# Generation of two "independent" points on an elliptic curve of $j$-invariant $\neq 0, 1728$

Dmitrii Koshelev[0000−0002−4796−8989]
dimitri.koshelev@gmail.com

Parallel Computation Laboratory, École Normale Supérieure de Lyon, France
http://www.ens-lyon.fr/en

**Abstract.** This article is dedicated to a new generation method of two "independent" $\mathbb{F}_q$-points $P_0$, $P_1$ on almost any ordinary elliptic curve $E$ over a finite field $\mathbb{F}_q$ of large characteristic. In particular, the method is relevant for all standardized and real-world elliptic curves of $j$-invariants different from 0, 1728. The points $P_0$, $P_1$ are characterized by the fact that nobody (even a generator) knows the discrete logarithm $\log_{P_0}(P_1)$ in the group $E(\mathbb{F}_q)$. Moreover, only one square root extraction in $\mathbb{F}_q$ (instead of two ones) is required in comparison with all previous generation methods.

**Keywords:** endomorphism rings · generation of "independent" points · isotrivial elliptic curves · Mordell–Weil lattices.

## 1 Introduction

There is a misconception among many academics that elliptic curve cryptography (ECC) finally and irrevocably gives way to post-quantum cryptography (PQC). This is because academia is mainly funded by governments. They are really interested in moving to PQC in the near future, since government information must remain classified for a long time. At the same time, according to a series of experts, a powerful quantum computer may be invented already in our lifetime.

However, ECC in fact experiences an all-time flourishing due to its active application to protect the majority of blockchains, including cryptocurrencies. First, they are inherently opposed to state control, hence any standardized cryptography is foreign for them. Second, their emphasis on elliptical cryptography stems from concerns about efficiency of cryptographic protocols. Blockchains use multi-party computation (MPC) with plenty of parties, so the transition to PQC would lead to a catastrophic slowdown.

Unfortunately, actors of the blockchain world are greedy enough to fund research even if it is directly related to practice. They are rather focused on urgent issues of software implementation or protocol design. In this connection, blockchain enthusiasts and applied mathematicians rarely communicate with each other. That is why the latter are often unfamiliar with needs that arise in today's elliptic cryptography. And in vain, because it is based on interesting sections of the theory of elliptic curves. Meanwhile, developers do not have enough time, desire, and skills to speed up ECC by mathematical methods, and not by program optimizations.

This article takes one more small step towards integrating the two professional communities. The role of the connecting thread is performed by the task of generating transparently several *"independent" $\mathbb{F}_q$-points* on an elliptic curve $E$ over a finite field $\mathbb{F}_q$. "Independence" means that non-trivial linear relations between the points are unknown to anyone. We will focus on the case when the characteristic of $\mathbb{F}_q$ is large and $E: y^2 = f(x)$ is an ordinary (a.k.a. non-supersingular) curve. Today, other pairs $E/\mathbb{F}_q$ are considered to be suspicious and hence they are (almost) never used.

At the moment, in order to obtain one point in $E(\mathbb{F}_q)$ people frequently give preference to the naive try-and-increment method (represented, e.g., in [15, Section 8.2.1]) iterating the $x$-coordinate. Alternatively, one can resort to any constant-time map to $E(\mathbb{F}_q)$. In recent years, there are breakthroughs in constructing such maps [19, Section 3]. Nevertheless, it is worth stressing that constant-timeness is in fact superfluous in our context, because the seed of a generation process is not secret. On the contrary, it must be as public as possible so that everyone can make sure in honesty of a generator. That is why, we can forget about tricky deterministic maps without loss of generality.

Given $x \in \mathbb{F}_q$, the condition $\sqrt{f(x)} \in \mathbb{F}_q$ can be checked without computing before that the square root, namely via the Legendre symbol $\left(\frac{f(x)}{q}\right)$. Generally speaking, this symbol (with whatever argument from $\mathbb{F}_q$) should be determined by widespread Euclidean-type algorithms running in the bit time $\Theta(\log^2(q))$. Moreover, they prove themselves well in practice [18,26]. Meanwhile, two $x$-coordinates are on average enough to meet the desired condition. As is well known (e.g. from [29]), extracting $\sqrt{\cdot} \in \mathbb{F}_q$ costs $\Theta(\log^3(q))$ bit operations, which is an order of magnitude more expensive than $\left(\frac{\cdot}{q}\right)$. Finally, when one needs more than one $\mathbb{F}_q$-point on $E$, nothing prevents to repeat multiple times the try-and-increment method with another seed.

A novel approach for solving the generation task is suggested in [20]. That article dwells on the extreme, but important case of $j$-invariant 0. The current article follows the same conception, but in the general case, that is, without significant restrictions on $j$-invariant. In contrast to [20], we will succeed in constructing just two "independent" points in $E(\mathbb{F}_q)$. Nonetheless, this already significantly affects performance, taking into account the cumulative effect. Indeed, it seemed earlier that $n \in \mathbb{N}$ "independent" points cannot be generated faster than by finding $n$ roots (usually square) in the field $\mathbb{F}_q$. We will justify that $\lceil n/2 \rceil$ quadratic roots turn out to be sufficient for that purpose.

Throughout the text, $D < 0$ stands for the complex multiplication (CM) discriminant of $E$. As we will see, unlike $j(E) = 0$ (equivalently, $D = -3$), a new difficulty appears whenever $D$ is large by absolute value. Fortunately, we will successfully circumvent this obstacle. It is worth noting that conservative cryptographers prefer and, at the same time, regulators standardize elliptic curves of huge CM discriminants, because they seem more secure as opposed to others. For instance, the source [4] recommends to pick $D = D_0 c^2$ with $c \in \mathbb{N}$ and the square-free natural $-D_0 > 2^{100}$. The given condition is fulfilled for a random curve $E$ with a high probability.

The problem of “independent” points has long painful history. For example, such points occurred in the notorious *Dual Elliptic Curve Deterministic Random Bit Generator (Dual_EC_DRBG)* invented by NIST. It was heavily criticized, because NIST proposed its own points $P$, $Q$ without any transparent explanation of their origin. Therefore, the cryptographic community was suspicious concerning a possible backdoor (see, e.g., [5,6]). In addition, Dual_EC_DRBG had other disadvantages [9,30], which raised even more doubts about the meaning of its use. Under public pressure, the last actual version of the standard [3] does not contain this generator anymore. Nevertheless, as shown in [16], it can be modified to be safe if the discrete logarithm between the points is truly unknown.

It is also worth mentioning a *Password-Authenticated Key Exchange (PAKE)* protocol under the name *SPAKE2* [1,2]. It was one of the candidates (but not the winner) of the relatively recent public selection process [13] under the auspices of CFRG. SPAKE2 likewise requires two “independent” points denoted by $M$, $N$ in the draft [23]. Bearing in mind the bitter experience of Dual_EC_DRBG, the creators of the given draft explicitly indicate where the points come from.

The third classical example is the so-called *Pedersen hash function* to the group $E(\mathbb{F}_q)$ (a.k.a. *Pedersen commitment*) [25, Section 3], which is also contained in [11, Section 3]. The Pedersen hash is naturally generalized to an arbitrary number $n$ of “independent” points (see, e.g., [7]). The larger number $n$, the better compression can be provided by the hash function, since it always returns one $\mathbb{F}_q$-point on $E$ regardless of $n$. Thereby, it is very tempting to take tremendous values of $n$ (such as $2^{28}$), which really happens in practice [10].

Entities of any cryptographic scheme operating with a large number of “independent” points face a difficult choice. They are obliged either to store/transmit the points or to regenerate them (or others) every time from a short seed. The first approach requires large amount of memory or a channel with good bandwidth, respectively. In turn, the second one forces entities to spend a part of their running time on a monotonous point regeneration. Consequently, any noticeable acceleration of the given subroutine deserves attention to lean towards the second solution.

To be honest, implementers of cryptosystems pay little attention to the problem of generating “independent” points, since it is not the bottleneck in contrast to other primitives on elliptic curves such as a *Multi-Scalar Multiplication (MSM)* [14]. By the way, MSM is essentially evaluating the Pedersen hash function. Nevertheless, the problem under consideration should be rigorously discussed by the scientific society. Otherwise, there is a risk that an ad hoc speed up proposed by a junior developer may come to a catastrophic security error not identified until the release of a software product.

The mathematics underlying the new generation method is quite exotic for elliptic cryptography. It is about elliptic curves over the rational function field

$\mathbb{F}_q(u)$ and about their *Mordell–Weil lattices*. The author prefers [31], [32, Chapter III] as sources on the topic. It is hoped that Mordell–Weil lattices will sooner or later find other exciting applications in ECC or PQC, just as this was done in the past by pairings and isogenies, respectively. Such a prospect has a right to exist, because any $\mathbb{F}_q$-isogeny $E_0 \to E_1$ between elliptic $\mathbb{F}_q$-curves is realized as a point of $E_1$ over the function field $\mathbb{F}_q(E_0)$.

## 2 Mathematical preliminaries

Consider $E\colon y^2 = f(x) := x^3 + ax + b$, an elliptic curve over a finite field $\mathbb{F}_q$ of characteristic 5 or greater. Since the curve $E$ is ordinary by our assumption, its endomorphism ring is an order $\mathbb{Z}[\tau] = \mathbb{Z} \oplus \tau\mathbb{Z}$ in some imaginary quadratic field. Throughout the present paper, we will freely use basic facts from the theory of such quadratic orders (see, e.g., [12, Section 7]).

As above, denote by $D < 0$ the complex multiplication discriminant of $E$, that is, the discriminant of $\mathbb{Z}[\tau]$. As is customary, $\widehat{\tau}$ stands for the dual, i.e., the complex conjugate of $\tau$. The integers

$$d := \deg(\tau) = \tau\widehat{\tau}, \qquad t := \mathrm{tr}(\tau) = \tau + \widehat{\tau}$$

are commonly called the degree (or norm) and trace of $\tau$, respectively. Be careful, $t$ is not in general the Frobenius trace of $E$, which is usually associated with this letter in the literature on elliptic curves. Clearly,

$$\tau = \frac{t + \sqrt{D}}{2}, \qquad \widehat{\tau} = \frac{t - \sqrt{D}}{2}$$

are solely the roots of the quadratic polynomial $m_\tau(x) := x^2 - tx + d$ with discriminant $D = t^2 - 4d$. Besides, it will be convenient to have before our eyes the elementary formula

$$\deg(n_0 + n_1\tau) = n_0^2 + n_0 n_1 t + n_1^2 d, \tag{1}$$

where $n_0, n_1 \in \mathbb{Z}$.

Let's extend the field $\mathbb{F}_q$ to the (infinite) function field $F := \mathbb{F}_q(u)$ in one variable $u$. Now, fix the quadratic twist $E^g : g(u)y^2 = f(x)$ by means of a function $g \in F^*$. Certainly, $E^g$ is a meaningless twist if $\sqrt{g} \in F$ because of the isomorphism

$$E^g \to E \qquad (x, y) \mapsto (x, \sqrt{g}\cdot y).$$

Trivially, $j(E^g) = j(E)$ and, vice versa, every elliptic $F$-curve with the given $j$-invariant $\neq 0, 1728$ is isomorphic over $F$ to the twist $E^h$ for some $h \in F^*$. Such curves are said to be *isotrivial*. Thus, without loss of generality, we can deal with the form $E^g$ rather than with the conventional Weierstrass form.

Hereafter, put $\mathcal{E} := E^g$ to simplify the notation. Recall that the (finitely generated) Mordell–Weil group $\mathcal{E}(F)$ is equipped with the so-called *naive height*

$$h\colon \mathcal{E}(F) \to \mathbb{N} \qquad h(P) := \begin{cases} \deg(x) & \text{if} \quad P = (x, y), \\ 0 & \text{if} \quad P = (0:1:0). \end{cases}$$

In particular, $h(P) = h(-P)$. Based on $h$ and on [32, Theorem III.4.3], we get the *canonical height* and *pairing*

$$\begin{aligned} &\widehat{h}\colon \mathcal{E}(F) \to \mathbb{Q}_{\geqslant 0} && \widehat{h}(P) := \frac{1}{2}\lim_{n\to\infty}\frac{h(2^nP)}{4^n}, \\ &\langle\cdot,\cdot\rangle\colon \mathcal{E}(F)^2 \to \mathbb{Q} && \langle P, Q\rangle := \widehat{h}(P+Q) - \widehat{h}(P) - \widehat{h}(Q), \end{aligned} \tag{2}$$

respectively. It is useful to remember that $2\widehat{h}(P) = \langle P, P\rangle$ for each point $P$ and $\widehat{h}(P) = 0$ if and only if the order of $P$ is finite. The fact that $\widehat{h}$ always takes rational values will be at the core of proving Theorem 1. Curiously, as stressed in [32, Remark III.4.3.1], the analogue of $\widehat{h}$ for elliptic curves over number fields (probably) takes transcendental values at all non-torsion points.

Since $\mathcal{E}$ is a twist of $E$, the endomorphism rings of $E$, $\mathcal{E}$ are identical. Therefore, $\mathcal{E}(F)$ is not only a group, but also an $\mathbb{Z}[\tau]$-module.

**Lemma 1.** *For any $\varphi \in \mathbb{Z}[\tau]$ and $P, Q \in \mathcal{E}(F)$ there are the identities*

$$\widehat{h}(\varphi(P)) = \deg(\varphi)\widehat{h}(P), \qquad \langle\varphi(P), \varphi(Q)\rangle = \deg(\varphi)\langle P, Q\rangle.$$

*Proof.* The second identity immediately follows from the first one, hence we can concentrate on establishing it. For compactness, introduce the even function $\varphi_x := x \circ \varphi$ whose $\deg(\varphi_x) = 2\deg(\varphi)$. We lack the generalized naive height

$$h_{\varphi_x}(P) := \deg(\varphi_x(P)) = h(\varphi(P))$$

from [33, Section VIII.6]. In this notation, $h = h_x$. Note that

$$h_{\varphi_x}(2^nP) = h(\varphi(2^nP)) = h(2^n\varphi(P))$$

for each $n \in \mathbb{N}$. By virtue of [33, Proposition VIII.9.1], we have:

$$\widehat{h}(P) = \widehat{h}_{\varphi_x}(P) := \frac{1}{\deg(\varphi_x)}\lim_{n\to\infty}\frac{h_{\varphi_x}(2^nP)}{4^n} = \frac{\widehat{h}(\varphi(P))}{\deg(\varphi)}.$$

The lemma is proved. □

Let's suppose that we are given two short $F$-points $P_i$ on the curve $\mathcal{E}$, where $i \in \mathbb{Z}/2$. "Short" means that their canonical heights $\widehat{h}_i := \widehat{h}(P_i)$ as well as $e := \langle P_0, P_1\rangle$ are pretty small and hence they can be found. Also, we need the values $v_i := \langle P_i, \tau(P_{i+1})\rangle$. Provided that $D$ is large, we cannot directly compute them. Fortunately, one value is expressed through the other.

**Lemma 2.** *There is the linear relation $v_0 + v_1 = te$.*

*Proof.* For the sake of compactness, put $Q_i := P_i + \tau(P_{i+1})$. It is suggested to borrow from [32, Theorem III.4.3.b.iii] the *parallelogram law*

$$\widehat{h}(Q_0 + Q_1) + \widehat{h}(Q_0 - Q_1) = 2\widehat{h}(Q_0) + 2\widehat{h}(Q_1). \tag{3}$$

In accordance with Lemma 1 and the formula (1), we get the sequence of equalities

$$\widehat{h}(Q_0 \pm Q_1) = \widehat{h}\big((1 \pm \tau)(P_0 \pm P_1)\big) = \deg(1 \pm \tau)\widehat{h}(P_0 \pm P_1) = (1 \pm t + d)(\widehat{h}_0 + \widehat{h}_1 \pm e).$$

Thereby, the left-hand side of the law (3) is equal to

$$2(1+d)(\widehat{h}_0 + \widehat{h}_1) + 2te.$$

On the other hand,

$$\widehat{h}(Q_i) = \langle P_i, \tau(P_{i+1})\rangle + \widehat{h}(P_i) + \widehat{h}(\tau(P_{i+1})) = v_i + \widehat{h}_i + d\widehat{h}_{i+1}.$$

As a result, the right-hand side of the law (3) coincides with

$$2(v_0 + v_1) + 2(1+d)(\widehat{h}_0 + \widehat{h}_1).$$

Equating the two sides, we obtain the statement of the lemma. □

It is time to put into play the (symmetric) Gram matrix

$$M_2 := \begin{pmatrix} \langle P_0, P_0\rangle & \langle P_0, P_1\rangle \\ * & \langle P_1, P_1\rangle \end{pmatrix} = \begin{pmatrix} 2\widehat{h}_0 & e \\ * & 2\widehat{h}_1 \end{pmatrix}.$$

Its determinant has the form $\det(M_2) = 4\widehat{h}_0\widehat{h}_1 - e^2$. According to [32, Lemma III.11.5], the points $P_0$, $P_1$ are linearly independent over $\mathbb{Z}$ (in the strict sense) if and only if the matrix $M_2$ is non-degenerate. We are able to say more by virtue of the following theorem.

**Theorem 1.** *If $\Delta := -D\det(M_2)$ is not a square in $\mathbb{Q}$, then the points $P_0$, $P_1$ are linearly independent over $\mathbb{Z}[\tau]$.*

*Proof.* Introduce yet another (symmetric) Gram matrix

$$M_4 := \begin{pmatrix} \langle P_0, P_0\rangle & \langle P_0, \tau(P_0)\rangle & \langle P_0, P_1\rangle & \langle P_0, \tau(P_1)\rangle \\ * & \langle \tau(P_0), \tau(P_0)\rangle & \langle \tau(P_0), P_1\rangle & \langle \tau(P_0), \tau(P_1)\rangle \\ * & * & \langle P_1, P_1\rangle & \langle P_1, \tau(P_1)\rangle \\ * & * & * & \langle \tau(P_1), \tau(P_1)\rangle \end{pmatrix}.$$

By applying again Lemma 1 and the formula (1), it is easily checked that

$$M_4 = \begin{pmatrix} 2\widehat{h}_0 & t\widehat{h}_0 & e & v_0 \\ * & 2d\widehat{h}_0 & v_1 & de \\ * & * & 2\widehat{h}_1 & t\widehat{h}_1 \\ * & * & * & 2d\widehat{h}_1 \end{pmatrix}.$$

The points $P_0$, $P_1$ are independent over $\mathbb{Z}[\tau]$ if and only if $P_0$, $\tau(P_0)$, $P_1$, $\tau(P_1)$ are independent over $\mathbb{Z}$, that is, the matrix $M_4$ is non-degenerate. To avoid manual computation of its determinant the reader can resort to the code [21] written in Magma. It says that after the substitution $v_0 = te - v_1$ (cf. Lemma 2) we have:

$$\det(M_4) = \rho(v_1)^2, \qquad \text{where} \qquad \rho(v_1) := v_1^2 - tev_1 + (de^2 + D\widehat{h}_0\widehat{h}_1).$$

Further, the discriminant of the quadratic $\mathbb{Q}$-polynomial $\rho$ equals

$$\mathrm{disc}(\rho) = (te)^2 - 4(de^2 + D\widehat{h}_0\widehat{h}_1) = D(e^2 - 4\widehat{h}_0\widehat{h}_1),$$

which is nothing but $\varDelta$ in the statement of the theorem. If $\sqrt{\varDelta} \notin \mathbb{Q}$, then the polynomial $\rho$ does not have roots in the field $\mathbb{Q}$. Since $v_1$ is on the contrary a rational number, $\rho(v_1) \neq 0$ and thus the matrix $M_4$ is non-degenerate. □

**Lemma 3.** *Assume that the initial curve $E$ is not isogenous to a curve of $j$-invariant* 1728 (*i.e., of $D = -4$*). *If in addition $e = 0$ and $\widehat{h}_0 = \widehat{h}_1$, then the premise of Theorem 1 is fulfilled.*

*Proof.* Under the lemma conditions, $\varDelta = -4D\widehat{h}_1^2$. Consequently, $\sqrt{\varDelta} \in \mathbb{Q}$ if and only if $\sqrt{-D} \in \mathbb{Q}$ or, equivalently, $D = -4c^2$ for some $c \in \mathbb{N}$. This amounts to the fact that $E$ is vertically isogenous (necessarily over $\mathbb{F}_q$) to a certain curve of $j = 1728$ as stems, e.g., from [17, Theorems 25.1.2 and 25.4.6]. The lemma is proved. □

## 3 Generation of two "independent" points

First of all, let's specify for self-completeness (in Algorithm 1) the naive try-and-increment method of generating just one point in $E(\mathbb{F}_q)$.

**Algorithm 1:** Naive generation method of one point

```
Data: a seed ∈ {0,1}* and a cryptographic hash function η: {0,1}* → F_q.
Result: a point in E(F_q).
begin
    i := 0;
    x := η(seed||i);
    while (f(x)/q) = −1 do
        i := i + 1;
        x := η(seed||i);
    end
    y := √f(x);
    return (x, y).
end
```

In this section, we will deal solely with points $P_0, P_1 \in \mathcal{E}(F)$ independent over $\mathbb{Z}[\tau]$. In particular, they are of infinite order. It is necessary to give a strict definition of "independent" $\mathbb{F}_q$-points on $E$. It is reasonable to consider for this role the specializations $P_0(u)$, $P_1(u)$ at an element $u \in \mathbb{F}_q$. Of course, the reduction $\mathcal{E}(u)$ is assumed to be $\mathbb{F}_q$-isomorphic to $E$, that is, $\sqrt{g(u)} \in \mathbb{F}_q$. We are obliged to require independence of $P_0$, $P_1$ over $\mathbb{Z}[\tau]$, not only over $\mathbb{Z}$. The fact is that in the cryptographic context the group $E(\mathbb{F}_q)$ is cyclic. Thereby, the endomorphism $\tau$ acts on $E(\mathbb{F}_q)$ as the scalar multiplication $[\lambda]$, where $\lambda$ is one of the two roots of the polynomial $m_\tau$ modulo the group order.

Evidently, there are at most a finite number of elements $u$ at which $P_0$, $P_1$, or $g$ is not correctly defined. Besides, it is worth emphasizing that $\log_{P_0(u)}(P_1(u))$ may be in principle a simple instance of the discrete logarithm problem (DLP) for specific $u$. Sometimes, the equality $P_0(u) = P_1(u)$ even takes place. However, for general $u$, the DLP between $P_0(u)$, $P_1(u)$ seems intractable. Otherwise, it would be surprising and perhaps would affect solving the DLP for two abstract points of $E(\mathbb{F}_q)$.

The new try-and-increment method of generating two "independent" points in $E(\mathbb{F}_q)$ is formalized in Algorithm 2. Up to an $F$-isomorphism of $\mathcal{E}$, it is enough to take $g \in \mathbb{F}_q[u]$. This permits to avoid the inversion operation in $\mathbb{F}_q$ during the evaluation $g(u)$. It is of paramount importance to pick a canonical seed and a cryptographically strong hash function $\eta$. Failing that, a dishonest entity can choose a value $u$ (and then a preimage from $\eta^{-1}(u)$) for which the samples $P_0(u)$, $P_1(u)$ are weak from the viewpoint of the DLP. The same security requirement has to be respected in the case of Algorithm 1 executed twice.

**Algorithm 2:** New generation method of two "independent" points

```
Data: a seed ∈ {0,1}* and a cryptographic hash function η: {0,1}* → F_q,
a polynomial g ∈ F_q[u] and points (x0, y0), (x1, y1) ∈ E^g(F) independent over
Z[τ].
Result: two "independent" points in E(F_q).
begin
    i := 0;
    u := η(seed||i);
    while (g(u)/q) = −1 do
        i := i + 1;
        u := η(seed||i);
    end
    v := √g(u);
    return (x0(u), v y0(u)), (x1(u), v y1(u)).
end
```

It remains to concretize the second data line of Algorithm 2 to bring it to mind. Given two elliptic $\mathbb{F}_q$-curves $E_i$ of $j$-invariant $\neq 0, 1728$ (equivalently, the coefficients $a, b \neq 0$), Mestre [24] and Kuwata–Wang [22] separately found a function $g \in F$ and two non-torsion points $P_i \in E_i^g$. Whenever $E_1$ is a quadratic twist of $E_0$ (unique over $\mathbb{F}_q$), their result gives rise to a natural deterministic map

$\mathbb{F}_q \to (E_0 \times E_1)(\mathbb{F}_q)$ or just $\mathbb{F}_q \to E_0(\mathbb{F}_q)$ if we do not need a point in $E_1(\mathbb{F}_q)$. The latter map is known in the cryptographic literature under the name *simplified SWU map* [8, Section 7], [34, Section 4.1]. By the way, its bottleneck likewise consists in extracting one square root in $\mathbb{F}_q$.

Here, we are on the contrary interested in the tweaked case $E_0 = E_1 =: E$. Roughly speaking, constant-timeness is sacrificed for the second point on the same curve. In this case, Mestre–Kuwata–Wang's formulas have the form

$$x_0 := \frac{b(u^6 - 1)}{au^2(1 - u^4)}, \qquad x_1 := x_0 u^2, \qquad y := u^3.$$

These functions satisfy the equation $f(x_0)y^2 = f(x_1)$. Therefore, we possess the points

$$P_0 := (x_0, 1), \qquad P_1 := (x_1, y) \tag{4}$$

on the twist $\mathcal{E} = E^g$ with respect to

$$g := f(x_0) = \frac{f(x_1)}{y^2} = \frac{num}{den},$$

where

$$num := b\big(b^2u^{12} + 3b^2u^{10} + (a^3 + 6b^2)u^8 + (2a^3 + 7b^2)u^6 + (a^3 + 6b^2)u^4 + 3b^2u^2 + b^2\big),$$
$$den := -\big(au^2(u^2+1)\big)^3.$$

All the above formulas are verified in Magma [21].

Furthermore, this computer algebra system allows to compute the canonical height (pairing). Be careful, Magma's height $\widehat{h}$ is two times larger than in the definition (2), while the pairing $\langle \cdot, \cdot \rangle$ is consistent. It turns out that $\widehat{h}_0 = \widehat{h}_1 = 2$ and $e = 0$ for the current points $P_0$, $P_1$. Consequently, $\det(M_2) = 16 \neq 0$ and so $P_0$, $P_1$ are independent over $\mathbb{Z}$, confirming the fact already established in [24]. Thus, the *Mordell–Weil rank* $r$ of $\mathcal{E}$, i.e., the rank of $\mathcal{E}(F)$ is no less than 2. In accordance with Theorem 1 and Lemma 3, the points $P_0$, $P_1$ are moreover independent over $\mathbb{Z}[\tau]$ (in particular, $r \geqslant 4$) unless $E$ is isogenous to a $j = 1728$ curve denoted by $E_4$. If so, this does not imply that there is a $\mathbb{Z}[\tau]$-dependency between $P_0$, $P_1$. We just do not know an answer concerning this question.

Even if $E \sim E_4$, another pair of short $F$-points on $\mathcal{E}$ probably exists for which the premise of Theorem 1 holds. Nonetheless, there is no essential advantage of such a curve $E$ as compared with $E_4$ itself. First, a (multi-)scalar multiplication on $E$ is slower than on $E_4$, because the latter enjoys the GLV decomposition technique [17, Section 11.3.3] with an order 4 automorphism as $\tau$. And second, the curve $E$ is not much secure than $E_4$, generally speaking. With rare exception, we are able to reduce the DLP from $E$ to $E_4$ by evaluating an $\mathbb{F}_q$-isogeny $E \to E_4$ (of degree $c = \sqrt{-D}/2$) in a polylogarithmic time. These words are justified by the recent breakthrough [27] (cf. [28]) in evaluating isogenies of large prime degrees, not to mention those of smooth degrees (see, e.g., [17, Section 25.6]).

In fact, $\mathbb{F}_q$-curves $E_4$ are not so popular among implementers in contrast to $\mathbb{F}_q$-curves $E_3$ of $j$-invariant 0. This is due to the fact that $\mathrm{Aut}(E_3) \simeq \mathbb{Z}/6$ is greater than $\mathrm{Aut}(E_4) \simeq \mathbb{Z}/4$. As a consequence, curves $E_3$ have more symmetries than ones $E_4$, which impacts on performance of various cryptographic primitives such as pairings [15, Section 3.2.5]. To sum up, Algorithm 2 instantiated by the points (4) is always relevant in real-world cryptography unless $j(E) = 0$. In turn, the previous source [20] treats this special case by providing a generator up to four "independent" $\mathbb{F}_q$-points on $E_3$. If desired, a multi-point generator can be likewise constructed without any problems for curves $E_4$. The author is sure about that, because as well as for $F$-curves of $j = 0$, there are many $F$-curves of $j = 1728$ having moderate Mordell–Weil ranks (see again [24]).

**Acknowledgements.** The author expresses his gratitude to Damien Stehlé and Michael Tsfasman for introducing him deeper to the theory of Euclidean lattices. Also, it is necessary to note the help with Mordell–Weil lattices, provided in different years on `www.mathoverflow.net` by Joseph H. Silverman, Noam D. Elkies, and Will Sawin.

## References

1. Abdalla, M., Barbosa, M.: Perfect forward security of SPAKE2 (2019), `https://eprint.iacr.org/2019/1194`
2. Abdalla, M., Pointcheval, D.: Simple password-based encrypted key exchange protocols. In: Menezes, A. (ed.) Topics in Cryptology – CT-RSA 2005. Lecture Notes in Computer Science, vol. 3376, pp. 191–208. Springer, Berlin, Heidelberg (2005)
3. Barker, E., Kelsey, J.: Recommendation for random number generation using deterministic random bit generators (NIST Special Publication 800-90A Revision 1) (2015), `https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final`
4. Bernstein, D.J., Lange, T.: CM field discriminants (2013), `https://safecurves.cr.yp.to/disc.html`
5. Bernstein, D.J., Lange, T., Niederhagen, R.: Dual EC DRBG (2013), `https://projectbullrun.org/dual-ec`
6. Bernstein, D.J., Lange, T., Niederhagen, R.: Dual EC: A standardized back door. In: Ryan, P.Y.A., Naccache, D., Quisquater, J.J. (eds.) The New Codebreakers. Lecture Notes in Computer Science, vol. 9100, pp. 256–281. Springer, Berlin, Heidelberg (2016)
7. Bottinelli, P.: Breaking Pedersen hashes in practice (2023), `https://research.nccgroup.com/2023/03/22/breaking-pedersen-hashes-in-practice`
8. Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M.: Efficient indifferentiable hashing into ordinary elliptic curves. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 237–254. Springer, Berlin, Heidelberg (2010)
9. Brown, D.R.L., Gjøsteen, K.: A security analysis of the NIST SP 800-90 elliptic curve random number generator. In: Menezes, A. (ed.) Advances in Cryptology – CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 466–481. Springer, Berlin, Heidelberg (2007)
10. Buterin, V.: Open problem: Ideal vector commitment (2020), `https://ethresear.ch/t/open-problem-ideal-vector-commitment/7421`

11. Chaum, D., van Heijst, E., Pfitzmann, B.: Cryptographically strong undeniable signatures, unconditionally secure for the signer. In: Feigenbaum, J. (ed.) Advances in Cryptology – CRYPTO 1991. Lecture Notes in Computer Science, vol. 576, pp. 470–484. Springer, Berlin, Heidelberg (1992)
12. Cox, D.A., with contributions by Lipsett, R.: Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication, AMS Chelsea Publishing, vol. 387. American Mathematical Society, Providence, 3 edn. (2022)
13. Crypto Forum Research Group (CFRG): PAKE selection process (2020), https://github.com/cfrg/pake-selection
14. El Housni, Y., Botrel, G.: EdMSM: Multi-scalar-multiplication for recursive SNARKs and faster Montgomery multiplication (2022), https://eprint.iacr.org/2022/1400
15. El Mrabet, N., Joye, M. (eds.): Guide to pairing-based cryptography. Cryptography and Network Security Series, Chapman and Hall/CRC, New York (2017)
16. Farashahi, R.R., Schoenmakers, B., Sidorenko, A.: Efficient pseudorandom generators based on the DDH assumption. In: Okamoto, T., Wang, X. (eds.) Public Key Cryptography – PKC 2007. Lecture Notes in Computer Science, vol. 4450, pp. 426–441. Springer, Berlin, Heidelberg (2007)
17. Galbraith, S.D.: Mathematics of public key cryptography. Cambridge University Press, New York (2012)
18. Hamburg, M.: Computing the Jacobi symbol using Bernstein–Yang (2021), https://eprint.iacr.org/2021/1271
19. Koshelev, D.: Some remarks on how to hash faster onto elliptic curves (2021), https://eprint.iacr.org/2021/1082
20. Koshelev, D.: Generation of "independent" points on elliptic curves by means of Mordell–Weil lattices (2022), https://eprint.iacr.org/2022/794
21. Koshelev, D.: Magma code (2023), https://github.com/dishport/Generation-of-two-independent-points-on-an-elliptic-curve-of-j-invariant-not-0-1728
22. Kuwata, M., Wang, L.: Topology of rational points on isotrivial elliptic surfaces. International Mathematics Research Notices **1993**(4), 113–123 (1993)
23. Ladd, W., Kaduk, B.: SPAKE2, a PAKE (2022), https://datatracker.ietf.org/doc/draft-irtf-cfrg-spake2
24. Mestre, J.F.: Rang de courbes elliptiques d'invariant donné. Comptes Rendus de l'Académie des Sciences. Série 1, Mathématique **314**(12), 297–319 (1992)
25. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) Advances in Cryptology – CRYPTO 1991. Lecture Notes in Computer Science, vol. 576, pp. 129–140. Springer, Berlin, Heidelberg (1992)
26. Pornin, T.: X25519 implementation for ARM Cortex-M0/M0+ (2020), https://github.com/pornin/x25519-cm0
27. Robert, D.: Evaluating isogenies in polylogarithmic time (2022), https://eprint.iacr.org/2022/1068
28. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. Lecture Notes in Computer Science, vol. 14008, pp. 472–503. Springer, Cham (2023)
29. Sarkar, P.: Computing square roots faster than the Tonelli–Shanks/Bernstein algorithm. Advances in Mathematics of Communications (2022), https://www.aimsciences.org/article/doi/10.3934/amc.2022007
30. Schoenmakers, B., Sidorenko, A.: Cryptanalysis of the dual elliptic curve pseudorandom generator (2006), https://eprint.iacr.org/2006/190

31. Schütt, M., Shioda, T.: Mordell–Weil lattices, A Series of Modern Surveys in Mathematics, vol. 70. Springer, Singapore (2019)
32. Silverman, J.H.: Advanced topics in the arithmetic of elliptic curves, Graduate Texts in Mathematics, vol. 151. Springer, New York (1994)
33. Silverman, J.H.: The arithmetic of elliptic curves, Graduate Texts in Mathematics, vol. 106. Springer, New York, 2 edn. (2009)
34. Wahby, R.S., Boneh, D.: Fast and simple constant-time hashing to the BLS12-381 elliptic curve. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(4), 154–179 (2019)