,b}$ if we are given $P, f_2(Q), f_2(P+Q)$, then coordinates of Q are provided by

$$\begin{cases} x_Q = f_2(Q)\frac{(y_Pf_2(P+Q)+x_P)(bf_2(Q)+a)+(af_2(Q)+b)(x_Pf_2(P+Q)+y_P)}{(bf_2(Q)+a)(f_2(P+Q)-f_2(Q)+x_Py_P(f_2(Q)f_2(P+Q)-1))}, \\ y_Q = \frac{f_2(Q)}{x_Q}. \end{cases} \tag{10}$$

In projective coordinates, formulas (7) and (8) can be computed as efficiently as formulas [8] for Montgomery curves. In this way, doubling requires $2M+2S+c$, and differential addition has a cost equal to $4M+2S$. More details can be found in [3].

2.2 Huff's isogenies computation using compression techniques

This subsection will present formulas for isogeny computation from [3], where it is also shown how to compute isogeny of an odd degree using a compression function $f_2(x,y) = xy$ of degree 2.

Theorem 1. *[3] 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 ψ. Let $A = \prod_{i=1}^{s} \alpha_i$ and $B = \prod_{i=1}^{s} \beta_i$. Let us define*

$$\psi(P) = \left(x_P(-1)^s \prod\nolimits_{Q\neq(0,0)\in F} x_{P+Q}, y_P(-1)^s \prod\nolimits_{Q\neq(0,0)\in F} y_{P+Q}\right). \tag{11}$$

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 $b' = \frac{b}{B^2} = \frac{b}{\prod_{i=1}^{s} y_{Q_i}^2}$.

Theorem 2. *[3] Let F be the kernel of the odd-degree isogeny ψ. For compression function of degree 2 given by $f_2(x,y) = xy$ let us note that $f_2(\psi(P))$ is provided by*

$$f_2(\psi(P)) = \left(x_P(-1)^s \prod_{Q\neq(0,0)\in F} x_{P+Q} \cdot y_P(-1)^s \prod_{Q\neq(0,0)\in F} y_{P+Q}\right), \tag{12}$$

which is equal to

$$f_2(\psi(P)) = \left(x_Py_P \prod\nolimits_{Q\neq(0,0)\in F} x_{P+Q}y_{P+Q}\right) = \left(f_2(P) \prod\nolimits_{Q\in F^+} f_2(P+Q)f_2(P-Q)\right), \tag{13}$$

where F^+ is the set $\{(\alpha_i, \beta_i) : i = 1, \dots, s\}$.

To find the coefficients a' and b' of Huff's curve $H_{a',b'}$, if $f_2(P) = x_Py_P = r_P$, one can use formulas from [3] for x^2 and y^2 as rational functions of r, where $r = xy$ is compression function of degree 2:

$$\begin{aligned} x^2 &= \frac{r(ar+b)}{br+a}, \\ y^2 &= \frac{r(br+a)}{ar+b}. \end{aligned} \tag{14}$$

Finally

$$\begin{aligned} a' &= \frac{a}{\left(\prod_{i=1}^{s} x_{Q_i}\right)^2} = a\prod_{i=1}^{s} \frac{(br_{Q_i}+a)}{r_{Q_i}(ar_{Q_i}+b)}, \\ b' &= \frac{b}{\left(\prod_{i=1}^{s} x_{Q_i}\right)^2} = b\prod_{i=1}^{s} \frac{(ar_{Q_i}+b)}{r_{Q_i}(br_{Q_i}+a)}. \end{aligned} \tag{15}$$

2.3 General Huff's curves

General Huff's curves over K are provided by the equation [9]

$$G_{\overline{a},\overline{b}} \ : \ \overline{x}(\overline{a}\overline{y}^2 - 1) = \overline{y}(\overline{b}\overline{x}^2 - 1), \tag{16}$$

where $\overline{a} \neq \overline{b}$ and $\overline{a}, \overline{b} \neq 0$. The neutral element is the point $\overline{O} = (0, 0)$, and for any point $\overline{P} = (\overline{x}_P, \overline{y}_P)$ the opposite point $-\overline{P} = -(\overline{x}_P, \overline{y}_P) = (-\overline{x}_P, -\overline{y}_P)$. The addition law for two points $\overline{P} = (\overline{x}_{\overline{P}}, \overline{y}_{\overline{P}})$, $\overline{Q} = (\overline{x}_{\overline{Q}}, \overline{y}_{\overline{Q}})$ on $H_{\overline{a},\overline{b}}$ is given by

$$\begin{cases} \overline{x}_{\overline{R}} = \frac{(\overline{x}_{\overline{P}}+\overline{x}_{\overline{Q}})(\overline{a}\overline{y}_{\overline{P}}\overline{y}_{\overline{Q}}+1)}{(\overline{b}\overline{x}_{\overline{P}}\overline{x}_{\overline{Q}}+1)(1-\overline{a}\overline{y}_{\overline{P}}\overline{y}_{\overline{Q}})}, \\ \overline{y}_{\overline{R}} = \frac{(\overline{y}_{\overline{P}}+\overline{y}_{\overline{Q}})(\overline{b}\overline{x}_{\overline{P}}\overline{x}_{\overline{Q}}+1)}{(1-\overline{b}\overline{x}_{\overline{P}}\overline{x}_{\overline{Q}})\overline{a}\overline{y}_{\overline{P}}\overline{y}_{\overline{Q}}+1)}, \end{cases} \tag{17}$$

where $\overline{R} = \overline{P} + \overline{Q} = (\overline{x}_{\overline{R}}, \overline{y}_{\overline{R}})$.

Doubling and differential addition on general Huff's curve using the compression function $f_2(\overline{x}, \overline{y}) = \overline{xy}$ are given by

$$f_2([2]\overline{P}) = \frac{4f_2(\overline{P})\left(\overline{a}\overline{b}f_2(\overline{P})^2+\left(\overline{a}+\overline{b}\right)f_2(\overline{P})+1\right)}{(\overline{a}\overline{b}f_2(\overline{P})^2-1)^2}, \tag{18}$$

$$f_2(\overline{P} + \overline{Q})f_2(\overline{P} - \overline{Q}) = \left(\frac{f_2(\overline{P})-f_2(\overline{Q})}{\overline{a}\overline{b}f_2(\overline{P})f_2(\overline{Q})-1}\right)^2. \tag{19}$$

One can also find

$$\begin{aligned} &f_2(\overline{P} + \overline{Q}) + f_2(\overline{P} - \overline{Q}) \\ =\ &\frac{2(\overline{a}\overline{b}f_2(\overline{P})f_2(\overline{Q})^2+\overline{a}\overline{b}f_2(\overline{P})^2f_2(\overline{Q})+2\overline{b}f_2(\overline{P})f_2(\overline{Q})+2\overline{a}f_2(\overline{P})f_2(\overline{Q})+f_2(\overline{Q})+f_2(\overline{P}))}{(\overline{a}\overline{\overline{b}}f_2(\overline{P})f_2(\overline{Q})-1)^2} \end{aligned} \tag{20}$$

using the method described in [4]. The correctness of the formulas presented above can be checked using the program *General_Huff_diff_add_doub_rec_correctness_checking* from [6].

Moreover, one can also find the formula for point recovery. For generic points $\overline{P} = (\overline{x}_P, \overline{y}_P), \overline{Q} = (\overline{x}_Q, \overline{y}_Q)$ on $G_{\overline{a},\overline{b}}$, if we are given $\overline{P}, f_2(\overline{Q}), f_2(\overline{P} + \overline{Q})$, then the coordinates of $\overline{Q}$ are provided by

$$\begin{cases} \overline{x}_2 = f_2(\overline{Q})\frac{(\overline{a}\overline{y}_1f_2(\overline{P}+\overline{Q})+\overline{x}_1)(\overline{b}f_2(\overline{Q})+1)+(\overline{a}f_2(\overline{Q})+1)(\overline{b}\overline{x}_1f_2(\overline{P}+\overline{Q})+\overline{y}_1)}{(\overline{b}f_2(\overline{Q})+1)(f_2(\overline{P}+\overline{Q})-f_2(\overline{Q})+\overline{x}_1\overline{y}_1(\overline{a}\overline{b}f_2(\overline{Q})f_2(\overline{P}+\overline{Q})-1)}, \\ \overline{y}_2 = \frac{f_2(\overline{Q})}{\overline{x}_2}. \end{cases} \tag{21}$$

2.4 General Huff's isogenies computation using compression techniques

In this subsection will be presented formulas for isogeny computation from [10] and from [3], where it is shown how to compute isogeny of an odd degree using a compression function $f_2(\overline{x},\overline{y}) = \overline{xy}$ of degree 2.

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

Theorem 3. *([10], Theorem 5.) Define*

$$\overline{\psi}(\overline{P}) = \left(\overline{x}_P \prod\nolimits_{\overline{Q}\neq(0,0)\in\overline{F}} \tfrac{-\overline{x}_{\overline{P}+\overline{Q}}}{\overline{x}_{\overline{Q}}}, \overline{y}_{\overline{P}} \prod\nolimits_{\overline{Q}\neq(0,0)\in\overline{F}} \tfrac{-\overline{y}_{\overline{P}+\overline{Q}}}{\overline{y}_{\overline{Q}}}\right). \tag{22}$$

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

Now we present how to compute isogeny $f_2(\overline{\psi})$ using point compression.

Corollary 1. *[3] Let $\overline{F}$ be the kernel of the odd-degree isogeny $\overline{\psi}$. For compression function of degree 2 given by $f_2(\overline{x},\overline{y}) = \overline{xy}$ let us note that $f_2(\psi(P))$ is provided by*

$$f_2(\overline{\psi}(\overline{P})) = \left(\overline{x}_{\overline{P}} \prod\nolimits_{\overline{Q}\neq(0,0)\in\overline{F}} \tfrac{-\overline{x}_{\overline{P}+\overline{Q}}}{\overline{x}_{\overline{Q}}} \cdot \overline{y}_{\overline{P}} \prod\nolimits_{\overline{Q}\neq(0,0)\in\overline{F}} \tfrac{-\overline{y}_{\overline{P}+\overline{Q}}}{\overline{y}_{\overline{Q}}}\right), \tag{23}$$

which is equal to

$$f_2(\overline{\psi}(\overline{P})) = \left(\overline{x}_{\overline{P}}\overline{y}_{\overline{P}} \prod\nolimits_{\overline{Q}\neq(0,0)\in\overline{F}} \tfrac{\overline{x}_{\overline{P}+\overline{Q}}\overline{y}_{\overline{P}+\overline{Q}}}{\overline{x}_{\overline{Q}}\overline{y}_{\overline{Q}}}\right) = \left(f_2(\overline{P}) \prod\nolimits_{\overline{Q}\in\overline{F}^+} \tfrac{f_2(\overline{P}+\overline{Q})f_2(\overline{P}-\overline{Q})}{f_2(\overline{Q})^2}\right), \tag{24}$$

where $\overline{F}^+$ is the set $\{(\overline{\alpha}_i, \overline{\beta}_i) : i = 1 \dots s\}$.

To find the coefficients $\overline{a}'$ and $\overline{b}'$ of general Huff's curve $G_{\overline{a}',\overline{b}'}$, one can use similar formulas to these from [3] for $\overline{x}^2$ and $\overline{y}^2$ as rational functions of $\overline{r}$, where $\overline{r} = \overline{xy}$ is compression function of degree 2:

$$\begin{aligned} \overline{x}^2 &= \tfrac{\overline{r}(\overline{ar}+1)}{\overline{b}\overline{r}+1}, \\ \overline{y}^2 &= \tfrac{\overline{r}(\overline{b}\overline{r}+1)}{\overline{ar}+1}. \end{aligned} \tag{25}$$

Finally,

$$\begin{aligned} \overline{a}' &= \overline{a}^{\ell}\overline{B}^4 = \overline{a}^{\ell} \prod\nolimits_{i=1}^{s} \left(\tfrac{\overline{r}_{\overline{Q}_i}(\overline{b}\overline{r}_{\overline{Q}_i}+1)}{\overline{ar}_{\overline{Q}_i}+1}\right)^2, \\ \overline{b}' &= \overline{b}^{\ell}\overline{A}^4 = \overline{b}^{\ell} \prod\nolimits_{i=1}^{s} \left(\tfrac{\overline{r}_{\overline{Q}_i}(\overline{ar}_{\overline{Q}_i}+1)}{\overline{b}\overline{r}_{\overline{Q}_i}+1}\right)^2. \end{aligned} \tag{26}$$

3 Application to Velusqrt

This section shows how to apply the Velusqrt algorithm for Huff's and general Huff's curves. From the computational point of view, the following corollary will be important

Corollary 2 *If compression function f_d is of degree d and $GCD(d, \ell) = 1$, where ℓ is odd one can evaluate $h_S(X) = \prod_{s \in S}(X - f_d([s]P))$ using set S and an index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.*

Let us note that in such case (the same as in the case of the compression function of degree 2), if F is a group of order ℓ, then for every $P_1, P_2 \in F$ holds $f_d(P_1) = f_d(P_2)$ iff $P_1 = \pm P_2$, as same as for compression function of degree 2. It means that for every $i, j \in S$ holds that $f_d([i]P) = f_d([j]P)$ iff $i = \pm j$ so there will not be any additional redundancy.

3.1 Compression functions of degree 4

As will be shown later, to apply the Velusqrt technique to the Huff's and general Huff's curves isogeny computation, it is convenient to use the formula for $f_{4,x^2}(P+Q) + f_{4,x^2}(P-Q), f_{4,x^2}(P+Q)f_{4,x^2}(P-Q)$ or $f_{4,y^2}(P+Q) + f_{4,y^2}(P-Q), f_{4,y^2}(P+Q)f_{4,y^2}(P-Q)$, where $f_{4,x^2}(P) = x_P^2$ is compression function of degree 4, similarly $f_{4,y^2}(P) = y_P^2$ is compression function of degree 4 also.

We will show that $f_{4,x^2}(P) = x_P^2$ is the compression function of degree 4.

In [11], Kohel was studied symmetric quartic models over binary fields with a rational 4-torsion point T. He showed that a genus one curve admits translations by rational points and translation morphism $\tau_T = P + T$ on curve E is projectively linear (induced by a linear transformation of the ambient projective space), iff E is a degree n model determined by a complete linear system in $\mathbb{P}^{n-1}$ and T is in the n-torsion subgroup. Such a method was used in [5] to obtain high-degree compression functions on many alternative models of elliptic curves.

In this paper, we use his ideas to find new compression functions of high degree (degree 4) for Huff's curves and general Huff's curves. The compression functions for which we are looking for are invariant on the action of involution and translation by specific point T, in this case of order 2, which means that for the compression function of degree 4 holds that $f_4(P) = f_4(Q)$ iff $Q = \pm P + [k]T$, for $k = \overline{0, 1}$.

Let us note that if $r = x^2$, then for each r we can find two distinct x's at most. Moreover, using Huff's curve equation, for each x one can find at most two distinct y's, which means that there are at most four distinct points $P_i, i = \overline{1, 4}$, having the same value of compression $f_{4,x^2}(P_i) = r$. One can find that the compression function $f_{4,x^2}(P)$ is invariant under involution and translation by 2-torsion point $(0 : 1 : 0)$, because $(x, y) + (0 : 1 : 0) = (-x, \frac{1}{y})$. Then, $r = f_{4,x^2}(P)$ for $P \in \{(x, y), (-x, -y), (-x, \frac{1}{y}), (x, -\frac{1}{y})\}$.

What is more, this compression function is the same on Huff's and general Huff's curve.

We will firstly find formulas for $f_{4,x^2}(P+Q)+f_{4,x^2}(P-Q)$ and $f_{4,x^2}(P+Q)f_{4,x^2}(P-Q)$ on Huff's curve. Let $r_P = f_{4,x^2}(P)$ and $r_Q = f_{4,x^2}(Q)$, then

$$f_{4,x^2}(P+Q)+f_{4,x^2}(P-Q) = \frac{s_1(r_P, r_Q)}{s_0(r_P, r_Q)}, \tag{27}$$

$$f_{4,x^2}(P+Q)f_{4,x^2}(P-Q) = \frac{s_2(r_P, r_Q)}{s_0(r_P, r_Q)}, \tag{28}$$

where

$$\begin{aligned} s_0(r_P, r_Q) &= \left(r_P r_Q - 1\right)^2, \\ s_1(r_P, r_Q) &= -2\left({r_P}^2 r_Q + r_P {r_Q}^2 + \tfrac{8a^2-4b^2}{b^2} r_P r_Q + r_P + r_Q\right), \\ s_2(r_P, r_Q) &= \left(r_P - r_Q\right)^2. \end{aligned} \tag{29}$$

Formula for doubling $f_{4,x^2}([2]P)$ is equal to $\frac{N(r,a,b)}{D(r,a,b)}$, where

$$\begin{aligned} N(r,a,b) &= 4r\left(r^2 + \tfrac{4a^2-2b^2}{b^2} r + 1\right), \\ D(r,a,b) &= \left(r^2-1\right)^2. \end{aligned} \tag{30}$$

We can similarly find formulas for $f_{4,y^2}(P+Q)+f_{4,y^2}(P-Q)$ and $f_{4,y^2}(P+Q)f_{4,y^2}(P-Q)$

$$f_{4,y^2}(P+Q)+f_{4,y^2}(P-Q) = \frac{t_1(r_P, r_Q)}{t_0(r_P, r_Q)}, \tag{31}$$

$$f_{4,y^2}(P+Q)f_{4,y^2}(P-Q) = \frac{t_2(r_P, r_Q)}{t_0(r_P, r_Q)}, \tag{32}$$

where

$$\begin{aligned} t_0(r_P, r_Q) &= \left(r_P r_Q - 1\right)^2, \\ t_1(r_P, r_Q) &= -2\left({r_P}^2 r_Q + r_P {r_Q}^2 + \tfrac{8b^2-4a^2}{a^2} r_P r_Q + r_P + r_Q\right), \\ t_2(r_P, r_Q) &= \left(r_P - r_Q\right)^2. \end{aligned} \tag{33}$$

Proof. Formulas presented in (29) and (33) can be obtained using the method described in [5]. The correctness of the formulas presented above can be checked using the program *Huff_Correctness_x_square* from [6].

Similarly, we will show that $f_{4,x^2}(\overline{P}) = \overline{x}^2$ is compression function of degree 4 on general Huff's curve. Let us note that if $\overline{r} = \overline{x}^2$, then for each $\overline{r}$ we can find two distinct $\overline{x}$'s at most. Moreover, using Huff's curve equation, for each $\overline{x}$ one can find at most two distinct $\overline{y}$'s, what means that there are at most four distinct points $\overline{P}_i, i = \overline{1,4}$, having the same value of compression $f_{4,\overline{x}^2}(\overline{P}_i) = \overline{r}$. One can find that the compression function $f_{4,\overline{x}^2}(\overline{P})$ is invariant under involution and translation by 2-torsion point $(0:1:0)$, because $(\overline{x}, \overline{y}) + (0:1:0) = (-\overline{x}, \frac{1}{\overline{a}y})$. Then, $\overline{r} = f_{4,\overline{x}^2}(\overline{P})$ for $\overline{Q} \in \{(\overline{x}, \overline{y}), (-\overline{x}, -\overline{y}), (-\overline{x}, \frac{1}{\overline{a}y}), (\overline{x}, -\frac{1}{\overline{a}y})\}$.

We will firstly find formulas for $f_{4,\overline{x}^2}(\overline{P}+\overline{Q})+f_{4,\overline{x}^2}(\overline{P}-\overline{Q})$ and $f_{4,\overline{x}^2}(\overline{P}+\overline{Q})f_{4,\overline{x}^2}(\overline{P}-\overline{Q})$ on general Huff's curve. Let $\overline{r}_1=f_{4,\overline{x}^2}(\overline{P})$ and $\overline{r}_2=f_{4,\overline{x}^2}(\overline{Q})$, then

$$f_{4,\overline{x}^2}(\overline{P}+\overline{Q})+f_{4,\overline{x}^2}(\overline{P}-\overline{Q})=\frac{\overline{s}_1(\overline{r}_1,\overline{r}_2)}{\overline{s}_0(\overline{r}_1,\overline{r}_2)}, \tag{34}$$

$$f_{4,\overline{x}^2}(\overline{P}+\overline{Q})f_{4,\overline{x}^2}(\overline{P}-\overline{Q})=\frac{\overline{s}_2(\overline{r}_1,\overline{r}_2)}{\overline{s}_0(\overline{r}_1,\overline{r}_2)}, \tag{35}$$

where

$$\begin{aligned}
\overline{s}_0(\overline{r}_1,\overline{r}_2)&=\left(\overline{r}_1\overline{r}_2-\tfrac{1}{\overline{b}^2}\right)^2,\\
\overline{s}_1(\overline{r}_1,\overline{r}_2)&=\tfrac{2}{\overline{b}^2}\left(\overline{r}_1^2\overline{r}_2+\overline{r}_1\overline{r}_2^2+\tfrac{8\overline{a}-4\overline{b}}{\overline{b}^2}\overline{r}_1\overline{r}_2+\tfrac{1}{\overline{b}^2}\overline{r}_1+\tfrac{1}{\overline{b}^2}\overline{r}_2\right),\\
\overline{s}_2(\overline{r}_1,\overline{r}_2)&=\tfrac{1}{\overline{b}^4}\left(\overline{r}_1-\overline{r}_2\right)^2.
\end{aligned} \tag{36}$$

Formula for doubling $f_{4,\overline{x}^2}([2]\overline{P})$ is equal to $\frac{N(\overline{r},\overline{a},\overline{b})}{D(\overline{r},\overline{a},\overline{b})}$, where

$$\begin{aligned}
N(\overline{r},\overline{a},\overline{b})&=\tfrac{4}{\overline{b}^2}\overline{r}\left(\overline{r}^2+\tfrac{4\overline{a}-2\overline{b}}{\overline{b}^2}\overline{r}+\tfrac{1}{\overline{b}^2}\right),\\
D(\overline{r},\overline{a},\overline{b})&=\left(r^2-\tfrac{1}{\overline{b}^2}\right)^2.
\end{aligned} \tag{37}$$

We can similarly find formulas for $f_{4,\overline{y}^2}(\overline{P}+\overline{Q})+f_{4,\overline{y}^2}(\overline{P}-\overline{Q})$ and $f_{4,\overline{y}^2}(\overline{P}+\overline{Q})f_{4,\overline{y}^2}(\overline{P}-\overline{Q})$

$$f_{4,\overline{y}^2}(\overline{P}+\overline{Q})+f_{4,\overline{y}^2}(\overline{P}-\overline{Q})=\frac{\overline{t}_1(\overline{r}_1,\overline{r}_2)}{\overline{t}_0(\overline{r}_1,\overline{r}_2)}, \tag{38}$$

$$f_{4,\overline{y}^2}(\overline{P}+\overline{Q})f_{4,\overline{y}^2}(\overline{P}-\overline{Q})=\frac{\overline{t}_2(\overline{r}_1,\overline{r}_2)}{\overline{t}_0(\overline{r}_1,\overline{r}_2)}, \tag{39}$$

where

$$\begin{aligned}
\overline{t}_0(\overline{r}_1,\overline{r}_2)&=\left(\overline{r}_1\overline{r}_2-\tfrac{1}{\overline{a}^2}\right)^2,\\
\overline{t}_1(\overline{r}_1,\overline{r}_2)&=\tfrac{2}{\overline{a}^2}\left(\overline{r}_1^2\overline{r}_2+\overline{r}_1\overline{r}_2^2+\tfrac{8\overline{b}-4\overline{a}}{\overline{a}^2}\overline{r}_1\overline{r}_2+\tfrac{1}{\overline{a}^2}\overline{r}_1+\tfrac{1}{\overline{a}^2}\overline{r}_2\right),\\
\overline{t}_2(\overline{r}_1,\overline{r}_2)&=\tfrac{1}{\overline{a}^4}\left(\overline{r}_1-\overline{r}_2\right)^2.
\end{aligned} \tag{40}$$

Proof. Formulas presented in (36) and (40) can be obtained using the method described in [5]. The correctness of the formulas presented above can be checked using the program *GeneralHuff_Correctness_x_square* from [6].

3.2 Velusqrt on general Huff's curves

Moody and Shumov showed in [10] how to obtain ℓ-isogeny on general Huff's curves using kernel polynomials.

The isogeny formula is in this case given by

$$\overline{\psi}=\left(\frac{\overline{xg}(\overline{x})}{\overline{g}(0)(b\overline{x})^{2s}\overline{g}\left(\frac{1}{\overline{b}\overline{x}}\right)},\frac{\overline{y}\overline{h}(\overline{y})}{\overline{h}(0)(\overline{ay})^{2s}\overline{h}\left(\frac{1}{\overline{ay}}\right)}\right), \tag{41}$$

where $\overline{a}' = \overline{a}^{\ell}\overline{h}(0)^2$ and $\overline{b}' = \overline{b}^{\ell}\overline{g}(0)^2$.

Let $\overline{F} = \{(0,0), (\overline{\alpha}_i, \overline{\beta}_i), (-\overline{\alpha}_i, -\overline{\beta}_i) : i = 1 \dots s\}$, where $-(\overline{\alpha}_i, \overline{\beta}_i) = (-\overline{\alpha}_i, -\overline{\beta}_i)$, is the kernel of an isogeny $\overline{\psi}$ of degree ℓ, where $\ell = 2s+1$. Functions $\overline{g}(\overline{x})$ and $\overline{h}(\overline{x})$ are given by

$$\begin{aligned} \overline{g}(x) &= \prod_{i=1}^{s} \left(x^2 - \overline{\alpha_i}^2\right), \\ \overline{h}(y) &= \prod_{i=1}^{s} \left(y^2 - \overline{\beta_i}^2\right). \end{aligned} \tag{42}$$

Using Corollary 2, we conclude that $\overline{g}(\alpha)$ and $\overline{h}(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

Theorem 4. *Let us note that using the compression function $f_2(\overline{P}) = \overline{xy} = \overline{r}$ one obtains that*

$$f_2\left(\psi(\overline{P})\right) = \left(\frac{\overline{r}\overline{g}_2\left(\frac{\overline{r}(\overline{a}\overline{r}+1)}{\overline{b}\overline{r}+1}\right)\overline{h}_2\left(\frac{\overline{r}(\overline{b}\overline{r}+1)}{\overline{a}\overline{r}+1}\right)}{\overline{g}_2(0)\overline{h}_2(0)\left(\overline{a}\overline{b}\overline{r}\right)^{2s}\overline{g}_2\left(\frac{\overline{b}\overline{r}+1}{\overline{b}^2\overline{r}(\overline{a}\overline{r}+1)}\right)\overline{h}_2\left(\frac{\overline{a}\overline{r}+1}{\overline{a}^2\overline{r}(\overline{b}\overline{r}+1)}\right)} \right), \tag{43}$$

where $\overline{r}_i = \overline{\alpha}_i^2, \overline{g}_2(z) = \prod_{i=1}^{s}\left(z - \frac{\overline{r}_i(\overline{a}\overline{r}_i+1)}{\overline{b}\overline{r}_i+1}\right), \overline{h}_2(z) = \prod_{i=1}^{s}\left(z - \frac{\overline{r}_i(\overline{b}\overline{r}_i+1)}{\overline{a}\overline{r}_i+1}\right)$ and $\overline{a}' = \overline{a}^{\ell}\overline{h}_2(0)^2$ and $\overline{b}' = \overline{b}^{\ell}\overline{g}_2(0)^2$.

Proof. The formula for evaluation of the isogeny $f_2\left(\overline{\psi}(\overline{P})\right)$ is a straightforward adaptation of the formula (41). What is more, function $\overline{g}_2(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \frac{\overline{r}(\overline{a}\overline{r}_{\overline{G}}+1)}{\overline{b}\overline{r}_{\overline{G}}+1}$, where $\overline{r}_{\overline{G}} = \overline{x}_{\overline{G}}\overline{y}_{\overline{G}}$ for $\overline{G} = (\overline{x}_{\overline{G}}, \overline{y}_{\overline{G}})$ and $f(\overline{P}) = \frac{\overline{r}(\overline{a}\overline{r}+1)}{\overline{b}\overline{r}+1}$, where $\overline{r} = \overline{xy}$ for any $\overline{P} = (\overline{x}, \overline{y})$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to $\overline{s}_0, \overline{s}_1, \overline{s}_2$, respectively.

In the same manner, function $\overline{h}_2(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \frac{\overline{r}(\overline{b}\overline{r}_{\overline{G}}+1)}{\overline{a}\overline{r}_{\overline{G}}+1}$, where $\overline{r}_{\overline{G}} = \overline{x}_{\overline{G}}\overline{y}_{\overline{G}}$ for $\overline{G} = (\overline{x}_{\overline{G}}, \overline{y}_{\overline{G}})$ and $f(\overline{P}) = \frac{\overline{r}(\overline{b}\overline{r}+1)}{\overline{a}\overline{r}+1}$, where $\overline{r} = \overline{xy}$ for any $\overline{P} = (\overline{x}, \overline{y})$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to $\overline{t}_0, \overline{t}_1, \overline{t}_2$, respectively.

Using Corollary 2, we conclude that $\overline{g}_2(\alpha)$ and $\overline{h}_2(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

Theorem 5. *Let us note that using the compression function $f_{4,\overline{x}^2}(\overline{P}) = \overline{x}^2 = \overline{r}$ one obtains that*

$$f_{4,\overline{x}^2}\left(\psi(\overline{P})\right) = \left(\frac{\overline{r}\overline{g}_{4,\overline{x}^2}\left(\overline{r}\right)^2}{\overline{g}_{4,\overline{x}^2}(0)^2(\overline{b}^2\overline{r})^{2s}\overline{g}_{4,\overline{x}^2}\left(\frac{1}{\overline{b}^2\overline{r}}\right)^2} \right), \tag{44}$$

where $\tilde{D}(\overline{r}_i, \overline{a}, \overline{b}) = \frac{\tilde{N}(\overline{r}_i,\overline{a},\overline{b})}{\tilde{D}(\overline{r}_i,\overline{a},\overline{b})}$ is a rational function of $\overline{r}_i, \overline{a}, \overline{b}$ returning $f_{4,\overline{y}^2}([2]\overline{P})$ having $\overline{r}_i = f_{4,\overline{x}^2}(\overline{P}) = \overline{\alpha}_i^2$, functions $\tilde{N}(\overline{r}_i, \overline{a}, \overline{b}), \tilde{D}(\overline{r}_i, \overline{a}, \overline{b})$ are given by

$$\begin{aligned} \tilde{N}(\overline{r}_i, \overline{a}, \overline{b}) &= \tfrac{4r}{b^2}\left(r + \tfrac{1}{b}\right)^2, \\ \tilde{D}(\overline{r}_i, \overline{a}, \overline{b}) &= r^4 + \tfrac{4a-4b}{b^2}r^3 + \tfrac{-8a+6b}{b^3}r^2 + \tfrac{4a-4b}{b^4}r + \tfrac{1}{b^4}, \end{aligned} \tag{45}$$

and $\overline{g}_{4,\overline{x}^2}(z) = \prod_{i=1}^{s}(z-\overline{r}_i), h_{4,\overline{x}^2}(z) = \prod_{i=1}^{s}\left(z-\tilde{D}(\overline{r}_i,\overline{a},\overline{b})\right)$ *and* $\overline{a}' = \overline{a}^{\ell}\overline{h}_{4,\overline{x}^2}(0)^2$, $\overline{b}' = \overline{b}^{\ell}\overline{g}_{4,\overline{x}^2}(0)^2$.

Proof. The formula for evaluation of the isogeny $f_{4,\overline{x}^2}\left(\overline{\psi}(\overline{P})\right)$ is a straightforward adaptation of the formula (41). What is more, function $\overline{g}_{4,\overline{x}^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \overline{r}_{\overline{G}}$, where $\overline{r}_{\overline{G}} = \overline{x}_{\overline{G}}^2$ for $\overline{G} = (\overline{x}_{\overline{G}}, \overline{y}_{\overline{G}})$ and $f(\overline{P}) = \overline{r}$, where $\overline{r} = \overline{x}^2$ for any $\overline{P} = (\overline{x}, \overline{y})$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to $\overline{s}_0, \overline{s}_1, \overline{s}_2$ given by Equation (36), respectively.

In the same manner, function $\overline{h}_{4,\overline{x}^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \overline{r}_{\overline{G}}$, where $\overline{r}_{\overline{G}} = \overline{y}_{\overline{G}}^2$ for $\overline{G} = (\overline{x}_{\overline{G}}, \overline{y}_{\overline{G}})$ and one can replace $f(\overline{P}) = \overline{r}$, where $\overline{r} = \overline{y}^2$ for any $\overline{P} = (\overline{x}, \overline{y})$ by $f_{4,\overline{y}^2}([2]\overline{P}) = \tilde{D}(\overline{r}, \overline{a}, \overline{b})$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to $\overline{t}_0, \overline{t}_1, \overline{t}_2$ given by Equation (40), respectively.

Proof of formulas for coefficients of curves given in Theorem 5 is analogous to the proof of Theorem 8 for Huff's curve. Formula for $f_{4,\overline{y}^2}([2]\overline{P})$ knowing $f_{4,\overline{x}^2}(\overline{P}) = \overline{r}$ can be found using the method described in [5]. The correctness of the formulas presented above can be checked using the program *GeneralHuff_Correctness_x_square* from [6].

It is worth noting that this formula is equal to $\frac{\tilde{N}(\overline{r},\overline{a},\overline{b})}{\tilde{D}(\overline{r},\overline{a},\overline{b})}$, where functions $\tilde{N}(\overline{r},\overline{a},\overline{b}), \tilde{D}(\overline{r},\overline{a},\overline{b})$ are given by Equation (45).

Having the one compression of the element of the kernel $f_{4,y^2} = \beta_i^2$ we can also obtain other elements, using, for example, formulas for differential addition given in (36) and doubling, obtained by using the method described in [5].

This formula is given by $\frac{N(\overline{r},\overline{b},\overline{a})}{D(\overline{r},\overline{b},\overline{a})}$ (remember that general Huff's curve is symmetric: $G(\overline{x},\overline{y})_{\overline{a},\overline{b}} = G(\overline{y},\overline{x})_{\overline{b},\overline{a}}$), where $N(\overline{r},\overline{a},\overline{b})$ and $D(\overline{r},\overline{a},\overline{b})$ are provided by (37).

The correctness of the formulas presented above can be checked using the program *GeneralHuff_Correctness_x_square* from [6].

Using Corollary 2, we conclude that $\overline{g}_{4,\overline{x}^2}(\alpha)$ and $\overline{h}_{4,\overline{x}^2}(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

Theorem 6. *Let us note that using the compression function $f_{4,\overline{y}^2}(\overline{P}) = \overline{y}^2 = \overline{r}$ one obtains that*

$$f_{4,\overline{y}^2}\left(\psi(\overline{P})\right) = \left(\frac{\overline{r}\overline{h}_{4,\overline{y}^2}\left(\overline{r}\right)^2}{\overline{h}_{4,\overline{y}^2}(0)^2(\overline{a}^2\overline{r})^{2s}\overline{h}_{4,\overline{y}^2}\left(\frac{1}{\overline{a}^2\overline{r}}\right)^2}\right), \tag{46}$$

where $\tilde{D}(\overline{r}_i,\overline{b},\overline{a}) = \frac{N(\overline{r}_i,\overline{b},\overline{a})}{D(\overline{r}_i,\overline{b},\overline{a})}$ is a rational function of $\overline{r}_i, \overline{b}, \overline{a}$ returning $f_{4,\overline{x}^2}([2]\overline{P})$ having $\overline{r}_i = f_{4,\overline{y}^2}(\overline{P}) = \overline{\beta}_i^2$, functions $N(\overline{r}_i,\overline{a},\overline{b}), D(\overline{r}_i,\overline{a},\overline{b})$ are given by Equation (45) *and $\overline{h}_{4,\overline{y}^2}(z) = \prod_{i=1}^{s}\left(z-\overline{\beta}_i^2\right), g_{4,\overline{y}^2}(z) = \prod_{i=1}^{s}\left(z-\tilde{D}(\overline{r}_i)\right)$ and*
$\overline{a}' = \overline{a}^{\ell}\overline{h}_{4,\overline{y}^2}(0)^2, \overline{b}' = \overline{b}^{\ell}\overline{g}_{4,\overline{y}^2}^2$.

Proof. The formula for evaluation of the isogeny $f_{4,\overline{y}^2}\left(\overline{\psi}(\overline{P})\right)$ is a straightforward adaptation of the formula (41). What is more, function $\overline{h}_{4,\overline{y}^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \overline{r}_{\overline{G}}$, where $\overline{r}_{\overline{G}} = \overline{y}^2_{\overline{G}}$ for $\overline{G} = (\overline{x}_{\overline{G}}, \overline{y}_{\overline{G}})$ and $f(\overline{P}) = \overline{r}$, where $\overline{r} = \overline{y}^2$ for any $\overline{P} = (\overline{x}, \overline{y})$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to $\overline{t}_0, \overline{t}_1, \overline{t}_2$ given by Equation (40), respectively.

In the same manner, function $\overline{g}_{4,\overline{y}^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \overline{r}_{\overline{G}}$, where $\overline{r}_{\overline{G}} = \overline{x}^2_{\overline{G}}$ for $\overline{G} = (\overline{x}_{\overline{G}}, \overline{y}_{\overline{G}})$ and one can replace $f(\overline{P}) = \overline{r}$, where $\overline{r} = \overline{x}^2$ for any $\overline{P} = (\overline{x}, \overline{y})$ by $f_{4,\overline{x}^2}([2]\overline{P}) = \tilde{D}(\overline{r}, \overline{b}, \overline{a})$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to $\overline{s}_0, \overline{s}_1, \overline{s}_2$ given by Equation (36), respectively.

Proof of formulas for coefficients of curves given in Theorem 6 is analogous to the proof of Theorem 8 for Huff's curve. Formula for $f_{4,\overline{x}^2}([2]\overline{P})$ knowing $f_{4,\overline{y}^2}(\overline{P}) = \overline{r}$ can be found using the method described in [5]. The correctness of the formulas presented above can be checked using the program *GeneralHuff_Correctness_x_square* from [6].

It is worth noting that this formula is equal to $\frac{\tilde{N}(\overline{r},\overline{b},\overline{a})}{\tilde{D}(\overline{r},\overline{b},\overline{a})}$ because of the symmetry of general Huff's curve, where functions $\tilde{N}(\overline{r}, \overline{a}, \overline{b}), \tilde{D}(\overline{r}, \overline{a}, \overline{b})$ are given by Equation (45).

Having the one compression of the element of the kernel $f_{4,x^2} = \alpha_i^2$ we can also obtain other elements, using, for example, formulas for differential addition given in (40) and doubling, obtained by using the method described in [5].

This formula is given by $\frac{N(\overline{r},\overline{a},\overline{b})}{D(\overline{r},\overline{a},\overline{b})}$, where $N(\overline{r}, \overline{a}, \overline{b})$ and $D(\overline{r}, \overline{a}, \overline{b})$ are provided by (37).

The correctness of the formulas presented above can be checked using the program *GeneralHuff_Correctness_x_square* from [6].

Using Corollary 2, we conclude that $\overline{g}_{4,\overline{y}^2}(\alpha)$ and $\overline{h}_{4,\overline{y}^2}(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

3.3 Velusqrt on Huff's curves

We can use Equation (41) to obtain an isogeny evaluation formula using kernel polynomials on Huff's curves. Let $F = \{(0,0), (\alpha_i, \beta_i), (-\alpha_i, -\beta_i) : i = 1 \ldots s\}$, where $-(\alpha_i, \beta_i) = (-\alpha_i, -\beta_i)$, is the kernel of an isogeny ψ of degree ℓ, where $\ell = 2s + 1$. Let us define functions $g(x)$ and $h(x)$ as

$$\begin{array}{r} g(x) = \prod_{i=1}^{s}\left(x^2 - \alpha_i^2\right), \\ h(y) = \prod_{i=1}^{s}\left(y^2 - \beta_i^2\right). \end{array} \qquad (47)$$

Let ξ be an isomorphism from Huff's curve $H_{a,b}$ to general Huff's curve $G_{\overline{a},\overline{b}}$, where $\overline{a} = \frac{1}{b^2}, \overline{b} = \frac{1}{a^2}$. For $P = (x, y)$ the isomorphism ξ has the form $\overline{P} = \xi(P) = (ax, by) = (\overline{x}, \overline{y})$, Using isomorphism $\xi : G_{\overline{a},\overline{b}} \to H_{a,b}$ and its inverse $\xi^{-1} : H_{a,b} \to G_{\overline{a},\overline{b}}$, we can make the following transformations. Using ξ, we can transform Equation (41)

$$\begin{aligned}
&\overline{\psi}(\overline{P}) = \overline{\psi}(\overline{x},\overline{y}) = \overline{\psi}\left(\xi_x(P),\xi_y(P)\right)\\
&= \left(\frac{axa^{2s}g(x)}{a^{2s}g(0)\left(\overline{b}a^2x\right)^{2s}g\left(\frac{1}{\overline{b}a^2x}\right)}, \frac{byb^{2s}h(y)}{b^{2s}h(0)(\overline{a}b^2y)^{2s}h\left(\frac{1}{\overline{a}b^2y}\right)}\right)\\
&= \left(\frac{axg(x)}{g(0)\left(\frac{1}{a^2}a^2x\right)^{2s}g\left(\frac{1}{\frac{1}{a^2}a^2x}\right)}, \frac{byh(y)}{h(0)\left(\frac{1}{b^2}b^2y\right)^{2s}h\left(\frac{1}{\frac{1}{b^2}b^2y}\right)}\right)\\
&= \left(\frac{axg(x)}{g(0)(x)^{2s}g\left(\frac{1}{x}\right)}, \frac{byh(y)}{h(0)(y)^{2s}h\left(\frac{1}{y}\right)}\right).
\end{aligned} \tag{48}$$

Using ξ^{-1} one obtains

$$\begin{aligned}
&\psi(P) = \xi\left(\overline{\psi}(\overline{P})\right) = \overline{\psi}\left(\frac{\xi_x(P)}{a'}, \frac{\xi_y(P)}{b'}\right)\\
&= \left(\frac{axg(x)}{(-1)^s\frac{a}{g(0)}g(0)(x)^{2s}g\left(\frac{1}{x}\right)}, \frac{byh(y)}{(-1)^s\frac{b}{h(0)}h(0)(y)^{2s}h\left(\frac{1}{y}\right)}\right)\\
&= \left(\frac{(-1)^s xg(x)}{x^{2s}g\left(\frac{1}{x}\right)}, \frac{(-1)^s yh(y)}{y^{2s}h\left(\frac{1}{y}\right)}\right),
\end{aligned} \tag{49}$$

because $a' = (-1)^s \frac{a}{g(0)}$ and $b' = (-1)^s \frac{b}{h(0)}$.

Using Corollary 2, we conclude that $g_2(\alpha)$ and $h_2(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

Theorem 7. *Let us note that using the compression function $f_2(P) = xy = r$ one obtains that*

$$f_2\left(\psi(P)\right) = \left(\frac{rg_2\left(\frac{r(ar+b)}{br+a}\right)h_2\left(\frac{r(br+a)}{ar+b}\right)}{r^{2s}g_2\left(\frac{br+a}{r(ar+b)}\right)h_2\left(\frac{ar+b}{r(br+a)}\right)}\right), \tag{50}$$

where $r_i = \alpha_i\beta_i$, $g_2(z) = \prod_{i=1}^{s}\left(z - \frac{r_i(ar_i+b)}{br_i+a}\right)$ and $h_2(z) = \prod_{i=1}^{s}\left(z - \frac{r_i(br_i+a)}{ar_i+b}\right)$ and $a' = (-1)^s\frac{a}{g_2(0)}$ and $b' = (-1)^s\frac{b}{h_2(0)}$.

Proof. The formula for evaluation of the isogeny $f_2\left(\psi(P)\right)$ is a straightforward adaptation of the formula (49). What is more, function $g_2(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \frac{r(ar_G+b)}{br_G+a}$, where $r_G = x_G y_G$ for $G = (x_G, y_G)$ and $f(P) = \frac{r(ar+b)}{br+a}$, where $r = xy$ for any $P = (x, y)$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to s_0, s_1, s_2 given by Equation (29), respectively.

In the same manner, function $h_2(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = \frac{r(br_G+a)}{ar_G+b}$, where $r_G = x_G y_G$ for $G = (x_G, y_G)$ and $f(P) = \frac{r(br+a)}{ar+b}$, where $r = xy$ for any $P = (x, y)$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to t_0, t_1, t_2 given by Equation (33), respectively.

Theorem 8. *Let us note that using the compression function $f_{4,y^2}(P) = x^2 = r$ one obtains that*

$$f_{4,x^2}\left(\psi(P)\right) = \left(\frac{r g_{4,x^2}\left(r\right)^2}{r^{2s} g_{4,x^2}\left(\frac{1}{r}\right)^2} \right), \tag{51}$$

where $\tilde{D}(r_i, a, b) = \frac{\tilde{N}(r_i,a,b)}{\tilde{D}(r_i,a,b)}$ is a rational function of r_i returning $f_{4,y^2}([2]P)$ having $r_i = f_{4,x^2}(P) = \alpha_i^2$, functions $\tilde{N}(r,a,b), \tilde{D}(r,a,b)$ are given by Equation (52) and $g_{4,x^2}(z) = \prod_{i=1}^{s}\left(z - r_i^2\right)$, $h_{4,x^2}(z) = \prod_{i=1}^{s}\left(z - \tilde{D}(r_i,a,b)\right)$ and $a' = (-1)^s \frac{a}{g_{4,x^2}(0)}$ and $b' = (-1)^s \frac{b}{h_{4,x^2}(0)}$.

Proof. We begin showing some observations. Let us note that for elements of kernel F having compression $r_i = \alpha_i^2$ we cannot decide what the value of is β_i, because all points $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i), (\alpha_i, -\frac{1}{\beta_i}), (-\alpha_i, \frac{1}{\beta_i})$ lie on the curve $H_{a,b}$ and all these points have the same value of compression f_{4,x^2}, but only two of these points belong to the kernel F: $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)$. Of course, having $r_i = \alpha_i^2$, one can find α_i by computing roots of degree 2 polynomial $r - x^2$. In such a case, both roots α_i and $-\alpha_i$ are proper because points $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)$ both belong to the kernel F of ℓ-isogeny.

In the next step, having α_i (or $-\alpha_i$, we may assume that we have α_i) it is necessary to find proper β_i. Having α_i and using Huff's curve equation, one can find two possible values of y-coordinate: $y_1 = \beta_i$ or $y_2 = -\frac{1}{\beta_i}$. Unfortunately, in this case, only one value is proper. Let us note that if one computes ℓ-isogeny, where ℓ is an odd number, then every element (α_i, β_i) of the kernel F has odd order, but $-(\alpha_i, \beta_i) + (0:1:0) = (\alpha_i, -\frac{1}{\beta_i})$ has an order equal to 2ℓ. So one can, for both possible values of y-coordinates $y_1 = \beta_i, y_2 = \frac{1}{\beta_i}$ check the order of element (α_i, y_j), for $j = 1, 2$ and then decide which element is the correct element of the kernel F. Unfortunately, this method seems to be slow and generally useless in practical implementations.

Now we will show another, much faster way of obtaining necessary compressions $\{\beta_i^2 : i = i, \ldots, s\}$ of points of kernel F. At first, let us note that we are interested in the computation of ℓ-degree isogenies, where ℓ is odd. What is more, if (α_i, β_i) belongs to the kernel of the isogeny, and then such point has order ℓ. Points $(-\alpha_i, \frac{1}{\beta_i}), (\alpha_i, -\frac{1}{\beta_i})$ can be obtained by translation of points $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)$ by 2-torsion point $(0:1:0)$, so their order must be equal to 2ℓ. Now we will show the most important observation. Let us note that for $P \in \{(\alpha_i, \beta_i), (-\alpha_i, -\beta_i), (\alpha_i, -\frac{1}{\beta_i}), (-\alpha_i, \frac{1}{\beta_i})\}$, however, one can obtain two different values of compressions $f_{4,y^2}(P)$, because $f_{4,y^2}(P) = \beta_i^2$ or $f_{4,y^2}(P) = \frac{1}{\beta_i^2}$, but for point $[2]P$ there is only one possible value of compression $f_{4,y^2}([2]P)$. What is more, if point $P \in F$, then $[2]P \in F$ iff $\#F$ is odd.

Points $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)$ are of order ℓ and points $(\alpha_i, -\frac{1}{\beta_i}) = -(\alpha_i, \beta_i) + (0:1:0), (-\alpha_i, \frac{1}{\beta_i}) = (\alpha_i, \beta_i) + (0:1:0)$ are of order 2ℓ. Now let us note that for $P \in \{(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)\}$, point $[2]P$ is of order ℓ and also $[2]P$ belongs to the kernel F generated by any of points $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)$.

On the other hand, for $P \in \{(\alpha_i, -\frac{1}{\beta_i}), (-\alpha_i, \frac{1}{\beta_i})\}$, however, point P is of order 2ℓ, but point $[2]P$ is of order ℓ and also $[2]P$ belongs to the kernel F generated by any of points $(\alpha_i, \beta_i), (-\alpha_i, -\beta_i)$, because $[2](\alpha_i, -\frac{1}{\beta_i}) = [2](-(\alpha_i, \beta_i) + (0 : 1 : 0)) = -[2](\alpha_i, \beta_i)$ and similarly $[2](-\alpha_i, \frac{1}{\beta_i}) = [2]((\alpha_i, \beta_i) + (0 : 1 : 0)) = [2](\alpha_i, \beta_i)$. Because points $[2](\alpha_i, \beta_i)$ and $-[2](\alpha_i, \beta_i)$ are opposite points, their values of compression f_{4,y^2}, are the same.

It means that for any point $P \in \{(\alpha_i, \beta_i), (-\alpha_i, -\beta_i), (\alpha_i, -\frac{1}{\beta_i}), (-\alpha_i, \frac{1}{\beta_i})\}$, the value of compression $f_{4,y^2}([2]P)$ is the same.

Let us note that if $\tilde{D}(r_i, a, b)$ is a rational function of r_i returning $f_{4,y^2}([2]P)$ having $r_i = f_{4,x^2}(P) = \alpha_i^2$, then $\prod_{i=1}^{s} \left(z - \beta_i^2\right) = \prod_{i=1}^{s} \left(z - \tilde{D}(r, a, b)\right)$, where also holds that $f_{4,y^2}(P) = \beta_i^2$

On the other hand, having the one compression of the element of the kernel $f_{4,y^2}(P) = \beta_i^2$ we can also obtain other elements, using, for example, formulas for differential addition given in (29) and doubling, obtained by using the method described in [5].

This formula is given by $\frac{N(r,b,a)}{D(r,b,a)}$ (remember that general Huff's curve is symmetric: $H(x, y)_{a,b} = H(y, x)_{b,a}$), where $N(r, a, b)$ and $D(r, a, b)$ are provided by(30).

The Formula for evaluation of the isogeny $f_{4,x^2}(\psi(P))$ is a straightforward adaptation of the formula (49). What is more, function $g_{4,x^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = r_G$, where $r_G = x_G^2$ for $G = (x_G, y_G)$ and $f(P) = r$, where $r = x^2$ for any $P = (x, y)$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to s_0, s_1, s_2 given by Equation (29), respectively.

In the same manner, function $h_{4,x^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = r_G$, where $r_G = y_G^2$ for $G = (x_G, y_G)$ and one can replace $f(P) = r$, where $r = y^2$ for any $P = (x, y)$ by $f_{4,y^2}([2]P) = \tilde{D}(r, a, b)$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to t_0, t_1, t_2 given by Equation (33), respectively.

The correctness of the formulas presented above can be checked using the program *Huff_Correctness_x_square* from [6].

Now we will show how to compute $f_{4,y^2}([2]P)$ having $f_{4,x^2}(P) = r$.

Theorem 9. *If $f_{4,x^2}(P) = r$, then $f_{4,y^2}([2]P) = \frac{\tilde{N}(r,a,b)}{\tilde{D}(r,a,b)}$, where*

$$\begin{aligned} \tilde{N}(r, a, b) &= 4\tfrac{a^2}{b^2} r(r + 1)^2, \\ \tilde{D}(r, a, b) &= r^4 + \tfrac{4a^2-4b^2}{b^2} r^3 + \tfrac{-8a^2+6b^2}{b^2} r^2 + \tfrac{4a^2-4b^2}{b^2} r + 1. \end{aligned} \tag{52}$$

Proof. This formula can be found using the program from [4] and using modifications for high-degree compressions described in [5]. Formula for $f_{4,y^2}([2]P)$ knowing $f_{4,x^2}(P) = r$ can be found using the method described in [5]. The correctness of the formulas presented above can be checked using the program *Huff_Correctness_x_square* from [6].

Using Corollary 2, we conclude that $g_{4,x^2}(\alpha)$ and $h_{4,x^2}(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

Theorem 10. *Let us note that using the compression function $f_{4,y^2}(P) = y^2 = r$ one obtains that*

$$f_{4,y^2}\left(\psi(P)\right) = \left(\frac{r h_{4,y^2}\left(r\right)^2}{r^{2s} h_{4,y^2}\left(\frac{1}{r}\right)^2}\right), \tag{53}$$

where $\tilde{D}(r_i, b, a) = \frac{\tilde{N}(r_i,b,a)}{\tilde{D}(r_i,b,a)}$ is a rational function of r_i, b, a returning $f_{4,x^2}([2]P)$ having $r_i = f_{4,y^2}(P) = \alpha_i^2$, functions $\tilde{N}(r, a, b), \tilde{D}(r, a, b)$ are given by Equation (52) and $h_{4,y^2}(z) = \prod_{i=1}^{s}\left(z - \beta_i^2\right), g_{4,y^2}(z) = \prod_{i=1}^{s}\left(z - \tilde{D}(r_i, b, a)\right)$ and $a' = \frac{a}{g_{4,y^2}(0)}$ and $b' = \frac{b}{h_{4,y^2}(0)}$.

Proof. The formula for evaluation of the isogeny $f_{4,y^2}\left(\psi(P)\right)$ is a straightforward adaptation of the formula (49). What is more, function $h_{4,y^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = r_G$, where $r_G = y_G^2$ for $G = (x_G, y_G)$ and $f(P) = r$, where $r = y^2$ for any $P = (x, y)$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to t_0, t_1, t_2 given by Equation (40), respectively.

In the same manner, function $g_{4,y^2}(z)$ can be computed using Algorithm 1, where for this function hold that $\alpha = r_G$, where $r_G = x_G^2$ for $G = (x_G, y_G)$ and one can replace $f(P) = r$, where $r = x^2$ for any $P = (x, y)$ by $f_{4,x^2}([2]P) = \tilde{D}(r, b, a)$. Functions F_0, F_1, F_2 appearing in Algorithm 1 are equal to s_0, s_1, s_2 given by Equation (29), respectively.

Proof of formulas for coefficients of curves given in Theorem 10 is analogous to the proof of Theorem 8 for Huff's curve. Formula for $f_{4,x^2}([2]P)$ knowing $f_{4,y^2}(P) = r$ can be found using the method described in [5]. The correctness of the formulas presented above can be checked using the program *Huff_Correctness_x_square* from [6]. It is worth noting that this formula is equal to $\frac{\tilde{N}(r,b,a)}{\tilde{D}(r,b,a)}$ because of the symmetry of Huff's curve, where functions $\tilde{N}(r, a, b)$, $\tilde{D}(r, a, b)$ are given by Equation (52).

Having the one compression of the element of the kernel $f_{4,x^2} = \alpha_i^2$ we can also obtain other elements, using, for example, formulas for differential addition given in (33) and doubling, obtained by using the method described in [5].

This formula is given by $\frac{N(r,a,b)}{D(r,a,b)}$, where $N(r, a, b)$ and $D(r, a, b)$ are provided by(30).

The correctness of the formulas presented above can be checked using the program *Huff_Correctness_x_square* from [6].

Using Corollary 2, we conclude that $g_{4,y^2}(\alpha)$ and $h_{4,y^2}(\alpha)$ can be computed using index system from Example 1 for any $\alpha \in \mathbb{F}_q$ in $\tilde{O}\left(\sqrt{\ell}\right)$ operations.

4 Conclusion

This paper presents the Velusqrt method's application to the Huff's and general Huff's curve models. Although the formula for the computation of ℓ-isogeny using kernel polynomial for general Huff's curve is known and was given in [10], we found a similar formula for the case of Huff's curves. What is more, we presented many different compression functions suitable for such applications. Presented by us, compression functions of degree 4 seem to be efficient for the evaluation of ℓ-isogeny. They seem to be also reasonable for computation of the ℓ-isogenous curves.

What is more, for all presented by us compression functions, to apply the Velusqrt algorithm (Algorithm 1), one can use the same index system as presented in Example 1.

Acknowledgments

I want to thank Robert Dryło and Tomasz Kijko for joint works on elliptic curves' compression functions. Moreover, I want to thank Robert Dryło for automating the process of finding differential addition and doubling formulas for a given compression function using the Gröbner basis mechanism and being the lead author of the programs from [4] and [5] that I adapted in this paper.

References

1. D. Bernstein, L. De Feo, A. Leroux, and B. Smith, "Faster computation of isogenies of large prime degree," *arXiv preprint arXiv:2003.10118*, 2020.
2. J. Chávez-Saab, J.-J. Chi-Domínguez, S. Jaques, and F. Rodríguez-Henríquez, "The SQALE of CSIDH: Square-root Vélu quantum-resistant isogeny action with low exponents." Cryptology ePrint Archive, Report 2020/1520, 2020. https://eprint.iacr.org/2020/1520.
3. R. Dryło, T. Kijko, and M. Wroński, "Efficient Montgomery-like formulas for general Huff's and Huff's elliptic curves and their applications to the isogeny-based cryptography." Cryptology ePrint Archive, Report 2020/526, 2020. https://eprint.iacr.org/2020/526.
4. R. Dryło, T. Kijko, and M. Wroński, "Determining formulas related to point compression on alternative models of elliptic curves," *Fundamenta Informaticae*, vol. 169, no. 4, pp. 285–294, 2019.
5. M. Wroński, T. Kijko, and R. Dryło, "High-degree compression functions on alternative models of elliptic curves and their applications," *Submitted to: Fundamenta Informaticae*.
6. R. Dryło, T. Kijko, and M. Wroński. https://github.com/Michal-Wronski/Huff-compression.git., 2020. [Online; accessed 18-January-2021].
7. M. Joye, M. Tibouchi, and D. Vergnaud, "Huff's model for elliptic curves," in *International Algorithmic Number Theory Symposium*, pp. 234–250, Springer, 2010.
8. P. L. Montgomery, "Speeding the Pollard and elliptic curve methods of factorization," *Mathematics of Computation*, vol. 48, pp. 243–264, 1987.

9. H. Wu and R. Feng, "Elliptic curves in Huff's model," *Wuhan University Journal of Natural Sciences*, vol. 17, no. 6, pp. 473–480, 2012.
10. D. Moody and D. Shumow, "Analogues of Vélu's formulas for isogenies on alternate models of elliptic curves," *Mathematics of Computation*, vol. 85, no. 300, pp. 1929–1951, 2016.
11. D. Kohel, "Efficient arithmetic on elliptic curves in characteristic 2," in *International Conference on Cryptology in India*, pp. 378–398, Springer, 2012.