Faster Scalar Multiplication on the x-Line: Three-Dimensional GLV Method with Three-Dimensional Differential Addition Chains

Hairong Yi[1,2(✉)], Guiwen Luo[1,2], and Dongdai Lin[1]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
{yihairong,luoguiwen,ddlin}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing 100049, China

Abstract. On the quadratic twist of a GLV curve, we explore faster scalar multiplication on its x-coordinate system utilizing three-dimensional GLV method. We construct and implement two kinds of three-dimensional differential addition chains, one of which is uniform and the other is non-uniform but runs faster. Implementations show that at about 254-bit security level, the triple scalar multiplication using our second differential addition chains runs about 26% faster than the straightforward computing using Montgomery ladder, and about 6% faster that the double scalar multiplication using DJB chains.

Keywords: Scalar multiplication · GLV methods · Differential Addition Chains · DJB chains

1 Introduction

Elliptic curve cryptography (ECC) plays an important role in the public key cryptosystems. Various schemes and numerous techniques about ECC have been studied to meet the different needs (basically, efficiency and security) in different settings. In 2014, Costello, Hisil and Smith [6] implemented a very fast elliptic curve scalar multiplication, optimized for Diffie-Hellman Key Exchange at the 128-bit security level. This very efficient scheme involved two crucial ideas: using x-coordinate-only systems and the two-dimensional GLV method on the x-line.

The idea for computing scalar multiplication on elliptic curves by only x-coordinates arose earlier. Montgomery's explicit formulas [14] for the arithmetic on x-coordinate of a Montgomery curve together with his eponymous ladder provided a full solution to its implementation. It is often used as a technique called point compression, for storing or transmitting fewer bits of information in

C. Carlet et al. (Eds.): C2SI 2019, LNCS 11445, pp. 236–253, 2019.
https://doi.org/10.1007/978-3-030-16458-4_14

some specific situation. The Gallant-Lambert-Vanstone (GLV) method [9] is a typical and important technique for speeding up scalar multiplication on certain kinds of elliptic curves. It exploits a fast endomorphism on the curve, replacing a single large scalar multiplication with two scalar multiplications with only a half bit lengths. Then this two-dimensional GLV will result in a twofold performance speedup using parallel computation. Since m-dimensional GLV would probably lead to m-fold performance acceleration by parallel computation, higher dimensional GLV method has also been intensively considered [8,10,12]. In 2009, Galbraith, Lin and Scott [8] proposed the so-called GLS curves and indicated that on restricted GLS curves with j-invariant 0 or 1728, the four-dimensional GLV can be implemented. Further in 2012, Longa and Sica [12] combined the ideas in [9] and [8] and realized the four-dimensional GLV on the quadratic twists of all previous GLV curves appeared in [9].

In [6], Costello et al. have shown a positive effect of applying the two-dimensional GLV method to the scalar multiplication on the x-line. Motivated by the potential acceleration of higher dimensional GLV method, in this paper we investigate the performance of applying the three-dimensional GLV method to the x-coordinate-only systems. We choose the elliptic curves using the approach of [12], which are originally tailored for four-dimensional GLV method but can be transformed to three-dimensional variants as well. The remaining key issue is the chosen of higher dimensional differential addition chains (DACs) used in multi-scalar multiplication, where the "differential" property of the chain is required by the incomplete (pseudo-) operations on the x-line.

The two-dimensional DACs have all kinds of constructions in literature. For example, Schoenmakers' chain in [17], Akishita's chain in [1]. In [6] another three different two-dimensional DACs are implemented: PRAC chains [13,17], AK chains [2] and DJB chains [3]. Each of them offers a different combination of speed, uniformity and constant-time execution. The research about higher dimensional DACs are comparatively less. Early in 2006, Brown [4] extended Bernstein's idea (i.e. DJB chain) to general d-dimensional DACs, but it has been patented. For bypassing it, around 2016 SubramanyaRao tried the three-dimensional analogue of Schoenmakers' chain in [15] and that of Akishita's chain in [16]. The first attempt failed because it was showed to be more expensive than one-dimensional Montgomery ladder. The second one succeeded in competing with Montgomery ladder by about 22% speedup at the 256-bit security level. However there were no results of comparing it with the two-dimensional DACs. The latest progress is made in [11], 2017, by Hutchinson and Karabina. They constructed a d-dimensional DAC which has some similarities with that of Brown's [4], and made specific theoretical comparisons between some known d-point multiplication algorithms showing the algorithm based on their construction to be superior.

In this paper, we propose two kinds of constructions of three-dimensional DACs. The first one is a straightforward extension to higher dimension of the DJB chain, which is totally different from the patented one of Brown's in [4]. It inherits the property of uniformity. The second one comes from our direct observation, which is not a higher dimensional analogue of any two-dimensional DAC.

It is not uniform, but runs very fast. And it runs faster than SubramanyaRao's second chain described in [16] since it saves one more point operation in each iteration. Our final comparison experiments on testing scalar multiplication on the x-coordinate system utilizing different dimensional GLV methods show that at about 254-bit security level, our uniform three-dimensional DAC runs about 9% faster than Montgomery ladder, and the non-uniform one runs about 26% faster than Montgomery ladder and about 6% faster than the two-dimensional DJB chain.

Paper Organization. The rest of the paper is arranged as follows. In Sect. 2, we begin with a brief introduction to the four-dimensional GLV method proposed in [12], and see how to deduce a three-dimensional GLV variant directly in the same settings. In Sect. 3, we describe the basic operations on the x-coordinate system and deploy higher dimensional GLV method on this system. Then in Sect. 4 we describe explicitly the construction of our two three-dimensional DACs, and give some examples. In Sect. 5, from both the theoretic and experimental aspect, we compare the performance of scalar multiplication on the x-coordinate utilizing different dimensional GLV methods together with different DACs. Finally in Sect. 6 we draw our conclusion.

2 Three and Four-Dimensional GLV Method

The 4-dimensional GLV method proposed by Longa and Sica in [12] consists of the chosen of elliptic curves that are equipped with two efficient endomorphisms and 4-dimensional scalar decomposition. They use the quadratic twists of elliptic curves, of which the idea comes from GLS curves, and propose a specific twofold Cornacchia-type algorithm for scalar decomposition. Here we give a brief introduction.

Let $\mathbb{F}_p$ be a finite field where p is prime. A GLV curve over $\mathbb{F}_p$ is an elliptic curve $E/\mathbb{F}_p$ of whom the group of rational points has an almost prime order (cofactor ≤ 4), equipped with an efficiently computable $\mathbb{F}_p$-endomorphism ϕ. Assume that $X^2 + rX + s \in \mathbb{Z}[X]$ is the characteristic polynomial of ϕ, and π is the p-th Frobenius endomorphism of E. Let $E'/\mathbb{F}_{p^2}$ be a quadratic twist of $E(\mathbb{F}_{p^2})$, via the twisting $\mathbb{F}_{p^4}$-isomorphism $t_2 : E \to E'$. We then obtain two efficient endomorphisms on E', $\varPhi = t_2\phi t_2^{-1}$ and $\varPsi = t_2\pi t_2^{-1}$, both defined over $\mathbb{F}_{p^2}$. Suppose that $\langle P\rangle \subset E'(\mathbb{F}_{p^2})$ is a large subgroup of prime order n. Then we have $\varPhi^2(P)+r\varPhi(P)+sP = \mathcal{O}_{E'}$ and $\varPsi^2(P)+P = \mathcal{O}_{E'}$, together with $\varPhi(P) = \lambda P$ and $\varPsi(P) = \mu P$ where $\lambda, \mu \in [1, n-1]$ is a root of X^2+rX+s modulo n, X^2+1 modulo n respectively.

Define the 4-dimensional GLV reduction map w.r.t. $\{1, \varPhi, \varPsi, \varPhi\varPsi\}$

$$\begin{aligned} \mathfrak{f}: \quad & \mathbb{Z}^4 \to \mathbb{Z}/n \\ (x_1, x_2, x_3, x_4) & \mapsto x_1 + x_2\lambda + x_3\mu + x_4\lambda\mu \pmod{n}. \end{aligned}$$

Then $\ker \mathfrak{f}$ is a lattice. Applying the twofold Cornacchia-type algorithm [12, 18] to $\ker \mathfrak{f}$, we can find a short basis $\{v_1, v_2, v_3, v_4\}$ to this lattice with $\max_i |v_i| \leq 3.41(\sqrt{1 + |r| + s})n^{1/4}$, where $|\cdot|$ denotes the maximum norm. For any scalar

$k \in [1, n)$, express $(k, 0, 0, 0) = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + \alpha_4 v_4$ as the $\mathbb{Q}$-linear combination of the basis $\{v_i\}$. Then round α_i to the nearest integers $a_i = \lfloor \alpha_i \rceil$, and let $(k_1, k_2, k_3, k_4) = (k, 0, 0, 0) - \sum_{i=1}^{4} a_i v_i$. Finally we have

$$kP = k_1 P + k_2 \Phi(P) + k_3 \Psi(P) + k_4 \Phi\Psi(P)$$

with

$$\max_i(|k_i|) \leq 6.82(\sqrt{1 + |r| + s})n^{1/4}.$$

For our demand in the following, actually we need the 3-dimensional GLV method. The above way for choosing target curves is exactly what we want. But we delete the dimension controlled by "$\Phi\Psi$" from the 4-dimensional version to obtain a 3-dimensional representation. As for the 3-dimensional scalar decomposition, following the same way as above, then we obtain a 3-dimensional GLV reduction map $\tilde{\mathfrak{f}}$ w.r.t. $\{1, \Phi, \Psi\}$. For finding a short basis of the lattice $\ker \tilde{\mathfrak{f}}$, we can apply LLL algorithm. Then using the way described as above we can obtain

$$kP = \ell_1 P + \ell_2 \Phi(P) + \ell_3 \Psi(P) \quad \text{with} \;\; \ell_i = O(k^{1/3}).$$

3 Projection to the x-Line

The projective line $\mathbb{P}^1$ can be viewed as a quotient variety of E by the subgroup $\{\pm 1\}$ of the automorphism group $\mathrm{Aut}(E)$ of E. Particularly, if $\{\pm 1\}$ is exactly the whole group $\mathrm{Aut}(E)$ then $\mathbb{P}^1$ is the so-called Kummer variety of E. When E is given by a Weierstrass equation (or Montgomery form as [6] exploited), then the quotient map $E \to \mathbb{P}^1$ is just the projection to x-coordinate.

Some of the operations of E are well-defined on the x-line, like the scalar multiplication. Assume that P is a point of E. Given the x-coordinate $x(P)$ of P and some scalar k, then $x(kP)$ can be computed using the well-known Montgomery ladder. Based on this property, Diffie-Hellman Key Exchange can be executed on the pure x-coordinate settings. Furthermore in [6] Costello et al. require E to be twist-secure, for ensuring the compactness of this system.

However the x-line is not a group. The typical obstruction is that, only given the x-coordinates $x(P)$, $x(Q)$ of two points P, Q of E, one can not distinguish $x(P \oplus Q)$ and $x(P \ominus Q)$. But once one of them is known, then the other is clear.

Basic Pseudo-Operations on the x-Line. In our settings, the target elliptic curve $E'/\mathbb{F}_{p^2}$ as described in Sect. 2 is always defined by a short Weierstrass equation: $y^2 = x^3 + a_4 x + a_6$. For reader's convenience, here we write the explicit formula (in projective coordinate form) for pseudo-doubling and pseudo-addition on the x-line, which one may refer to [5].

Let $x(P_i) = (X_i : Z_i)$, $i = 1, 2$ be the x-coordinates of two points P_1, P_2 on $E'(\mathbb{F}_{p^2})$. Let $x(P_1 \ominus P_2) = (\Delta X : \Delta Z)$. Assume $x(P_1 \oplus P_2) = (X : Z)$. Then

$$\begin{aligned} X &= \Delta Z\big(-4a_6 Z_1 Z_2 (X_1 Z_2 + X_2 Z_1) + (X_1 X_2 - a_4 Z_1 Z_2)^2\big), \\ Z &= \Delta X \big(X_1 Z_2 - X_2 Z_1\big)^2, \end{aligned}$$

and

$$x(2P_1) = \big((X_1^2 - a_4 Z_1^2)^2 - 8a_6 X_1 Z_1^3 : 4Z_1(X_1(X_1^2 + a_4 Z_1^2) + a_6 Z_1^3)\big).$$

Pseudo-Endomorphisms on the $\boldsymbol{x}$-Line. Every endomorphism of E induces a pseudo-endomorphism of $\mathbb{P}^1$, since it commutes with the negation map of E. On our target curve $E'/\mathbb{F}_{p^2}$, the two endomorphisms Φ and Ψ induce $\Phi_x : x \mapsto \Phi_x(x)$ and $\Psi_x : x \mapsto \Psi_x(x)$ on the x-line. And since they are commutative, we have

$$\Phi_x \Psi_x = (\Phi\Psi)_x = (\Psi\Phi)_x = \Psi_x \Phi_x.$$

Let $\langle P \rangle \subset E'(\mathbb{F}_{p^2})$ be the subgroup with large prime order as in Sect. 2. For any scalar k, we can find (k_1, k_2, k_3, k_4) with only a quater of the bit length of k, such that

$$kP = k_1 P + k_2 \Phi(P) + k_3 \Psi(P) + k_4 \Phi\Psi(P);$$

or we can find (ℓ_1, ℓ_2, ℓ_3) with only one third of the bit length of k, such that

$$kP = \ell_1 P + \ell_2 \Phi(P) + \ell_3 \Psi(P).$$

Denote by x the x-coordinate of P. Then we would like to compute $x(kP)$ using

$$(k_1 + k_2\Phi + k_3\Psi + k_4\Phi\Psi)_x(x) \tag{1}$$

$$\text{or} \qquad (\ell_1 + \ell_2\Phi + \ell_3\Psi)_x(x). \tag{2}$$

Actually here we only need to consider Eq. (2). We can expect to accelerate the computation of $x(kP)$ using this multi-scalar multiplication. But since there are only pseudo-operations on the x-line, addition chains are not feasible. And it forces us to propose a higher dimensional DAC to this (k_1, k_2, k_3, k_4) or (ℓ_1, ℓ_2, ℓ_3), just as the 2-dimensional case in [6], where computes

$$(m_1 + m_2\Phi)_x(x) \quad \text{or} \quad (m_1 + m_2\Psi)_x(x) \tag{3}$$

utilizing a 2-dimensional DAC.

4 Three-Dimensional Differential Addition Chains

A differential addition chain is an addition chain in which each sum is already accompanied by a difference, i.e. whenever a new chain element $M+N$ is formed by adding M and N, the difference $M-N$ was already in the chain. The chain is called n-dimensional, if every element in this chain has the form of n-tuple. For example, the well-known "Montgomery ladder" is a typical one-dimensional DAC.

In this section, we propose two kinds of constructions of 3-dimensional DACs. One is a higher dimensional analogue of the DJB chain. The other comes from observation directly, which is not a higher dimensional analogue of any 2-dimensional DAC.

4.1 A Uniform Three-Dimensional Differential Addition Chain

The DJB chain is a 2-dimensional DAC proposed by Bernstein in 2006. It is uniform, i.e. possesses the same execution pattern: add, double, add in each iteration, and hence can resist side-channel attacks. For more details one can refer to [3]. In the following, we generalize it to the 3-dimensional case.

Let S be the following set

$$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}.$$

As in the 2-dimensional case, in each iteration we compute part of the eight elements of the set $S(a,b,c) = (a,b,c) + S$.

We find that computing five of them is enough. For ensuring the property of uniform, the element of type (even, even, even) should be reserved. For a pair of elements in $S(a,b,c)$, if every component of them has the opposite parity, then we call them dual to each other. Among the following three dual pairs $\{$(odd, even, even), (even, odd, odd)$\}$, $\{$(even, odd, even), (odd, even, odd)$\}$, $\{$(even, even, odd), (odd, odd, even)$\}$, we omit one of two elements in every pair, and use three parameters d_1, d_2, d_3 to determine which ones are to be omitted, that is

$$\begin{aligned}
T_1 &= (a + (a + d_1 + 1 \mod 2),\ b + (b + d_1 \mod 2),\ c + (c + d_1 \mod 2)),\\
T_2 &= (a + (a + d_2 \mod 2),\ b + (b + d_2 + 1 \mod 2),\ c + (c + d_2 \mod 2)),\\
T_3 &= (a + (a + d_3 \mod 2),\ b + (b + d_3 \mod 2),\ c + (c + d_3 + 1 \mod 2)).
\end{aligned}$$

Next we give a recursive definition of the 3-dimensional differential addition chain.

Definition 1. *For a given 3-tuple of nonnegative integers (A,B,C), and $\{D_1, D_2, D_3\}$ where $D_1, D_2, D_3 \in \{0,1\}$, we define the set $C(\{D_i\}_{i=1}^3; A,B,C)$ recursively, as the set $C(\{d_i\}_{i=1}^3; a,b,c)$ added with the following five elements:*

$$\begin{aligned}
M_{-1} &= (A + (A + 1 \mod 2), B + (B + 1 \mod 2), C + (C + 1 \mod 2)),\\
M_0 &= (A + (A \mod 2), B + (B \mod 2), C + (C \mod 2)),\\
M_1 &= (A + (A + D_1 \mod 2), B + (B + D_1 + 1 \mod 2), C + (C + D_1 + 1 \mod 2)),\\
M_2 &= (A + (A + D_2 + 1 \mod 2), B + (B + D_2 \mod 2), C + (C + D_2 + 1 \mod 2)),\\
M_3 &= (A + (A + D_3 + 1 \mod 2), B + (B + D_3 + 1 \mod 2), C + (C + D_3 \mod 2)),
\end{aligned}$$

where $(a,b,c) = (\lfloor A/2 \rfloor, \lfloor B/2 \rfloor, \lfloor C/2 \rfloor)$ and (d_1, d_2, d_3) is taken as

$$\begin{array}{cc}
(d_1, d_2, d_3) & \textit{if } (a + A, b + B, c + C) \bmod 2\\
(1,0,0) & (1,0,0)\\
(0,1,1) & (0,1,1)\\
(0,1,0) & (0,1,0)\\
(1,0,1) & (1,0,1)\\
(0,0,1) & (0,0,1)\\
(1,1,0) & (1,1,0)\\
(D_1, D_2, D_3) & (0,0,0)\\
(1 - D_1, 1 - D_2, 1 - D_3) & (1,1,1).
\end{array}$$

Specially, for arbitrary D_1, D_2, D_3 let $C(\{D_i\}; 0, 0, 0)$ be the union of the sets

$$\begin{aligned} S_1 = \{&(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,-1,0), (1,0,-1), (0,1,-1),\\ &(1,1,-1)\}\\ S_2 = \{&(1,1,0), (1,0,1), (0,1,1), (1,-1,1), (-1,1,1)\} \end{aligned}$$

where S_2 can be computed from S_1.

Remark 1. The elements in S_2 can be computed from S_1. For example, $(1,1,0) = (1,0,0) + (0,1,0)$ with difference $(1,-1,0)$, and $(1,-1,1) = (1,0,0) + (0,-1,1)$ with difference $(1,1,-1)$, where $(1,0,0), (0,1,0), (1,-1,0), (0,-1,1)$ (i.e. (0,1,-1)) and $(1,1,-1)$ all belong to S_1.

Proposition 1. *The chain $C(\{D_i\}_{i=1}^3; A, B, C)$ defined above is a uniform DAC, starting from the set $S_1 \cup S_2$.*

Proof. Firstly, we should note that the chain $C(\{D_i\}; A, B, C)$ always contains $(1,1,1)$ when $(A, B, C) \neq (0,0,0)$, since $M_{-1} = (1,1,1)$ is contained in the chain for any $(A, B, C) \in \{0,1\}^3 \setminus (0,0,0)$. And $(1,1,1)$ is the addition of $(1,1,0)$ and $(0,0,1)$ whose difference is contained in S_1.

M_{-1} is the element of type (odd, odd, odd), and is equal to $(2a+1, 2b+1, 2c+1)$ for $(a,b,c) = (\lfloor A/2 \rfloor, \lfloor B/2 \rfloor, \lfloor C/2 \rfloor)$, which is also equal to the addition of any dual pair in the set $S(a,b,c)$, e.g. $(a,b,c) + (a+1, b+1, c+1)$. Note that there are four dual pairs in $S(a,b,c)$, and the definition says at least one complete dual pair is reserved in $C(\{d_i\}_{i=1}^3; a, b, c)$, whose difference is equal to $(1,1,1)$ or contained in S_1 or S_2.

M_0 is the element of type (even, even, even), and is equal to $(2a + 2(A \bmod 2), 2b + 2(B \bmod 2), 2c + 2(C \bmod 2))$, which is a double of the element $(a + (A \bmod 2), b + (B \bmod 2), c + (C \bmod 2))$, denoted by V. Note that $V \equiv (a + A, b + B, c + C) \bmod 2$, and in $C(\{d_i\}; a, b, c)$ the omitted elements modulo 2 are

$$T_1 \equiv (d_1 + 1, d_1, d_1) \bmod 2, T_2 \equiv (d_2, d_2 + 1, d_2) \bmod 2, T_3 \equiv (d_3, d_3, d_3 + 1) \bmod 2.$$

When $(a + A, b + B, c + C) \bmod 2 = (0,0,0)$ or $(1,1,1)$, then V is (even,even,even) or (odd, odd, odd) which must be contained in $C(\{d_i\}; a, b, c)$ by definition. Apart from these two cases, we observe that pairs $(d_1, a + A)$, $(d_2, b + B)$ and $(d_3, c + B)$ have the same parity, which implies that V modulo 2 is not equal to any T_i.

When $D_1 = 0$, $M_1 = (2a + 2(A \bmod 2), 2b + 1, 2c + 1)$, which is equal to

$$(a + (A \bmod 2), b, c) + (a + (A \bmod 2), b + 1, c + 1) \tag{4}$$

or

$$(a + (A \bmod 2), b + 1, c) + (a + (A \bmod 2), b, c + 1) \tag{5}$$

Note that their differences are contained in S_2 or S_1, hence in $C(\{b_i\}; a, b, c)$. Since $a + A$ and $d_1 + 1$ always have the opposite parity, then none of these four elements is equal to T_1. Assume one is equal to T_2 (or T_3), for example

$(a+(A \bmod 2), b, c)$. Then b and c have an opposite parity, in other word, $(b+1, c)$ and $(b, c+1)$ have the same parity, which implies that none of the two elements in (5) is equal to T_3 (or T_2). Therefore they belongs to $C(\{d_i\}; a, b, c)$.

When $D_1 = 1$, $M_1 = (2a+1, 2b+2(B \bmod 2), 2c+2(C \bmod 2))$ which is equal to the addition of $(a, b+(B \bmod 2), c+(C \bmod 2))$ and $(a+1, b+(B \bmod 2), c+(C \bmod 2))$. Their difference is $(1, 0, 0)$ contained in S_1. In the case that $(a+A, b+B, c+C) \bmod 2 = (0, 0, 0)$ or $(1, 1, 1)$, none of these two elements is equal to T_2, T_3 or T_1, and therefore they are contained in $C(\{b_i\}; a, b, c)$. Apart from these two cases, since pairs $(d_2+1, b+B)$ and $(d_3+1, c+C)$ always have the opposite parity, then none of them is equal to T_2 or T_3. If $(a, b+(B \bmod 2), c+(C \bmod 2))$ is equal to T_1, then $(b+B, c+C) \bmod 2 = (0, 0)$ or $(1, 1)$. In the former case, by definition $d_1 = 1$ and then $T_1 = (0, 1, 1)$, which is a contradiction. In the latter case, $d_1 = 0$ and then $T_1 = (1, 0, 0)$ which is also a contradiction.

Similarly, we can show that M_2 and M_3 are additions of some elements in $C(\{b_i\}; a, b, c)$ whose differences are already in this chain.

Finally, it is obviously uniform since each iteration contains 1 double and 4 additions where M_0 is the double and others are additions. □

Remark 2. Given a 3-tuple (k_1, k_2, k_3), if we want to compute the above 3-dimensional DAC of (k_1, k_2, k_3), the initial D_1, D_2, D_3 should be taken like this:

$$\begin{array}{ll} \text{if } (k_1,k_2,k_3) \bmod 2 = (1,0,0) \text{ or } (0,1,1), & D_1 \equiv k_1 \bmod 2; \text{arbitrary } D_2, D_3 \\ \text{if } (k_1,k_2,k_3) \bmod 2 = (0,1,0) \text{ or } (1,0,1), & D_2 \equiv k_2 \bmod 2; \text{arbitrary } D_1, D_3 \\ \text{if } (k_1,k_2,k_3) \bmod 2 = (0,0,1) \text{ or } (1,1,0), & D_3 \equiv k_3 \bmod 2; \text{arbitrary } D_1, D_2 \\ \text{if } (k_1,k_2,k_3) \bmod 2 = (0,0,0) \text{ or } (1,1,1), & \text{arbitrary } D_1, D_2, D_3 \end{array}$$

Example 1. Here is a simple example of computing the above 3-dimensional DAC of $(199, 331, 513)$, where we set the initial (D_1, D_2, D_3) to be $(1, 1, 1)$.

Stage	M_{-1}	M_0	M_1	M_2	M_3
1		Pre-computation of S_1 and S_2			
2	$(1,1,1)$	$(0,0,2)$	$(0,1,1)$	$(0,1,2)$	$(0,0,1)$
3	$(1,1,3)$	$(0,2,2)$	$(1,2,2)$	$(0,1,2)$	$(1,1,2)$
4	$(1,3,5)$	$(2,2,4)$	$(2,3,5)$	$(2,3,4)$	$(1,3,4)$
5	$(3,5,9)$	$(4,6,8)$	$(3,6,8)$	$(4,5,8)$	$(3,5,8)$
6	$(7,11,17)$	$(6,10,16)$	$(6,11,17)$	$(6,11,16)$	$(7,11,16)$
7	$(13,21,33)$	$(12,20,32)$	$(12,21,33)$	$(12,21,32)$	$(13,21,32)$
8	$(25,41,65)$	$(24,42,64)$	$(25,42,64)$	$(24,41,64)$	$(25,41,64)$
9	$(49,83,129)$	$(50,82,128)$	$(50,83,129)$	$(50,83,128)$	$(49,83,128)$
10	$(99,165,257)$	$(100,166,256)$	$(100,165,257)$	$(99,166,257)$	$(100,166,257)$
11	$(199,331,513)$	$(200,332,514)$	$(199,332,514)$	$(200,331,514)$	$(200,332,513)$

In this chain, for example, the (D_1, D_2, D_3) in Stage 10 is $(0, 0, 1)$, and hence the omitted elements are $\{$(odd,even,even),(even,odd,even),(odd,odd,even)$\} = \{(99, 166, 256), (100, 165, 256), (99, 165, 256)\}$. The five elements in this stage are additions of the elements in Stage 9, e.g. $(99, 165, 257) = (49, 83, 129) + (50, 82, 128)$ with difference $(-1, 1, 1)$ belonging to $S_2, (100, 166, 256) = 2 \cdot$

$(50, 83, 128)$ with difference $(0, 0, 0)$ belonging to $S_1, (100, 165, 257) = (50, 82, 128) + (50, 83, 129)$ with difference $(0, 1, 1)$ belonging to S_2.

4.2 A Faster Three-Dimensional Differential Addition Chain

Now we introduce a new construction of 3-dimensional DAC, which is not uniform but faster than the uniform one described in Sect. 4.1. It only needs four additions or three additions together with one double in each stage of iteration.

Let S be the set defined in Sect. 4.1. We classify part of the sets of four elements of S. We call the set

$$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$$

is of type C_0; the set

$$\begin{aligned} &\{(1,0,0),(0,1,0),(0,0,1),(1,1,0)\}\\ \text{or } &\{(1,0,0),(0,1,0),(0,0,1),(1,0,1)\}\\ \text{or } &\{(1,0,0),(0,1,0),(0,0,1),(0,1,1)\}\end{aligned}$$

is of type C_1; the set

$$\begin{aligned} &\{(1,0,0),(1,1,0),(1,0,1),(0,1,1)\}\\ \text{or } &\{(0,1,0),(1,1,0),(1,0,1),(0,1,1)\}\\ \text{or } &\{(0,0,1),(1,1,0),(1,0,1),(0,1,1)\}\end{aligned}$$

is of type C_2; the set

$$\begin{aligned} &\{(1,0,0),(0,1,0),(1,1,0),(1,0,1)\} \text{ or } \{(1,0,0),(0,0,1),(1,1,0),(1,0,1)\}\\ \text{or } &\{(1,0,0),(0,1,0),(1,1,0),(0,1,1)\} \text{ or } \{(0,1,0),(0,0,1),(1,1,0),(0,1,1)\}\\ \text{or } &\{(1,0,0),(0,0,1),(1,0,1),(0,1,1)\} \text{ or } \{(0,1,0),(0,0,1),(1,0,1),(0,1,1)\}\end{aligned}$$

is of type C_3; the set

$$\{(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$$

is of type C_4.

We use a 3-tuple (a, b, c) of non-negative integers to label each stage of our iteration. In the stage of any given (a, b, c) which is not equal to $(0, 0, 0)$, we compute four elements in $(a, b, c) + S$, of which the set is denoted by $E(a, b, c)$. It has the following form

$$E(a, b, c) = (a, b, c) + \delta,$$

where $\delta \subseteq S$ is of some type C_i for $i = 0, 1, 2, 3$ or 4.

Assume that (a, b, c) is the former stage of (A, B, C), where $a = \lfloor A/2 \rfloor$, $b = \lfloor B/2 \rfloor$, $c = \lfloor C/2 \rfloor$. And assume that $(a, b, c) \neq (0, 0, 0)$. We show in the following that in any case, there exists some set δ of type C_0, C_1, C_2, C_3 or C_4, such that $E(A, B, C)$ can be computed from $(a, b, c) + \delta$. Then we set it to be $E(a, b, c)$. In other word, every element of $E(A, B, C)$ is an addition of two elements of $E(a, b, c)$.

Assume that

$$E(A, B, C) = (A, B, C) + \varDelta, \text{ where } \varDelta \text{ is of type } C_i.$$

We denote by $t = (A, B, C) - (2a, 2b, 2c)$. Then t is an element of S.

If $i = 0$, i.e. $\varDelta = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}$, and when $t \in \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}$, we can set

$$\delta = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\},$$

which is of type C_0. We can check, for example, when $t = (0, 0, 1)$, then

$$E(A, B, C) = (2a, 2b, 2c) + \{(0, 0, 1), (1, 0, 1), (0, 1, 1), (0, 0, 2)\}$$

can be computed from $E(a, b, c) = (a, b, c) + \delta$ by either additions or double. When $t \in \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\}$, we can set

$$\delta = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\} \cup \{t\},$$

which is of type C_1. Also we can check, for example when $t = (1, 0, 1)$, then

$$E(A, B, C) = (2a, 2b, 2c) + \{(1, 0, 1), (2, 0, 1), (1, 1, 1), (1, 0, 2)\}$$

can be computed form $E(a, b, c) = (a, b, c) + \delta$ by additions. When $t = (1, 1, 1)$, we can set

$$\delta = \{(1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1)\},$$

which is of type C_2. In conclusion, we can use the following table to present the δ's in all cases.

$\varDelta = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}$		
t	δ	Type
$(0, 0, 0)$	$\varDelta$	C_0
$(1, 0, 0)/(0, 1, 0)/(0, 0, 1)$	$\varDelta$	C_0
$(1, 1, 0)/(1, 0, 1)/(0, 1, 1)$	$\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\} \cup \{t\}$	C_1
$(1, 1, 1)$	$\{(1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1)\}$	C_2

If $i = 1$, i.e. $\varDelta$ is of type C_1, assume that $\varDelta = \{(1, 0, 0), (0, 1, 0), (0, 0, 1), s\}$ where $s \in \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\}$. Then we can check

Here $s \cdot t$ is the vector with multiplying components, and $s^{\perp}$ is the vector $(1, 1, 1) - s$, also called the dual vector of s as in Sect. 4.1. Here we only check the case when $t \in \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\} \setminus \{s\}$. Then $E(A, B, C) = (2a, 2b, 2c) + t + \varDelta$ and $E(a, b, c) = (a, b, c) + \delta$. Hence we only need to show that

$$t + \varDelta = \{t + (1, 0, 0), t + (0, 1, 0), t + (0, 0, 1), t + s\}$$

$\Delta = \{(1,0,0),(0,1,0),(0,0,1),s\}$		
t	δ	Type
$(0,0,0)$	$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	C_0
$(1,0,0)/(0,1,0)/(0,0,1)$	Δ	C_1
s	Δ	C_1
$(s \neq) t \in \{(1,1,0),(1,0,1),(0,1,1)\}$	$\{s \cdot t, s^{\perp}, s, t\}$	C_3
$(1,1,1)$	$\{(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$	C_4

can be computed from $\delta = \{s \cdot t, s^{\perp}, s, t\}$.Obviously, $s \cdot t \neq s^{\perp}$ and both belong to $\{(1,0,0),(0,1,0),(0,0,1)\}$. The remaining element of $\{\{(1,0,0),(0,1,0),(0,0,1)\} \setminus \{s \cdot t, s^{\perp}\}$ can be expressed as $(1,1,1) - s \cdot t - s^{\perp}$, denoted by α. Since $\alpha \cdot t = (s - s \cdot t) \cdot t = s \cdot t - s \cdot t^2 = 0$ and only one component of α is non-zero, it implies that α is dual to t. Then $t + \alpha = (1,1,1) = s^{\perp} + s$ and hence every element of $E(A,B,C)$ is an addition of two elements of $E(a,b,c)$.

If $i = 2$, i.e. Δ is of type C_2, assume that $\Delta = \{s,(1,1,0),(1,0,1),(0,1,1)\}$ where $s \in \{(1,0,0),(0,1,0),(0,0,1)\}$. Also we can check the following

$\Delta = \{s,(1,1,0),(1,0,1),(0,1,1)\}$		
t	δ	Type
$(0,0,0)$	$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	C_0
s	Δ	C_2
$(s \neq) t \in \{(1,0,0),(0,1,0),(0,0,1)\}$	$\{s, t, s^{\perp}, s+t\}$	C_3
$(1,1,0)/(1,0,1)/(0,1,1)$	Δ	C_2
$(1,1,1)$	$\{(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$	C_4

If $i = 3$, i.e. Δ is of type C_3, we assume that $\Delta = \{s_1 \cdot s_2, k, s_1, s_2\}$ where $s_1 \neq s_2 \in \{(1,1,0),(1,0,1),(0,1,1)\}$ and $k = s_1^{\perp}$ or $s_2^{\perp}$. We can check

$\Delta = \{s_1 \cdot s_2, k, s_1, s_2\}$		
t	δ	Type
$(0,0,0)$	$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$	C_0
$s_1 \cdot s_2$ or k	Δ	C_3
the other[a]	$\{(1,0,0),(0,1,0),(0,0,1)\} \cup \{k^{\perp}\}$	C_1
s_1 or s_2	Δ	C_3
the other[b]	$\{k\} \cup \{(1,1,0),(1,0,1),(0,1,1)\}$	C_2
$(1,1,1)$	$\{(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$	C_4

[a] The remainder in $\{(1,0,0),(0,1,0),(0,0,1)\}$ that is not equal to $s_1 \cdot s_2$ or k.

[b] The remainder in $\{(1,1,0),(1,0,1),(0,1,1)\}$ that is not equal to s_1 or s_2.

If $i = 4$, i.e. $\Delta = \{(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$, we can check

$\Delta = \{(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$		
t	δ	Type
$(0,0,0)$	$\{(1,0,0),(0,1,0),(0,0,1),(1,1,0)\}$	C_1
$(1,0,0)/(0,1,0)/(0,0,1)$	$\{t\} \cup \{(1,1,0),(1,0,1),(0,1,1)\}$	C_2
$(1,1,0)/(1,0,1)/(0,1,1)$	Δ	C_4
$(1,1,1)$	Δ	C_4

In conclusion, we have the following

Proposition 2. *In the stage of* $(0,0,0)$, *we take* $E(0,0,0) = S_1 \cup S_2$ *where* S_1, S_2 *are defined in Definition 1. For any given 3-tuple* (k_1,k_2,k_3) *of non-negative scalars, let* $(A,B,C) = (k_1,k_2,k_3)$ *and* Δ *be the set of four elements of type* C_0*. Then compute recursively the whole chain, denoted by* $C(k_1,k_2,k_3)$*, using the above method. Then* $C(k_1,k_2,k_3)$ *is a DAC containing* (k_1,k_2,k_3)*.*

Proof. First, $(k_1,k_2,k_3) \in E(A,B,C) \subseteq C(k_1,k_2,k_3)$. Second, from the above procedure we see that every element (except the elements in the stage of $(0,0,0)$) in this chain is an addition of two former elements (may be the same) in this chain, of which the difference belongs to $S_1 \cup S_2$. Therefore, $C(k_1,k_2,k_3)$ is a DAC. □

Example 2. Take $(k_1,k_2,k_3) = (9,10,11)$. They all have 4 bits. Set $(A_4,B_4,C_4) = (9,10,11)$ and $\Delta_4 = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$ of type C_0. Then

$$E_4 = \{(9,10,11),(10,10,11),(9,11,11),(9,10,12)\}.$$

Set $(A_3,B_3,C_3) = (4,5,5)$. In this case $t = (1,0,1)$ and by table lookup $\Delta_3 = \{(1,0,0),(0,1,0),(0,0,1),(1,0,1)\}$ is of type C_1. Then

$$E_3 = \{(5,5,5),(4,6,5),(4,5,6),(5,5,6)\}.$$

Set $(A_2,B_2,C_2) = (2,2,2)$. In this case $t = (0,1,1)$ and by table lookup $\Delta_2 = \{(0,1,0),(0,0,1),(1,0,1),(0,1,1)\}$ is of type C_3. Then

$$E_2 = \{(2,3,2),(2,2,3),(3,2,3),(2,3,3)\}.$$

Set $(A_1,B_1,C_1) = (1,1,1)$. In this case $t = (0,0,0)$ and by table lookup $\Delta_1 = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$ is of type C_0. Then

$$E_1 = \{(1,1,1),(2,1,1),(1,2,1),(1,1,2)\}.$$

Finally $(A_0,B_0,C_0) = (0,0,0)$ and $t = (1,1,1)$, and then $\Delta_0 = \{(1,0,0),(1,1,0),(1,0,1),(0,1,1)\}$. Then

$$E_0 = \{(1,0,0),(1,1,0),(1,0,1),(0,1,1)\}.$$

If we denote by $E_i[j]$ the j-th element of E_i, then we have the relations:

$$\begin{array}{l} E_4[1] = E_3[1] + E_3[3],\ E_3[1] = E_2[1] + E_2[3],\ E_2[1] = E_1[1] + E_1[3],\ E_1[1] = E_0[1] + E_0[4] \\ E_4[2] = E_3[1] + E_3[4],\ E_3[2] = E_2[1] + E_2[4],\ E_2[2] = E_1[1] + E_1[4],\ E_1[1] = E_0[2] + E_0[3] \\ E_4[3] = E_3[2] + E_3[4],\ E_3[3] = E_2[2] + E_2[4],\ E_2[3] = E_1[2] + E_1[3],\ E_1[1] = E_0[2] + E_0[4] \\ E_4[4] = E_3[3] + E_3[4],\ E_3[4] = E_2[3] + E_2[4],\ E_2[4] = E_1[3] + E_1[4],\ E_1[1] = E_0[3] + E_0[4] \end{array}.$$

5 Comparison

Recall the settings described in Sects. 2 and 3. To compute $x(kP)$, now we can apply different dimensional DACs to different models: Eqs. (2) and (3) or $x(kP)$ directly. In this part, we analyze the performance of scalar multiplication $x(kP)$ utilizing the following four kinds of DACs: 1-dimensional Montgomery ladder (Ladder for short), 2-dimensional DJB chains (DJB), the 3-dimensional uniform differential addition chains (3-Uni.) and the 3-dimensional non-uniform one (3-Non-uni) described in Sect. 4.

5.1 Theoretic Analysis

For an elliptic curve in short Weiestrass form, using the projective coordinates, a general addition (write **A** for short) involves 12 field multiplications (12M) and 2 field squarings (2S), whereas a double (**D**) involves 7M+5S. If we restrict to the operations on x-line, then a pseudo-addition (**PA**) needs 9M+2S and a pseudo-double (**PD**) needs 6M+3S [5, Chap. 13]. Assume that the target group is of 256-bit security level, and the scalar k is a 256-bit number. Then the following table shows the theoretic field operations needed by those five DACs.

Table 1. Theoretical estimate

chain	dim.	rounds	#operations/bit	pre-comp.	#total operations
Ladder	1	256	1**PD** + 1**PA**	0	3840M + 1280S
DJB	2	~ 128	1**PD** + 2**PA**	1**A**	3084M + 898S
3-Uni.	3	~ 85	1**PD** + 4**PA**	4**A** + 5**PA**	3663M + 953S
3-Non-uni.	3	~ 85	$\leq$ 4**PA**	4**A** + 5**PA**	3153M + 698S

Here we note two things. First if we take 1S = 0.8M, our 3-dimensional non-uniform DAC needs 3711.4M, less than 3802.4M of the DJB chain. Second our 3-dimensional non-uniform DAC and the DAC in [11] taking $d = 3$ need almost the same number of field operations in each iteration and precomputation.

5.2 Implementation Results

We make the following experiments with computer algebra package MAGMA. We choose two families of elliptic curves E_1' and E_2' that are quadratic twists of GLV curves chosen from [12]. Let p be a prime. For a non-square element u in $\mathbb{F}_{p^2}$, E_1' and the two efficient $\mathbb{F}_{p^2}$-endomorphisms on the x-line are given by

$$E_1' : y^2 = x^3 - \frac{3}{4}ux^2 - 2u^2x - u^3, \quad \Phi_x(x) = \frac{x^2 - u^2\zeta}{\zeta^2(x - au)}, \quad \Psi_x(x) = u^{1-p}x^p.$$

Their characteristic polynomials are $\Phi^2 - \Phi + 2 = 0$, $\Psi^2 + 1 = 0$. Moreover take $p \equiv 1 \pmod 3$ and $\gamma \in \mathbb{F}_p$ to be an element of order 3. Then E_2' and its efficient $\mathbb{F}_{p^2}$-endomorphisms on the x-line are given by

$$E_2' : y^2 = x^3 + bu^3, \quad \Phi_x(x) = \gamma x, \quad \Psi_x(x) = u^{1-p}x^p$$

with $\Phi^2 + \Phi + 1 = 0$, $\Psi^2 + 1 = 0$. For each $i = 1, 2$, and each of $64, 128$ and 192 bit lengths of primes p, we choose 20 p's such that each $E_i'(\mathbb{F}_{p^2})$ is almost prime with the large prime subgroup of order n, and hence we obtain 20 target elliptic curves. On each curve, we randomly choose 20 pairs (k, P)'s where P is a rational point of the curve of order n and k is a scalar having the same bit length as n. We test the above four DACs in the computation of $x(kP)$. And for each (k, P), we make the following three sets of comparison experiments: 3-Uni versus Ladder, 3-Non-uni. versus Ladder and 3-non-uni. versus DJB[3]. We record their running times and take the average over the $20 \times 20 = 400$ data at each bit length of $64, 128$ and 192. Then we compute the ratio of speedup of the former DAC relative to the latter one. The implementation results are listed in Tables 1 and 2.

Table 2. The ratio of speedup tested in E_1'

p	64-bit	128-bit	192-bit
n	$125 \sim 126$-bit	$253 \sim 254$-bit	$381 \sim 382$-bit
3-Uni. vs Ladder	6.8%	11.2%	11.5%
3-Non-uni. vs Ladder	24.7%	28.7%	29.3%
3-Non-uni. vs DJB	3.9%	8.5%	9.7%

Table 3. The ratio of speedup tested in E_2'

p	64-bit	128-bit	192-bit
n	$125 \sim 126$-bit	$253 \sim 254$-bit	$381 \sim 382$-bit
3-Uni. vs Ladder	4.3%	7.0%	7.4%
3-Non-uni. vs Ladder	22.1%	24.2%	25.3%
3-Non-uni. vs DJB	3.0%	5.7%	6.8%

From the tables, first, we recognise that the performance depends on the GLV model that we choose. However, compared with straightforward computing with Ladder, the triple scalar multiplication using our 3-dimensional DACs runs faster evidently. Moreover, the triple scheme using 3-Non-uni. outperforms the double one using DJB chains. Second, the ratio of speedup increases when the

[3] In the two-dimensional GLV, we always utilize the endomorphism Ψ in the testing.

bit length of p grows. The case of 128-bit or equivalently when the security level is about 254-bit may provide more significant reference. In this case, by taking an average of the values in two tables, the triple scheme using 3-Uni. runs about 9% faster than straightforward computing; and that using 3-Non-uni. runs about 26% faster than straightforward computing, and about 6% faster than the double scheme using DJB chains (Table 3).

6 Conclusion and Future Research

We proposed two constructions of 3-dimensional DACs. One is a straightforward extension of the DJB chain to higher dimensional case, and it inherits the property of uniformity. The other is not uniform but runs faster than the uniform one as saving one more point operation in each iteration. On the quadratic twists of GLV curves, we implemented scalar multiplication on the x-coordinate systems, utilizing 1 to 3-dimensional GLV methods with corresponding dimensional DACs. Experiments show that at about 254-bit security level, the triple scalar multiplication using our uniform DACs runs about 9% faster than straightforward computing using Montgomery ladder; and that using our non-uniform DACs runs about 26% faster than that using Montgomery ladder, and about 6% faster than the double scalar multiplication using DJB chains.

As it is pointed out that, it would be more meaningful and we will consider to implement our higher dimensional DACs on the x-coordinate systems of some more advanced curves, such as the complete twisted Edwards curve described in [7], together with optimizing formulas for differential point tripling (or quadrupling) in various forms of curves. The 4-dimensional DACs deserve to be studied as well, since triple scalar multiplication on the x-coordinate performs well and many advanced elliptic curves are originally tailored for 4-dimensional GLV method. We considered the 4-dimensional extension of the uniform case, but found it inefficient for its heavy pre-computation and excessive operations in each iteration, see Appendix A. However, the non-uniform case may provide more possibilities. We leave these topics for future research.

Acknowledgement. We would like to thank Yuqing Zhu for his kind advice and selfless help on the first version of this work. And we would like to thank the anonymous reviewers for their detailed comments and suggestions. This work is supported by National Natural Science Foundation of China (Grant No. 61872359).

A Four-dimensional Case

If we consider further the straightforward 4-dimensional extension of DJB chains, we found that in each iteration we should compute $2 + (2^4 - 2)/2 = 9$ elements, containing 1 double and 8 additions, which is rather expensive and hence has no practical usage. For completeness, in this part we give its definition and a simple example. Its complex proof of correctness has been done by authors and one can also check it by computers.

For brief of notation we let $e_1 = (1,0,0,0), e_2 = (0,1,0,0), e_3 = (0,0,1,0)$, $e_4 = (0,0,0,1), e_5 = (1,1,0,0), e_6 = (1,0,1,0), e_7 = (1,0,0,1)$. Denote by n^4 the 4-tuple (n,n,n,n). Then the 7 elements omitted from $S(a,b,c,d)$[4] can be described as $T_i = (a,b,c,d) + (U_i \bmod 2)$ where $U_i = (a,b,c,d) + f_i^4 + e_i$ and $f_i \in \{0,1\}$ for $i = 1, \cdots, 7$.

Definition 2. *For a given 4-tuple of nonnegative integers (A,B,C,D) and the set $\{F_1, \cdots, F_7\}$ where $F_i \in \{0,1\}, i = 1, \cdots, 7$, the chain $C(\{F_i\}_{i=1}^7; A, B, C, D)$ is defined recursively, as the set $C(\{f_i\}_{i=1}^7; a,b,c,d)$ added with the following nine elements:*

$$\begin{aligned} M_{-1} &= (A,B,C,D) + ((A+1,\ B+1,\ C+1,\ D+1) \bmod 2), \\ M_0 &= (A,B,C,D) + ((A,\qquad B,\qquad C,\qquad D) \bmod 2), \end{aligned}$$

and for $i = 1, \cdots, 7$,

$$M_i = (A,B,C,D) + (N_i \bmod 2) \quad \text{where } N_i = (A,B,C,D) + (F_i+1)^4 + e_i.$$

Here $(a,b,c,d) = (\lfloor A/2 \rfloor, \lfloor B/2 \rfloor, \lfloor C/2 \rfloor, \lfloor D/2 \rfloor)$ and $(f_1, \cdots, f_7)$ is taken as

$(f_1, \cdots, f_7)$	$if(a+A, b+B,$ $c+C, d+D) \bmod 2$	$(f_1, \cdots, f_7)$	$if(a+A, b+B,$ $c+C, d+D) \bmod 2$
$(1,0,0,0,1,1,1)$	$(1,0,0,0)$	$(0,1,1,1,0,0,0)$	$(0,1,1,1)$
$(0,1,0,0,1,0,0)$	$(0,1,0,0)$	$(1,0,1,1,0,1,1)$	$(1,0,1,1)$
$(0,0,1,0,0,1,0)$	$(0,0,1,0)$	$(1,1,0,1,1,0,1)$	$(1,1,0,1)$
$(0,0,0,1,0,0,1)$	$(0,0,0,1)$	$(1,1,1,0,1,1,0)$	$(1,1,1,0)$
$(1,1,0,0,1,0,0)$	$(1,1,0,0)$	$(0,0,1,1,0,0,0)$	$(0,0,1,1)$
$(1,0,1,0,0,1,0)$	$(1,0,1,0)$	$(0,1,0,1,0,0,0)$	$(0,1,0,1)$
$(1,0,0,1,0,0,1)$	$(1,0,0,1)$	$(0,1,1,0,0,0,0)$	$(0,1,1,0)$
$(F_1, F_2, F_3, F_4, F_5, F_6, F_7)$	$(0,0,0,0)$	$(1-F_1, 1-F_2, 1-F_3, 1-F_4, 1-F_5, 1-F_6, 1-F_7)$	$(1,1,1,1)$.

Specially, for arbitrary $F_1, \cdots, F_7$, let $C(\{F_i\}; 0,0,0,0)$ be the union of the sets

$$\begin{aligned} S_1 = \{&(0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), \\ &(1,-1,0,0), (1,0,-1,0), (1,0,0,-1), (0,1,-1,0), (0,1,0,-1), (0,0,1,-1), \\ &(1,1,-1,0), (1,1,0,-1), (1,0,1,-1), (0,1,1,-1), \\ &(1,1,1,-1)\} \\ S_2 = \{&(1,1,0,0), (1,0,1,0), (1,0,0,1), (0,1,1,0), (0,1,0,1), (0,0,1,1), \\ &(1,1,1,0), (1,1,0,1), (1,0,1,1), (0,1,1,1), \\ &(1,-1,1,0), (1,-1,0,1), (1,0,-1,1), (0,1,-1,1), \\ &(-1,1,1,0), (-1,1,0,1), (-1,0,1,1), (0,-1,1,1), \\ &(1,1,-1,1), (1,-1,1,1), (-1,1,1,1), \\ &(1,1,-1,-1), (1,-1,1,-1), (1,-1,-1,1)\} \end{aligned}$$

where S_2 can be computed from S_1.

[4] As the analogous symbol used in Sect. 4.1, $S(a,b,c,d)$ is the set $(a,b,c,d) + \{0,1\}^4$.

We find that as the dimension of the chain increases, the pre-computation part becomes a heavy burden, and it grows exponentially w.r.t. the dimension. In some situation, this maybe a main disadvantage of computing scalar multiplication using higher dimensional DACs.

Example 3. Given a simple 4-tuple $(10, 9, 8, 7)$. The uniform 4-dimensional DAC of $(10, 9, 8, 7)$ is: $S_1 \cup S_2 \cup S_3$ where $S_3 =$

$$\{(1,1,1,1),(2,2,2,0),(2,1,1,1),(2,1,2,0),(2,2,1,0),(2,2,2,1),(2,2,1,1),(2,1,2,1),(2,1,1,0),$$
$$(3,3,3,1),(2,2,2,2),(3,2,2,2),(3,2,3,1),(3,3,2,1),(3,3,3,2),(3,3,2,2),(3,2,3,2),(3,2,2,1),$$
$$(5,5,5,3),(6,4,4,4),(5,4,4,4),(6,5,4,4),(5,5,4,3),(5,5,5,4),(5,5,4,4),(5,4,5,4),(5,4,4,3),$$
$$(11,9,9,7),(10,10,8,8),(10,9,9,7),(11,10,9,7),(11,9,8,7),(11,9,9,8),(10,10,9,7),(10,9,8,7),$$
$$(10.9.9.8)\}$$

References

1. Akishita, T.: Fast simultaneous scalar multiplication on elliptic curve with montgomery form. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 255–267. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45537-X_20
2. Azarderakhsh, R., Karabina, K.: A new double point multiplication algorithm and its application to binary elliptic curves with endomorphisms. IEEE Trans. Comput. **63**(10), 2614–2619 (2014)
3. Bernstein, D.J.: Differential addition chains. Technical Report (2006). http://cr.yp.to/ecdh/diffchain-20060219.pdf
4. Brown, D.R.: Multi-dimensional montgomery ladders for elliptic curves. Cryptology ePrint Archive, Report 2006/220 (2006). https://eprint.iacr.org/2006/220
5. Cohen, H., et al.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press (2005)
6. Costello, C., Hisil, H., Smith, B.: Faster compact diffie–hellman: endomorphisms on the x-line. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 183–200. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_11
7. Costello, C., Longa, P.: Four$\mathbb{Q}$: four-dimensional decompositions on a $\mathbb{Q}$-curve over the mersenne prime. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 214–235. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_10
8. Galbraith, S.D., Lin, X., Scott, M.: Endomorphisms for faster elliptic curve cryptography on a large class of curves. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 518–535. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_30
9. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 190–200. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_11
10. Zhi, H., Longa, P., Maozhi, X.: Implementing the 4-dimensional GLV method on GLS elliptic curves with j-invariant 0. Des. Codes Crypt. **63**(3), 331–343 (2012)
11. Hutchinson, A., Karabina, K.: Constructing multidimensional differential addition chains and their applications. J. Cryptographic Eng. 1–19 (2017)

12. Longa, P., Sica, F.: Four-dimensional gallant-lambert-vanstone scalar multiplication. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 718–739. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_43
13. Montgomery, P.L.: Evaluating recurrences of form $X_{m+n} = f(X_m, X_n, X_{m-n})$ via lucas chains (1983). ftp.cwi.nl:/pub/pmontgom/lucas.ps.gz
14. Montgomery, P.L.: Speeding the pollard and elliptic curve methods of factorization. Math. Comput. **48**(177), 243–264 (1987)
15. Rao, S.R.S.: A note on Schoenmakers algorithm for multi exponentiation. In: 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), vol. 4, pp. 384–391 (2015)
16. Subramanya Rao, S.R.: Three dimensional montgomery ladder, differential point tripling on montgomery curves and point quintupling on weierstrass' and edwards curves. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 84–106. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31517-1_5
17. Stam, M.: Speeding up subgroup cryptosystems. Technische Universiteit Eindhoven (2003)
18. Yi, H., Zhu, Y., Lin, D.: Refinement of the four-dimensional GLV method on elliptic curves. In: Adams, C., Camenisch, J. (eds.) SAC 2017. LNCS, vol. 10719, pp. 23–42. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72565-9_2