Batch point compression in the context of advanced pairing-based protocols

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 paper continues previous ones about compression of points on elliptic curves $E_b\colon y^2 = x^3 + b$ (with j-invariant 0) over a finite field $\mathbb{F}_q$ of characteristic $p > 3$. It is shown in detail how any two (resp., three) points from $E_b(\mathbb{F}_q)$ can be quickly compressed to two (resp., three) elements of $\mathbb{F}_q$ (apart from a few auxiliary bits) in such a way that the corresponding decompression stage requires to extract only one cubic (resp., sextic) root in $\mathbb{F}_q$. As a result, for many fields $\mathbb{F}_q$ occurring in practice, the new compression-decompression methods are more efficient than the classical one with the two (resp., three) x or y coordinates of the points, which extracts two (resp., three) roots in $\mathbb{F}_q$. As a by-product, it is also explained how to sample uniformly at random two (resp., three) "independent" $\mathbb{F}_q$-points on E_b essentially at the cost of only one cubic (resp., sextic) root in $\mathbb{F}_q$. Finally, the cases of four and more points from $E_b(\mathbb{F}_q)$ are commented on as well.

Keywords: batch point compression · conic bundles · cubic and sextic roots · elliptic curves of j-invariant 0 · generalized Kummer varieties · generating "independent" points · highly 2-adic fields · rationality problem.

1 Introduction

Nowadays, pairing-based cryptography [12] can be certainly considered as an independent fruitful area of public-key cryptography, which is interesting from both mathematical and practical points of view. There are countless pairing-based protocols, many of which have found applications in the real world. As an example, it is worth noting protocols based on composite-order groups such as Boneh–Goh–Nissim's (BGN) *somewhat homomorphic encryption* [6] or Boneh–Sahai–Waters's *fully collusion resistant traitor tracing* [7]. It is also impossible not to mention *succinct non-interactive zero-knowledge (NIZK) proofs* among which one of the most popular is Groth16 [17]. And their *recursive compositions* are constructed via chains of elliptic curves as first suggested in [3].

Unfortunately, composite-order subgroups of $E_b(\mathbb{F}_q)$ must be tremendous to be protected against sub-exponential factorization algorithms. By virtue of Hasse's inequality (see, e.g., [12, Theorem 2.9]), the order $\#E_b(\mathbb{F}_q) = q + O(\sqrt{q})$,

hence pairing computation on E_b is very expensive as confirmed in [19]. Fortunately, with the help of so-called *Freeman's transformation* [15] (cf. [18, Sections 9-10]), we can almost always rewrite a protocol in the composite-order setting to the prime-order one operating with point vectors from $E_b^n(\mathbb{F}_q)$ for a smaller q and some $n \in \mathbb{N}$. In this case, an instance of the *subgroup decision problem* is a (prime-)order subgroup of $E_b^n(\mathbb{F}_q)$. For the majority of protocols, it is sufficient to take $n = 2$, but there are some protocols (such as Katz–Sahai–Waters's *predicate encryption* [15, Section 7]) needing $n = 3$.

As said, e.g., in [11, Section 2.2], for the sake of efficiency of recursive proofs, one needs to leverage pairing-friendly elliptic curves defined over *highly 2-adic fields* $\mathbb{F}_q$, that is, the number $q - 1$ should be divisible by a non-small power 2^m, where $m \in \mathbb{N}$. This allows to apply the fast Fourier transform (FFT) in order to speed up the polynomial arithmetic over $\mathbb{F}_q$. To be definite, we will suppose that high 2-adicity takes place if $m \geqslant 3$, but in practice usually, $20 < m < 60$. Our choice follows from the fact that for $q \equiv 1 \pmod 8$, it is problematic to express a square root in $\mathbb{F}_q$ via one exponentiation (as is known, e.g., from [12, Section 5.1.7]). Of course, we can always utilize (Cipolla–Lehmer–)Müller's algorithm [34] having the same algebraic complexity $O(\log(q))$ regardless of m. Curiously, this is opposite to other square root algorithms whose complexity depends on m. Nonetheless, Müller's algorithm is still slightly less performant than an exponentiation operation in $\mathbb{F}_q$.

Recall that curves E_b are ordinary (a.k.a. non-supersingular) if and only if the characteristic $p \equiv 1 \pmod 3$ or, equivalently, a primitive cubic root $\omega := \sqrt[3]{1}$ lies in $\mathbb{F}_p$. Since only curves E_b possess an order 6 automorphism (of the form $[-\omega](x, y) := (\omega x, -y)$), according to [12, Section 3.2.5], such pairing-friendly ordinary curves are preferred in pairing-based cryptography. To the author's knowledge, at the moment, the most popular curves are BLS12-381 [39, Section 4.2.1] for a general use and BLS12-377 [11, Table 2] for one layer proof composition, where the numbers after the hyphen equal $\lceil \log_2(q) \rceil$. Moreover, the field $\mathbb{F}_q$ of the latter curve (in contrast to the former one) is highly 2-adic with $m = 46$. Among other things, the pages [21,22] specify 2-cycles of curves of j-invariant 0 (over highly 2-adic fields) among which only one is pairing-friendly.

In compliance with [20, Examples IV.1.3.5-6], elliptic curves are not *rational*, i.e., they are not birationally isomorphic to the affine line $\mathbb{A}^1$. Therefore, from the geometric point of view, the most compact representation of them is on the affine plane $\mathbb{A}^2_{(x,y)}$, for example in the Weierstrass form. Consequently, any point from $E_b^n(\mathbb{F}_q) \subset \mathbb{F}_q^{2n}$ is obviously represented with the help of $2n\lceil \log_2(q) \rceil$ bits. In particular, for $n = 2$ (resp., $n = 3$) and $\log_2(q) \approx 380$, we obtain ≈ 1520 (resp., ≈ 2280) bits, which is quite a lot. For instance, two $\mathbb{F}_q$-points constitute a half of the proof in Groth16 [17, Table 1]. In comparison, with the same 128-bit security level, classical (i.e., non-pairing-friendly) elliptic curves are defined over 256-bit fields $\mathbb{F}_q$. And many widespread cryptosystems on such curves (e.g., ECDH or ECDSA) do not require compressing several points at once, so it is sufficient to manipulate only 512 bits.

At the same time, by virtue of Hasse's inequality, $\mathbb{F}_q$-points on E_b can be compressed to about half with regard to information theory. There is the classical compression-decompression method representing a point as its x (resp., y) coordinate in addition to one (resp., two) bits to uniquely recover the initial y (resp., x) coordinate via extracting in $\mathbb{F}_q$ the square (resp., cubic) root. In comparison with standard arithmetical operations in $\mathbb{F}_q$, the latter one is very costly, because even for $q \not\equiv 1 \pmod 8$ (resp., $q \not\equiv 1 \pmod{27}$), it consists in one exponentiation in $\mathbb{F}_q$ according to Lemma 4 (resp., 3). As a result, after compressing $\mathbb{F}_q$-point vectors of length $n = 2$ (resp., $n = 3$), we obtain ≈ 760 (resp., ≈ 1140) bits at the price of n exponentiations in the decompression stage.

1.1 Brief description of the new compression method

Apart from $\tau_6 := [-\omega]$, there are on E_b the automorphisms

$$\tau_2 := \tau_6^3 \colon (x, y) \mapsto (x, -y), \qquad \tau_3 := \tau_6^4 \colon (x, y) \mapsto (\omega x, y)$$

of orders 2 and 3, respectively. For any $n \in \mathbb{N}$ and $m \in \{2, 3, 6\}$, consider the diagonal subgroup

$$G_{n,m} := \langle (\tau_m, \ldots, \tau_m) \rangle \simeq \mathbb{Z}/m$$

of the automorphism group on E_b^n. Notice that it is Frobenius invariant even if $\omega \notin \mathbb{F}_q$.

Further, introduce the $\mathbb{F}_q$-quotient $GK_{n,m} := E_b^n / G_{n,m}$, which is called *generalized Kummer variety* [40, Section 7], because for $m = 2$, this is a (usual) *Kummer variety* [40, Example 8.1]. Also, we need the notation of the quotient $\mathbb{F}_q$-cover $\varphi_{n,m} : E_b^n \to GK_{n,m}$, which, as usual [20, Theorem I.4.4], gives the function field extension $\mathbb{F}_q(GK_{n,m}) \hookrightarrow \mathbb{F}_q(E_b^n)$. Whenever $m = 2$ or $\omega \in \mathbb{F}_q$, by virtue of Artin's theorem (see, e.g., [33, Theorem VI.1.8]), $\varphi_{n,m}$ is a Galois cover whose Galois group equals $G_{n,m}$. Therefore, $\varphi_{n,m}$ is a *Kummer cover* due to [33, Theorem VI.6.2]. All of the above is illustrated with the famous examples $\varphi_{1,2}(x, y) = x$ and $\varphi_{1,3}(x, y) = y$.

We see that $GK_{1,m}$ are obviously rational curves. More generally, there is the analogous notion of *(geometrically) rational variety* as defined in [20, Example II.8.20.1]. Rationality of the surfaces $GK_{2,3}$, $GK_{2,6}$ is a classical fact. According to [35, Section 2], the threefold $GK_{3,6}$ is also rational and there are [9, Questions 1.3, 1.4] about rationality of $GK_{4,6}$, $GK_{5,6}$. In turn, the varieties $GK_{n,m}$ are never rational for $n \geqslant m$ in accordance with [40, Example 8.10], [35, Remark 2.9]. In fact, we are interested in $\mathbb{F}_q$-rationality of $GK_{n,m}$. In a cryptographic context this concept [38, Definition 6.1] first arose in so-called *torus-based cryptography* for compressing $\mathbb{F}_q$-points of *algebraic tori*. By the way, since pairing values can be interpreted as such points, this compression technique is known to be useful in pairing-based cryptography.

For the Kummer covers $\varphi_{n,m}$, computing an inverse image $\varphi_{n,m}^{-1}(P)$ of a point $P \in \varphi_{n,m}\big(E_b^n(\mathbb{F}_q)\big)$ can be implemented by means of extracting in $\mathbb{F}_q$ some root of degree m. Suppose that $GK_{n,m}$ is an $\mathbb{F}_q$-rational variety and there

are explicit formulas of a birational $\mathbb{F}_q$-isomorphism $\psi_{n,m} : GK_{n,m} \overset{\sim}{\dashrightarrow} \mathbb{A}^n$ and its inverse $\psi_{n,m}^{-1} : \mathbb{A}^n \overset{\sim}{\dashrightarrow} GK_{n,m}$. As is customary in algebraic geometry, the arrow $\dashrightarrow$ (resp., $\overset{\sim}{\dashrightarrow}$) means a (bi)rational map rather than an (iso)morphism, that is, the map may be undefined at some points. Treating them separately, we thus get a new compression-decompression method for all $\mathbb{F}_q$-points on E_b^n. Indeed, the compression (resp., decompression) stage consists in evaluating the map $\chi_{n,m} := \psi_{n,m} \circ \varphi_{n,m}$ at a general point $Q \in E_b^n(\mathbb{F}_q)$ (resp., finding $\chi_{n,m}^{-1}(R)$, where $R := \chi_{n,m}(Q)$).

For the surface $GK_{2,3}$ (resp., $GK_{2,6}$), rationality over $\mathbb{F}_q$ is explicitly established in Section 2 (resp., [26, Sections 2-3]), although these results cannot be considered very important for pure mathematics because of their simplicity. Besides, it turns out that $\mathbb{F}_q$-formulas of $\psi_{3,6}^{\pm 1}$, derived in [35, Section 2] for $b = -1$, are still valid for any $b \in \mathbb{F}_q^*$. Thereby, the threefold $GK_{3,6}$ is always $\mathbb{F}_q$-rational as well. However, if the field $\mathbb{F}_q$ is not highly 2-adic, to compress points from $E_b^2(\mathbb{F}_q)$ (resp., $E_b^3(\mathbb{F}_q)$) we will apply in Section 4 slightly another approach based on $\mathbb{F}_q$-rationality of $GK_{1,3}$ (resp., $GK_{2,3}$). Finally, since the varieties $GK_{n,3}$ are not rational for $n > 2$, we can only hope for breakthroughs concerning $\mathbb{F}_q$-rationality of $GK_{4,6}$, $GK_{5,6}$. The potential compression methods based on these quotient varieties are clearly more efficient, because more $\mathbb{F}_q$-points on E_b (namely 4 or 5) can be compressed at once.

1.2 Relevance of the new compression method

The task of compressing simultaneously several points on an elliptic $\mathbb{F}_q$-curve is not new. One of its solutions already arose in Fan et al.'article [14] for any number $N \in \mathbb{N}$ of points (and not necessarily for j-invariant 0) under the name *multiple point compression* (similarly to *double* and *triple* ones in [25]). The methods of these papers compress to $N + 1$ elements of $\mathbb{F}_q$ (not to mention supplementary bits), i.e., the representation is not optimal. Meanwhile, their decompression stages do not need to find any roots in $\mathbb{F}_q$, but only one inverse element. In comparison with root extraction, inverting in $\mathbb{F}_q$ enjoys Euclidean-type implementations [5,37] costing $O(1)$ field operations, although it is still more expensive than multiplying in $\mathbb{F}_q$.

It is now time to discuss the asymptotic setting as $N \to \infty$. Assume that one of the (optimal) compression methods, based on $GK_{n,m}$ with $n \leqslant 5$, is employed. To compress the majority of points we can apply the method separately $k := N \operatorname{div} n$ times. In turn, the packet of the last $N \bmod n$ points is individually handled. Unfortunately, in the corresponding decompression stage it is inevitable to extract k (or $k+1$ if $n \nmid N$) radicals ($\sqrt{\cdot}$, $\sqrt[3]{\cdot}$, or $\sqrt[6]{\cdot}$), since there is no technique like Montgomery's trick for multiple inversions. Thus, the overall running time amounts to $O(k \log(q))$ or, alternatively, $O(N \log(q))$ operations in $\mathbb{F}_q$ at least when $\mathbb{F}_q$ is not a highly 3-adic field.

Whenever N is large, Fan et al.'s solution becomes an order of magnitude slower than the $GK_{n,m}$-based methods. This tendency can already be seen in the cases $N \in \{4, 5\}$ processed in [14, Section 3]. While [14, Section 4] does

not contain an asymptotic complexity estimate in N of the decompression algorithm, it seemingly needs exponentially many, i.e., $O(2^N)$ operations in $\mathbb{F}_q$. In today's real-world cryptography, N can reach huge values as justified in the below examples. That is why, the compression-decompression algorithm under consideration has never been leveraged in practice as far as the author knows.

Of course, introducing an extra parameter $n \leqslant N$, it is possible to likewise apply Fan et al.'s method $k := N \operatorname{div} n$ times to store/transmit $k(n+1)$ plus $N \bmod n$ (plus 1 if $n \nmid N$) field elements. Consequently, this approach returns k (or $k+1$ if $n \nmid N$) more elements than N, the optimal quantity from the information theory point of view. The extreme case $n = 1$, i.e., $k = N$ means that the points are in their initial uncompressed form. On the other hand, the given block-wise technique permits to diminish the algebraic complexity to $O(2^n k)$. Despite this, it still loses to the technique via the generalized Kummer varieties in terms of the ratio of the compactness and the decompression speed.

It is worth noting recent Botrel–El Housni's work [8] devoted to accelerating multi-scalar multiplication (MSM) with huge numbers (such as 10^8) of basic points $P_i \in E_b(\mathbb{F}_q)$. In addition to them, there are a lot of other points derived from the basic ones in order to speed up MSM. As a result, N is in fact much greater than 10^8. It is said in [8, Section 5.2] that *"However, large MSM instances already use most available memory. For example, when $n = 10^8$ our implementation needs 58GB to store enough BLS12-377 curve points to produce a Groth16 proof."*. By the way, precomputing auxiliary points permitted to Botrel and El Housni to win one of the sections of ZPRIZE competition [1]. The moral of the story is that keeping a tremendous set of points is inevitable if the goal is to achieve record performance.

MSM lies at the heart of a few vector commitment schemes two of which are particularly popular, namely the *Pedersen commitment* [36, Section 3] and the pairing-based *Kate–Zaverucha–Goldberg* (or just *KZG*) one [24]. These schemes have many differences, but for us the most fundamental is the fact that the points P_i in the Pedersen scheme (unlike KZG) are generated without a trusted setup, as they are "independent" of each other. This means that P_i (at least some of them) can simply be regenerated each time to save storage or improve bandwidth. Secure and quick methods of obtaining such points P_i will be briefly discussed in Section 5.

Meanwhile, the KZG scheme has a series of other advantages, due to which it was decided to embed it (rather than the Pedersen scheme) in the new version of the Ethereum protocol. In this connection, a long-term public ceremony [13] was launched once and forever. Its objective is generating (on the curve BLS12-381) up to 2^{15} basic points of the form $P_i = s^i P_0$, where $s \in \mathbb{N}$ is a secret unknown to anyone. Unfortunately, there is no other choice but to (compactly) store or (efficiently) transmit such trusted points P_i.

2 Derivation of formulas

By analogy with [27, Theorem 9], we have

Lemma 1. *There is (up to a birational $\mathbb{F}_q$-isomorphism) the affine model*

$$GK_{2,3} = (y_1^2 - b)t^3 - (y_0^2 - b) \quad \subset \quad \mathbb{A}^3_{(t,y_0,y_1)}$$

for which the corresponding quotient map has the form

$$\varphi_{2,3} \colon E_b^2 \dashrightarrow GK_{2,3} \qquad (x_0, y_0, x_1, y_1) \mapsto \Big(\frac{x_0}{x_1}, y_0, y_1\Big).$$

Proof. Clearly, $\mathbb{F}_q(GK_{2,3}) = \mathbb{F}_q(E_b^2)^{G_{2,3}}$, that is, rational functions on $GK_{2,3}$ are $G_{2,3}$-invariant ones on E_b^2. Also, consider the field

$$F := \mathbb{F}_q(t, y_0, y_1) \subset \mathbb{F}_q(GK_{2,3}), \qquad \text{where} \qquad t := \frac{x_0}{x_1}.$$

Note that $F(x_1) = \mathbb{F}_q(E_b^2)$, because $x_0 = tx_1$. Since $x_1^3 = y_1^2 - b$, the extension degree $[\mathbb{F}_q(E_b^2) : F] \leqslant 3$. At the same time, $[\mathbb{F}_q(E_b^2) : \mathbb{F}_q(GK_{2,3})] = 3$ according to Artin's theorem. Thus, $F = \mathbb{F}_q(GK_{2,3})$. Finally, by looking at the equalities

$$t^3 = \frac{x_0^3}{x_1^3} = \frac{y_0^2 - b}{y_1^2 - b},$$

we obtain the aforementioned equation for $GK_{2,3}$. There are no other dependencies between the coordinates t, y_0, y_1, because $GK_{2,3}$ is a surface in $\mathbb{A}^3$. It remains to apply [20, Corollary I.4.5]. □

Theorem 1. *The generalized Kummer surface $GK_{2,3}$ is $\mathbb{F}_q$-rational.*

Proof. Let's borrow the approach used for proving [27, Theorem 12]. It is based on the theory of *conic bundles* (see, e.g., [27, Section 1.4]), but the reader can verify the formulas below (e.g., via the Magma code [31]) without knowledge of this theory. There is the natural conic bundle structure

$$\pi \colon GK_{2,3} \to \mathbb{A}^1_t \qquad (t, y_0, y_1) \mapsto t.$$

In other words, $GK_{2,3}$ can be seen as an $\mathbb{F}_q(t)$-conic. In a diagonal form,

$$GK_{2,3} = -y_0^2 + t^3 y_1^2 + b(1 - t^3).$$

Therefore, the degenerate (i.e., reducible or, equivalently, singular) fibers of π lie over $t \in \{0, \infty\} \cup \{\omega^i\}_{i=0}^2$, where $\infty := (1 : 0) \in \mathbb{P}^1$. More precisely, $\pi^{-1}(t) = L_t^+ \cup L_t^-$ for these t, where

$$L_0^{\pm} := \begin{cases} t = 0, \\ y_0 = \pm\sqrt{b}, \end{cases} \qquad L_\infty^{\pm} := \begin{cases} t = \infty, \\ y_1 = \pm\sqrt{b}, \end{cases} \qquad L_{\omega^i}^{\pm} := \begin{cases} t = \omega^i, \\ y_1 = \pm y_0. \end{cases}$$

First, after the transformation

$$\tau := \begin{cases} z_0 := y_0, \\ z_1 := ty_1, \end{cases} \qquad \tau^{-1} = \begin{cases} y_0 := z_0, \\ y_1 := z_1/t \end{cases}$$

we obtain the cubic surface

$$GK'_{2,3} := \tau(GK_{2,3}) \;=\; -z_0^2 + tz_1^2 + b(1-t^3) \quad \subset \quad \mathbb{A}^3_{(t,z_0,z_1)}.$$

We then *blow down* [20, Section V.3] one of the components $\tau(L_1^{\pm})$ by means of the transformation

$$\theta := \begin{cases} y_0 := \dfrac{z_0 - z_1}{1-t}, \\ y_1 := \dfrac{z_0 - tz_1}{1-t}, \end{cases} \qquad \theta^{-1} = \begin{cases} z_0 := -ty_0 + y_1, \\ z_1 := -y_0 + y_1, \end{cases}$$

coming to

$$S := \theta(GK'_{2,3}) \;=\; ty_0^2 - y_1^2 + b(t^2+t+1) \quad \subset \quad \mathbb{A}^3_{(t,y_0,y_1)}.$$

Further, blowing down simultaneously some pair of components over $t \in \{\omega, \omega^2\}$ has the form

$$\eta := \begin{cases} z_0 := \dfrac{(t+1)y_0 + y_1}{t^2+t+1}, \\ z_1 := \dfrac{ty_0 + (t+1)y_1}{t^2+t+1}, \end{cases} \qquad \eta^{-1} = \begin{cases} y_0 := (t+1)z_0 - z_1, \\ y_1 := -tz_0 + (t+1)z_1, \end{cases}$$

which gives the simpler surface

$$T := \eta(S) \;=\; tz_0^2 - z_1^2 + b \quad \subset \quad \mathbb{A}^3_{(t,z_0,z_1)}.$$

Note that the maps τ, θ, η respect the conic bundle π, that is, they can be seen as $\mathbb{F}_q(t)$-isomorphisms of conics. That is why, we avoid the tautology $t := t$ in their description. Finally, the projection $pr : T \xrightarrow{\sim} \mathbb{A}^2_{(z_0,z_1)}$ is a desired map, because $t = (z_1^2 - b)/z_0^2$. $\square$

For the compositions

$$\psi_{2,3} := pr \circ \eta \circ \theta \circ \tau, \qquad \chi_{2,3} := \psi_{2,3} \circ \varphi_{2,3},$$

the Magma code [31] results in the formulas

$$\chi_{2,3} \colon E_b^2 \dashrightarrow \mathbb{A}^2_{(z_0,z_1)} \qquad \chi_{2,3} = \begin{cases} z_0 := \dfrac{x_1(2x_0^2y_1 - x_0x_1(y_0 - y_1) - 2y_0x_1^2)}{y_0^2 - y_1^2}, \\ z_1 := \dfrac{x_0^3y_1 + 2x_0x_1(x_0y_1 - y_0x_1) - y_0x_1^3}{y_0^2 - y_1^2}, \end{cases}$$

$$\psi_{2,3}^{-1} \colon \mathbb{A}^2_{(z_0,z_1)} \xrightarrow{\sim} GK_{2,3} \qquad \psi_{2,3}^{-1} = \begin{cases} t := \dfrac{z_1^2 - b}{z_0^2}, \\ y_0 := \dfrac{z_0^3z_1 - 2z_0(z_0 - z_1)(z_1^2 - b) - (z_1^2 - b)^2}{z_0^3}, \\ y_1 := -\dfrac{z_0^2(z_0 - 2z_1) + (2z_0 - z_1)(z_1^2 - b)}{z_1^2 - b}. \end{cases}$$

Let's consider the cases when the denominators equal zero. Obviously, $t \in \{0, \infty\} \Rightarrow x_0 x_1 = 0$, and

$$y_0^2 - y_1^2 = 0 \quad \Leftrightarrow \quad \exists k \in \mathbb{Z}/6 \colon (x_1, y_1) = [-\omega]^k (x_0, y_0). \tag{1}$$

In turn, the next lemma is checked in Magma.

Lemma 2. *The variable $z_0 = 0$ (i.e., $z_1 = \pm\sqrt{b}$ under the condition $t \neq 0$) if and only if $(t, y_0, y_1) \in \mathrm{Im}(\varrho_\pm)$ for the sections of π given by*

$$\varrho_\pm \colon \mathbb{A}^1_t \dashrightarrow GK_{2,3} \qquad \varrho_\pm := \begin{cases} y_0 := \pm\sqrt{b}(2t+1), \\ y_1 := \dfrac{\pm\sqrt{b}(t+2)}{t}. \end{cases}$$

Furthermore, it is readily seen that

$$t = \frac{y_0 \mp \sqrt{b}}{\pm 2\sqrt{b}} = \frac{\pm 2\sqrt{b}}{y_1 \mp \sqrt{b}}$$

and we eventually get the conics

$$C_{\pm 1} := \mathrm{Im}(\varrho'_{\pm 1}) \;=\; (y_0 \mp \sqrt{b})(y_1 \mp \sqrt{b}) - 4b \quad \subset \quad \mathbb{A}^2_{(y_0, y_1)}, \tag{2}$$

where $\varrho'_{\pm 1} := pr \circ \varrho_\pm$ and $pr : GK_{2,3} \to \mathbb{A}^2_{(y_0, y_1)}$ by abuse of notation. The objects $C_{\pm 1}$, $\varrho'_{\pm 1}$ will be needed below to process one of the degenerate cases during (de)compression.

3 New compression method for two points

We need the auxiliary sets

$$V' := \big\{(x, y) \in E_b \mid xy = 0\big\} \cup \big\{\mathcal{O}\big\} \quad \subset \quad E_b[2] \cup E_b[3],$$

$$V \;:=\; E_b \times V' \;\cup\; V' \times E_b,$$

where $\mathcal{O} := (0 : 1 : 0)$. Formally, for two points $P_i = (x_i, y_i)$ from $E_b(\mathbb{F}_q) \setminus V'$, the new compression map has the form

$$\mathrm{com}_{2,3} \colon E_b^2(\mathbb{F}_q) \setminus V \;\hookrightarrow\; \mathbb{F}_q^2 \times [0, 5] \times [0, 2]$$

$$\mathrm{com}_{2,3}(P_0, P_1) := \begin{cases} (x_0, y_0, k, 0) & \text{if} \quad \exists k \in \mathbb{Z}/6 \colon P_1 = [-\omega]^k (P_0), \\ (t, x_1, k, 1) & \text{if} \quad \exists k \in \mathbb{Z}/2 \colon (y_0, y_1) \in C_{(-1)^k}, \\ (z_0, z_1, n, 2) & \text{otherwise.} \end{cases}$$

Here, $(z_0, z_1) = \chi_{2,3}(P_0, P_1)$ and $n \in [0, 2]$ is the position number of the element $x_1 \in \mathbb{F}_q^*$ in the set $\big\{\omega^i x_1\big\}_{i=0}^2 \cap \mathbb{F}_q^*$ with respect to some order in $\mathbb{F}_q^*$. For example, in the case of a prime q, this can be the usual numerical one. It

is worth noting that in the definition of $\mathrm{com}_{2,3}$ the condition (1) is successively checked by iterating over elements of $\mathbb{Z}/6$. The same strategy is applied for the condition of belonging to the conics $C_{(-1)^k}$ having the equation (2). Further, the set $[0,5]\times[0,2]$ clearly requires 5 bits for representing its elements. Finally, since in discrete logarithm cryptography points of small orders do not occur, we omit the definition of the compression map on $V(\mathbb{F}_q)$ for the sake of simplicity, although it can be easily defined if desired.

The corresponding decompression map is given as follows:

$$\mathrm{com}_{2,3}^{-1}\colon \mathrm{Im}(\mathrm{com}_{2,3}) \;\rightsquigarrow\; E_b^2(\mathbb{F}_q)\setminus V$$

$$\mathrm{com}_{2,3}^{-1}(z_0,z_1,m,\ell) = \begin{cases} (z_0,z_1,x_1,y_1) & \text{if} \quad \ell=0 \text{ and } (x_1,y_1)=[-\omega]^m(z_0,z_1),\\ (z_0z_1,y_0,z_1,y_1) & \text{if} \quad \ell=1 \text{ and } (y_0,y_1)=\varrho'_{(-1)^m}(z_0),\\ (tx_1,y_0,x_1,y_1) & \text{if} \quad \ell=2 \text{ and } (t,y_0,y_1)=\psi_{2,3}^{-1}(z_0,z_1), \end{cases}$$

where for $\ell=2$, the initial $x_1=\sqrt[3]{g_1}$ (for $g_1:=y_1^2-b$) can be determined with the help of $m=n$.

According to the next lemma, for $q\not\equiv 1 \pmod{27}$, the cubic root $\sqrt[3]{g_1}$ can be extracted at the cost of one exponentiation in $\mathbb{F}_q$ (in particular, without inverting the denominator of g_1).

Lemma 3. *Given a fraction $g=u/v\in(\mathbb{F}_q^*)^3$ such that $u,\ v\in\mathbb{F}_q^*$, we obtain:*

$$\sqrt[3]{g} = \begin{cases} g^{(2q-1)/3} \;=\; u\cdot(u^2v)^{(q-2)/3} & \text{if} \quad q\equiv 2 \pmod 3,\\ g^{(8q-5)/9} \;=\; u^3\cdot(u^8v)^{(q-4)/9} & \text{if} \quad q\equiv 4 \pmod 9,\\ g^{(q+2)/9} \;=\; uv^5\cdot(uv^8)^{(q-7)/9} & \text{if} \quad q\equiv 7 \pmod 9,\\ \zeta\cdot g^{(2q+7)/27} \;=\; \zeta uv^8\cdot(u^2v^{25})^{(q-10)/27} & \text{if} \quad q\equiv 10 \pmod{27},\\ \zeta\cdot g^{(q+8)/27} \;=\; \zeta uv^{17}\cdot(uv^{26})^{(q-19)/27} & \text{if} \quad q\equiv 19 \pmod{27} \end{cases}$$

for some $\zeta\in(\mathbb{F}_q^*)^{(q-1)/9}$.

Proof. Consider, e.g., the case $q\equiv 7 \pmod 9$, which is relevant for the curve BLS12-377. For $e:=(q+2)/9\in\mathbb{N}$, we have:

$$g^e = u^e\cdot v^{q-1-e} = u^e\cdot v^{(8q-11)/9} = uv^5\cdot(uv^8)^{(q-7)/9},$$
$$(g^e)^3 = g^{(q+2)/3} = g^{(q-1)/3}\cdot g = g.$$

The cases $q\equiv 4 \pmod 9$ and $q\equiv 10 \pmod{27}$ are similarly processed in [30, Equalities (2), (3)]. The remaining cases $q\equiv 2 \pmod 3$ and $q\equiv 19 \pmod{27}$ are left to the reader. □

To complete the picture, let's include the analogous result from [4, Section 5], [41, Section 4.2] for a square root.

Lemma 4. *Given a fraction $f = u/v \in (\mathbb{F}_q^*)^2$ such that u, $v \in \mathbb{F}_q^*$, we obtain:*

$$\sqrt{f} = \begin{cases} f^{(q+1)/4} \;=\; uv(uv^3)^{(q-3)/4} & \text{if} \quad q \equiv 3 \pmod 4, \\ \zeta \cdot f^{(q+3)/8} \;=\; \zeta uv^3 \cdot (uv^7)^{(q-5)/8} & \text{if} \quad q \equiv 5 \pmod 8 \end{cases}$$

for some $\zeta \in (\mathbb{F}_q^*)^{(q-1)/4}$.

Since the projective or *Jacobian coordinates* [12, Sections 2.3.2 and 10.7.9] are preferred in practice, the decompression stage does not require finding inverse elements at all. By definition, in these coordinates the curve E_b possesses the equations

$$\overline{E_b}\colon Y^2Z = X^3 + bZ^3, \qquad \overline{E_b}\colon Y^2 = X^3 + bZ^6,$$

respectively. And there are the birational isomorphisms

$$\sigma\colon \overline{E_b} \dashrightarrow E_b \qquad (X : Y : Z) \mapsto \left(\frac{X}{Z}, \frac{Y}{Z}\right), \qquad (X : Y : Z) \mapsto \left(\frac{X}{Z^2}, \frac{Y}{Z^3}\right),$$

respectively. By the way, in both cases,

$$\sigma^{-1}\colon E_b \dashrightarrow \overline{E_b} \qquad (x, y) \mapsto (x : y : 1).$$

If the compression stage starts from the projective or Jacobian coordinates, then even in the classical method it is necessary to compute one inverse in $\mathbb{F}_q$. Indeed, given two points $(X_i : Y_i : Z_i) \in \overline{E_b}(\mathbb{F}_q)$ with $Z_i \neq 0$, one needs the value $v := (Z_0Z_1)^{-1}$ in order to get $Z_0^{-1} = vZ_1$ and $Z_1^{-1} = vZ_0$. This famous trick, originally attributed to Montgomery, is clearly generalized to any number N of inversions. And in accordance with [16, Exercise 2.5.5], apart from one inversion, $3(N-1)$ field multiplications are sufficient for Montgomery's trick.

In the compression stage of the new method, instead of the two inversions v, $(y_0^2 - y_1^2)^{-1}$, only one is also enough, because

$$\chi_{2,3} \circ \sigma^{\times 2} = \left(\frac{\mathrm{num}_0}{\mathrm{den}}, \frac{\mathrm{num}_1}{\mathrm{den}}\right)\colon \quad \overline{E_b}^2 \dashrightarrow \mathbb{A}^2_{(z_0,z_1)}$$

for some polynomials num_i, $\mathrm{den} \in \mathbb{F}_q[X_i, Y_i, Z_i]_{i=0}^1$ trivially obtained from the formulas of $\chi_{2,3}$. To determine the position number n one needs to know Z_1^{-1}, hence we should in fact invert $Z_1 \cdot \mathrm{den}$. It is worth emphasizing that all of the above is equally valid for the degenerate cases $\ell \in \{0, 1\}$.

To sum up, a reference Magma implementation of $\mathrm{com}_{2,3}^{\pm 1}$ is represented in [31] for BLS12-377 in projective coordinates. It takes into account all the mentioned optimization tricks. The program code can be readily modified to treat other elliptic curves of j-invariant 0 over non-highly 3-adic fields $\mathbb{F}_q$.

4 Folklore compression method for two points and its variation for three ones

First, we put $f_i := x_i^3 + b$ and $g_i := y_i^2 - b$. Since the numbers 2, 3 are relatively prime, the roots $y_0 = \sqrt{f_0}$ and $x_1 = \sqrt[3]{g_1}$ can be extracted simultaneously, that

is, at the cost of a sixth root in $\mathbb{F}_q$. Indeed, it is sufficient to compute $\alpha := \sqrt[6]{h} = \sqrt{f_0}\sqrt[3]{g_1}$ for $h := f_0^3 g_1^2$, because $\sqrt[3]{g_1} = f_0 g_1/\alpha^2$ and $\sqrt{f_0} = \alpha/\sqrt[3]{g_1}$. Moreover, by analogy with $\sqrt{\cdot}$, $\sqrt[3]{\cdot}$ (see Lemmas 3, 4), the value α can be expressed via one exponentiation in $\mathbb{F}_q$ whenever $q \not\equiv 1 \pmod 8$, $q \not\equiv 1 \pmod{27}$. Note that

$$\begin{aligned} q \equiv 3 \pmod 4,\ q \equiv 2 \pmod 3 \quad &\Leftrightarrow \quad q \equiv 11 \pmod{12}, \\ q \equiv 3 \pmod 4,\ q \equiv 4 \pmod 9 \quad &\Leftrightarrow \quad q \equiv 31 \pmod{36}, \\ q \equiv 3 \pmod 4,\ q \equiv 7 \pmod 9 \quad &\Leftrightarrow \quad q \equiv 7 \pmod{36}. \end{aligned}$$

Lemma 5. *Given a fraction $h = u/v \in (\mathbb{F}_q^*)^6$ such that $u,\ v \in \mathbb{F}_q^*$, we obtain:*

$$\sqrt[6]{h} = \begin{cases} h^{(q+1)/12} \ = \ uv^9 \cdot (uv^{11})^{(q-11)/12} & \text{if} \quad q \equiv 11 \pmod{12}, \\ h^{(q+5)/36} \ = \ uv^{29} \cdot (uv^{35})^{(q-31)/36} & \text{if} \quad q \equiv 31 \pmod{36}, \\ h^{(5q+1)/36} \ = \ uv^5 \cdot (u^5 v^{31})^{(q-7)/36} & \text{if} \quad q \equiv 7 \pmod{36}. \end{cases}$$

The case $q \equiv 11 \pmod{12}$ is discussed in [28, Section 2.1]. The remaining cases are likewise verified. To save the space we skip other $q \not\equiv 1 (\mathrm{mod}\ 8)$, $q \not\equiv 1 (\mathrm{mod}\ 27)$ in the lemma, because they bring nothing new and they are not vitally important in the current article.

Thus, there is the compression map

$$E_b^2(\mathbb{F}_q) \setminus V \ \hookrightarrow \ \mathbb{F}_q^2 \times [0,5] \qquad (P_0, P_1) \mapsto (x_0, y_1, n),$$

where $n \in [0,5]$ is the position number of the element $y_0 x_1 \in \mathbb{F}_q^*$ in the set $\left\{(-1)^i \omega^j \cdot y_0 x_1\right\}_{i=0,j=0}^{1,2} \cap \mathbb{F}_q^*$ with respect to some order in $\mathbb{F}_q^*$. As above, n is used in the decompression stage for recovering the original y_0, x_1. Notice that at the heart of this method is $\mathbb{F}_q$-rationality of

$$E_b^2/G = GK_{1,2} \times GK_{1,3} \simeq \mathbb{A}^2_{(x_0,y_1)}, \qquad \text{where} \qquad G := G_{1,2} \times G_{1,3} \simeq \mathbb{Z}/6.$$

We will call the method *folklore*, because it does not require an algebraic geometry technique, so someone perhaps already knows it. The significant drawback of the folklore method consists in the fact that (in contrast to $\mathrm{com}_{2,3}$) it does not work efficiently over highly 2-adic fields $\mathbb{F}_q$, that is, Lemma 4 cannot be leveraged. The same drawback exists for the other method [26, Sections 2-3] based on $\mathbb{F}_q$-rationality of $GK_{2,6}$. Nevertheless, since the folklore one has a slightly simpler definition, we conclude that it is more preferred for use when possible.

Similarly, one can apply the folklore methodology to the new method with z_0, z_1 in order to compress three points $P_i = (x_i, y_i)$ from $E_b(\mathbb{F}_q) \setminus V'$. As earlier, consider the set

$$V \ := \ E_b^2 \times V' \ \cup \ E_b \times V' \times E_b \ \cup \ V' \times E_b^2.$$

It is about the compression map

$$E_b^3(\mathbb{F}_q) \setminus V \ \hookrightarrow \ \mathbb{F}_q^3 \times [0,5] \times [0,2] \times [0,1] \qquad (P_0, P_1, P_2) \mapsto (z_0, z_1, x_2, n, \ell, s),$$

where $(z_0, z_1, m, \ell) = \mathrm{com}_{2,3}(P_0, P_1)$ and, in the non-degenerate case $\ell = 2$, the number $n \in [0, 5]$ is the position of the element $x_1 y_2 \in \mathbb{F}_q^*$. In turn, for $\ell \in \{0, 1\}$, we put $n := m$ and the additional sign bit s is utilized to recover y_2 (regardless of P_0, P_1). Since for these ℓ the latter points are obtained without root computations, the overall complexity does not go beyond one exponentiation in $\mathbb{F}_q$.

Thus, we completely justified Tables 1, 2, which contain a complexity comparison (all the operations are carried out in $\mathbb{F}_q$) of the compression-decompression methods for two and three points, respectively. As is customary, the addition, subtraction, and multiplication operations in $\mathbb{F}_q$ are omitted, because they are much cheaper.

Taking this opportunity, the author emphasizes that arguments of the given paper, related to avoiding the inversion operation, are equally valid for the previous compression-decompression methods. In other words, the number of inversions in [27, Theorem 13], [26, Tables 1, 2] can be reduced to only one in the compression stage. By virtue of [16, Exercise 2.5.5] and Lemmas 3, 4, 5, this reduction requires no more than several tens of auxiliary multiplications. Their exact number heavily depends on q and a concrete implementation. So, providing the given number is outside the scope of the current article.

method	Galois group	compression	decompression
classical with x_0, x_1	$G_{1,2}^2$	one inversion	two $\sqrt{\cdot}$
classical with y_0, y_1	$G_{1,3}^2$		two $\sqrt[3]{\cdot}$
folklore with x_0, y_1	$G_{1,2} \times G_{1,3}$		one $\sqrt[6]{\cdot}$
new with z_0, z_1	$G_{2,3}$		one $\sqrt[3]{\cdot}$
from [14, Sections 2.1 and 2.2]	not applicable		for free

Table 1. Worst-case complexity for compressing $\overline{E_b}^2(\mathbb{F}_q)$ (with respect to the projective or Jacobian coordinates). The last method compresses to $\approx 3\lceil \log_2(q) \rceil$ bits and the other ones compress to $\approx 2\lceil \log_2(q) \rceil$ bits.

5 Generating uniformly at random "independent" points

From the introduction we know that the Pedersen commitment (and some other schemes) lacks a set of basic "independent" points $P_i \in E_b(\mathbb{F}_q)$. "Independence" means that a non-trivial linear relation between them is unknown to anyone. As explained below, the decompression methods of the previous sections are naturally interpreted as methods of generating the desired points P_i.

First of all, suppose that $q \equiv 2 \pmod 3$. Under this condition, curves E_b are supersingular and every element of $\mathbb{F}_q$ has the unique cubic root in $\mathbb{F}_q$. Although

method	Galois group	compression	decompression
classical with x_0, x_1, x_2	$G_{1,2}^3$	one inversion	three $\sqrt{\cdot}$
classical with y_0, y_1, y_2	$G_{1,3}^3$		three $\sqrt[3]{\cdot}$
folklore-classical with x_0, x_1, y_2	$G_{1,2}^2 \times G_{1,3}$		one $\sqrt[6]{\cdot}$ and one $\sqrt{\cdot}$
folklore-classical with x_0, y_1, y_2	$G_{1,2} \times G_{1,3}^2$		one $\sqrt[6]{\cdot}$ and one $\sqrt[3]{\cdot}$
new with z_0, z_1, z_2	$G_{2,3} \times G_{1,2}$		one $\sqrt[6]{\cdot}$
from [25, Section 3.2]	not applicable		for free

Table 2. Worst-case complexity for compressing $\overline{E_b}^3(\mathbb{F}_q)$ (with respect to the projective or Jacobian coordinates). The last method compresses to $\approx 4\lceil \log_2(q) \rceil$ bits and the other ones compress to $\approx 3\lceil \log_2(q) \rceil$ bits.

$\varphi_{n,3}$ are not Galois covers anymore, we still can find the inverse image under $\varphi_{n,3}$ via extracting a cubic root in $\mathbb{F}_q$. In particular,

$$\varphi_{1,3}^{-1} \colon \mathbb{F}_q \to E_b(\mathbb{F}_q), \qquad \varphi_{2,3}^{-1} \colon GK_{2,3}(\mathbb{F}_q) \to E_b^2(\mathbb{F}_q)$$

are true maps, that is, they are correctly defined for each input argument. The former is widely known as *Boneh–Franklin's encoding* [12, Section 8.3.2]. The latter gives rise to the new encoding $\chi_{2,3}^{-1} \colon \mathbb{F}_q^2 \to E_b^2(\mathbb{F}_q)$, because points at which $\psi_{2,3}^{-1}$ is not defined, as usual, can be independently processed. Thus, $\chi_{2,3}^{-1}$ allows to generate two "independent" $\mathbb{F}_q$-points on E_b twice as efficiently as $\varphi_{1,3}^{-1}$ applied two times.

At the moment, supersingular curves are not preferable in discrete logarithm cryptography because of their small embedding degrees ($\leqslant 3$ in the characteristic $p > 3$ according to [12, Section 4.3]). Thereby, we are obliged to deal with $q \equiv 1 \pmod 3$. In this case, $\varphi_{1,3}^{-1}$, $\varphi_{2,3}^{-1}$ are no longer well-defined maps along with the arbitrary $\varphi_{n,m}^{-1}$, where $n \leqslant 5$ and $m \in \{2, 3, 6\}$ as before. Nonetheless, this does not prevent from constructing the corresponding generation methods. As an example, the conventional x-coordinate method is based on $\varphi_{1,2}^{-1}$. Its exact description can be easily found in the literature (see, e.g., [32, Algorithm 1]). It is necessary to iterate vectors $v \in \mathbb{F}_q^n$ until $\chi_{n,m}^{-1}(v)$ lies in $E_b^n(\mathbb{F}_q)$. As during the decompression, (maximum three) supplementary bits can serve for choosing a concrete preimage from $\chi_{n,m}^{-1}(v)$. Moreover, the resulting point vectors are distributed uniformly at random in $E_b^n(\mathbb{F}_q)$ (up to a negligible error) if so are v and the bits. This is an immediate corollary from birationality of the map $\psi_{n,m}^{-1}$.

In the language of function fields, $\mathbb{F}_q(E_b^n) \simeq \mathbb{F}_q(\mathbb{A}^n)(\sqrt[m]{\mathfrak{f}})$ for a certain polynomial $\mathfrak{f} \in \mathbb{F}_q[\mathbb{A}^n]$. In this notation, $v \in \mathbb{F}_q^n$ should be sampled while $\left(\frac{\mathfrak{f}(v)}{q}\right)_m \neq 1$, where $\left(\frac{\cdot}{q}\right)_m$ is the m-th power residue symbol in $\mathbb{F}_q$. By definition, this symbol is the result of the exponentiation to $(q-1)/m \in \mathbb{N}$, but for our m, it enjoys Euclidean-type implementations [23] working in time $O(1)$. Meanwhile, m samples are enough on average to meet an m-th residue of the form $\mathfrak{f}(v)$.

This intuitive statement fits into [29, Lemma 1] for $n = 1$. It is not difficult to extend that lemma to the other values n by exploiting the fact that the number of $\mathbb{F}_q$-points on the hypersurface $u^m = \mathfrak{f}(v)$ in $\mathbb{A}^{n+1}$ is approximately equal to $\#E_b^n(\mathbb{F}_q) = q^n + O(q^{n-1/2})$. Therefore, computing one radical $\sqrt[m]{\cdot} \in \mathbb{F}_q$ is the unique bottleneck of the generation method under consideration.

The paper [29] develops another approach of generating the n points P_i through the *Mordell–Weil lattices* MW_m of the *elliptic surfaces* $y^2 = x^3 + t^m + c$, where $c \in \mathbb{F}_q^*$. The given approach has its pros and cons. On the one hand, it is generalized (as shown in [32]) to some extent to elliptic $\mathbb{F}_q$-curves of other j-invariants. Besides, the MW_m-based methods can in theory return (essentially at the price of $\sqrt[6]{\cdot}$) up to $34 \gg 5$ "independent" $\mathbb{F}_q$-points on E_b as stressed in [29, Section 3]. On the other hand, their resulting distributions on $E_b^n(\mathbb{F}_q)$ are far from uniform, because the input argument is taken from $\mathbb{F}_q$ instead of $\mathbb{F}_q^n$.

In conclusion, Table 3 summarizes the two represented generation methods (cf. [29, Table 2]). By analogy with Lemmas 3, 4, and 5, it is often possible to batch $\sqrt[4]{\cdot}$, $\sqrt[5]{\cdot}$ with inverting in $\mathbb{F}_q$, which is not superfluous in the MW_4, MW_5-based methods, respectively. As always, the projective or Jacobian coordinates should be used if we want to completely avoid the inverse operation. The x (resp., y) coordinate method through $GK_{1,2}$ (resp., $GK_{1,3}$) is omitted in the table for compactness, because it is equivalent to (resp., worse than) the method through MW_2 (resp., $GK_{2,3}$). Finally, the MW_3-based one is also useless again because of the first row of the table.

method based on	source	n	m	is uniform?
$GK_{2,3}$	this	2	3	yes
$GK_{1,2} \times GK_{2,3}$		3	6	
$GK_{5,6}$	theoretically	5		
MW_m	[29]	$m-1$	$\leqslant 5$	only if $m = 2$
MW_{360}	theoretically	34	6	no

Table 3. Methods of generating n "independent" $\mathbb{F}_q$-points on ordinary elliptic $\mathbb{F}_q$-curves $\overline{E_b}$. It is assumed everywhere that $m \mid q-1$. The average execution time for all of them amounts to $m\big(\frac{\cdot}{q}\big)_m + \sqrt[m]{\cdot}$.

References

1. ZPRIZE competition (2022), `https://www.zprize.io`
2. Aranha, D.F., Pagnin, E., Rodríguez-Henríquez, F.: LOVE a pairing. In: Longa, P., Ràfols, C. (eds.) Progress in Cryptology – LATINCRYPT 2021. Lecture Notes in Computer Science, vol. 12912, pp. 320–340. Springer, Cham (2021)

3. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Scalable zero knowledge via cycles of elliptic curves. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology – CRYPTO 2014. Lecture Notes in Computer Science, vol. 8617, pp. 276–294. Springer, Berlin, Heidelberg (2014)
4. Bernstein, D.J., Duif, N., Lange, T., Schwabe, P., Yang, B.Y.: High-speed high-security signatures. Journal of Cryptographic Engineering **2**(2), 77–89 (2012)
5. Bernstein, D.J., Yang, B.Y.: Fast constant-time gcd computation and modular inversion. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(3), 340–398 (2019)
6. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) Theory of Cryptography. TCC 2005. Lecture Notes in Computer Science, vol. 3378, pp. 325–341. Springer, Berlin, Heidelberg (2005)
7. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 573–592. Springer, Berlin, Heidelberg (2006)
8. Botrel, G., El Housni, Y.: Faster Montgomery multiplication and multi-scalar-multiplication for SNARKs. IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) **2023**(3), 504–521 (2023)
9. Catanese, F., Oguiso, K., Verra, A.: On the unirationality of higher dimensional Ueno-type manifolds. Revue Roumaine de Mathématiques Pures et Appliquées **60**(3), 337–353 (2015)
10. Chatterjee, S., Hankerson, D., Menezes, A.: On the efficiency and security of pairing-based protocols in the type 1 and type 4 settings. In: Hasan, M.A., Helleseth, T. (eds.) Arithmetic of Finite Fields. WAIFI 2010. Lecture Notes in Computer Science, vol. 6087, pp. 114–134. Springer, Berlin, Heidelberg (2010)
11. El Housni, Y., Guillevic, A.: Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition. In: Krenn, S., Shulman, H., Vaudenay, S. (eds.) Cryptology and Network Security. CANS 2020. Lecture Notes in Computer Science, vol. 12579, pp. 259–279. Springer, Cham (2020)
12. El Mrabet, N., Joye, M. (eds.): Guide to pairing-based cryptography. Cryptography and Network Security Series, Chapman and Hall/CRC, New York (2017)
13. Ethereum Foundation: ethereum/kzg-ceremony (2022), `https://github.com/ethereum/kzg-ceremony`
14. Fan, X., Otemissov, A., Sica, F., Sidorenko, A.: Multiple point compression on elliptic curves. Designs, Codes and Cryptography **83**(3), 565–588 (2017)
15. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 44–61. Springer, Berlin, Heidelberg (2010)
16. Galbraith, S.D.: Mathematics of public key cryptography. Cambridge University Press, New York (2012)
17. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology – EUROCRYPT 2016. Lecture Notes in Computer Science, vol. 9665, pp. 305–326. Springer, Berlin, Heidelberg (2016)
18. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) Advances in Cryptology – EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer, Berlin, Heidelberg (2008)
19. Guillevic, A.: Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R.

(eds.) Applied Cryptography and Network Security. ACNS 2013. Lecture Notes in Computer Science, vol. 7954, pp. 357–372. Springer, Berlin, Heidelberg (2013)
20. Hartshorne, R.: Algebraic geometry, Graduate Texts in Mathematics, vol. 52. Springer, New York, 8 edn. (1997)
21. Hopwood, D.: The pasta curves for Halo 2 and beyond (2020), https://electriccoin.co/blog/the-pasta-curves-for-halo-2-and-beyond
22. Hopwood, D.: Pluto/Eris supporting evidence (2021), https://github.com/daira/pluto-eris
23. Joye, M., Lapiha, O., Nguyen, K., Naccache, D.: The eleventh power residue symbol. Journal of Mathematical Cryptology **15**(1), 111–122 (2021)
24. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) Advances in Cryptology – ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 177–194. Springer, Berlin, Heidelberg (2010)
25. Khabbazian, M., Gulliver, T.A., Bhargava, V.K.: Double point compression with applications to speeding up random point multiplication. IEEE Transactions on Computers **56**(3), 305–313 (2007)
26. Koshelev, D.: Faster point compression for elliptic curves of j-invariant 0. Mathematical Aspects of Cryptography **12**(4), 115–123 (2021)
27. Koshelev, D.: New point compression method for elliptic $\mathbb{F}_{q^2}$-curves of j-invariant 0. Finite Fields and Their Applications **69**, Article 101774 (2021)
28. Koshelev, D.: Some remarks on how to hash faster onto elliptic curves (2021), https://eprint.iacr.org/2021/1082
29. Koshelev, D.: Generation of "independent" points on elliptic curves by means of Mordell–Weil lattices (2022), https://eprint.iacr.org/2022/794
30. Koshelev, D.: Indifferentiable hashing to ordinary elliptic $\mathbb{F}_q$-curves of $j = 0$ with the cost of one exponentiation in $\mathbb{F}_q$. Designs, Codes and Cryptography **90**(3), 801–812 (2022)
31. Koshelev, D.: Magma code (2022), https://github.com/dishport/Batch-point-compression-in-the-context-of-advanced-pairing-based-protocols
32. Koshelev, D.: Generation of two "independent" points on an elliptic curve of j-invariant $\neq 0, 1728$ (2023), https://eprint.iacr.org/2023/785
33. Lang, S.: Algebra, Graduate Texts in Mathematics, vol. 211. Springer, New York, 3 edn. (2002)
34. Müller, S.: On the computation of square roots in finite fields. Designs, Codes and Cryptography **31**(3), 301–312 (2004)
35. Oguiso, K., Truong, T.T.: Explicit examples of rational and Calabi–Yau threefolds with primitive automorphisms of positive entropy. Journal of Mathematical Sciences, the University of Tokyo **22**, 361–385 (2015)
36. 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)
37. Pornin, T.: Optimized binary GCD for modular inversion (2020), https://eprint.iacr.org/2020/972
38. Rubin, K., Silverberg, A.: Compression in finite fields and torus-based cryptography. SIAM Journal on Computing **37**(5), 1401–1428 (2008)
39. Sakemi, Y., Kobayashi, T., Saito, T., Wahby, R.S.: Pairing-friendly curves (2022), https://datatracker.ietf.org/doc/draft-irtf-cfrg-pairing-friendly-curves

40. Ueno, K.: Classification of algebraic varieties, I. Compositio Mathematica **27**(3), 277–342 (1973)
41. 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)

Appendix A. Compressing $E_b(\mathbb{F}_{q^2}) \times E_{b_2}(\mathbb{F}_q)$

Throughout the current supplementary section, we will assume that $q \equiv 1 \pmod 3$ or, equivalently, $\omega \in \mathbb{F}_q$. Unlike the main part of the paper, here the opposite situation would be drastically different as it becomes clear below. Given $\gamma \in \mathbb{F}_q^* \setminus (\mathbb{F}_q^*)^2$, let $b = b_0 + b_1\sqrt{\gamma}$ and $b_0, b_1, b_2 \in \mathbb{F}_q$ such that $bb_2 \neq 0$. Our goal is to simultaneously compress points $(x, y) = (x_0 + x_1\sqrt{\gamma}, y_0 + y_1\sqrt{\gamma})$ and (x_2, y_2) from the sets $E_b(\mathbb{F}_{q^2})$, $E_{b_2}(\mathbb{F}_q)$, respectively (here $x_j, y_j \in \mathbb{F}_q$). This problem is relevant for *pairing delegation* [2] and *type* 4 *pairings* [10, Section 3] whenever the embedding degree of the curve E_{b_2} is equal to 12. In this popular case, E_b is a sextic twist of E_{b_2} over the field $\mathbb{F}_{q^2}$. See [12, Section 3.2.5] to understand the significance of twists in pairing-based cryptography.

For compressing $E_b(\mathbb{F}_{q^2})$, it is suggested to apply the method from [27]. The given method extracts a cubic root in $\mathbb{F}_q$ in the decompression stage. Therefore, the concatenation of its result z_0, z_1 with x_2 gives rise to the compression method for $E_b(\mathbb{F}_{q^2}) \times E_{b_2}(\mathbb{F}_q)$ with the cost of a sextic root in $\mathbb{F}_q$, by analogy with compressing three $\mathbb{F}_q$-points in Section 4.

Table 4 exhibits a complexity comparison (all the operations are carried out in $\mathbb{F}_q$) of the compression-decompression methods for points in the projective or Jacobian coordinates. As is customary, the addition, subtraction, and multiplication operations in $\mathbb{F}_q$ are omitted, because they are much cheaper. We use the fact (e.g., from [12, Section 5.2.1]) that an inverse element (resp., square root) in $\mathbb{F}_{q^2}$ can be expressed via an inverse element (resp., two square roots) in $\mathbb{F}_q$. However, to the author's knowledge, a cubic root in $\mathbb{F}_{q^2}$ is not computed through a few radicals in $\mathbb{F}_q$. As a result, in comparison with Table 2, the new table does not contain the very slow methods with the coordinates y_0, y_1, x_2 or y_0, y_1, y_2.

The method of [27] is similar to the one of Sections 2, 3. It is based on $\mathbb{F}_q$-rationality of the surface

$$GK_b := \alpha(t)(y_0^2 + \gamma y_1^2 - b_0) - \beta(t)(2y_0y_1 - b_1) \quad \subset \quad \mathbb{A}^3_{(t,y_0,y_1)},$$

where $\alpha(t) := 3t^2 + \gamma$ and $\beta(t) := t(t^2 + 3\gamma)$. The latter is nothing but the generalized Kummer surface $R_b/[\omega]_2$ (up to a birational $\mathbb{F}_q$-isomorphism). Here,

$$R_b = \begin{cases} y_0^2 + \gamma y_1^2 = \rho_0 := x_0^3 + 3\gamma x_0 x_1^2 + b_0, \\ 2y_0y_1 = \rho_1 := \gamma x_1^3 + 3x_0^2 x_1 + b_1 \end{cases} \quad \subset \quad \mathbb{A}^4_{(x_0,x_1,y_0,y_1)}$$

is the *Weil restriction* (see, e.g., [38, Section 4]) of E_b, equipped with the $\mathbb{F}_q$-automorphism

$$[\omega]_2 \colon R_b \xrightarrow{\sim} R_b \qquad (x_0, x_1, y_0, y_1) \mapsto (\omega x_0, \omega x_1, y_0, y_1)$$

of order 3. Notice that

$$t = \frac{x_0}{x_1}, \qquad x_1 = \sqrt[3]{\frac{2y_0y_1 - b_1}{\alpha(t)}} = \sqrt[3]{\frac{y_0^2 + \gamma y_1^2 - b_0}{\beta(t)}}.$$

method	compression	decompression
classical with x_0, x_1, x_2	one inversion	three $\sqrt{\cdot}$
folklore-classical with x_0, x_1, y_2		one $\sqrt[6]{\cdot}$ and one $\sqrt{\cdot}$
new with z_0, z_1, x_2		one $\sqrt[6]{\cdot}$

Table 4. Worst-case complexity for compressing $\overline{E_b}(\mathbb{F}_{q^2})\times\overline{E_{b_2}}(\mathbb{F}_q)$ (with respect to the projective or Jacobian coordinates). All the methods compress to $\approx 3\lceil\log_2(q)\rceil$ bits.

Although [27] does not deal with the case $q \equiv 1 \pmod 4$ (including the BLS12-377 curve), it is not difficult to generalize the results to the given case if desired. We are not going to do this, because our purpose is opposite, namely to specify the $\mathbb{F}_q$-parametrization of GK_b as clearly as possible on the example of the BLS12-381 curve ($b_0 = b_1 = 4$ and $\gamma = -1$). That makes sense, since the description in [27, Section 3.1] is not sufficiently explicit.

First, $\sqrt{6} = \sqrt{-1}\cdot\sqrt{2}\cdot\sqrt{-3} \in \mathbb{F}_q$, because $\sqrt{-3} = 2\omega + 1 \in \mathbb{F}_q$, but $\sqrt{2} \notin \mathbb{F}_q$. Indeed, $4^2 \cdot 2$ is the norm of $b = 4(1 + \sqrt{-1})$ with respect to the extension $\mathbb{F}_{q^2}/\mathbb{F}_q$ and $\sqrt{b} \notin \mathbb{F}_{q^2}$ by virtue of [27, Remark 2]. Second, there is the birational $\mathbb{F}_q$-isomorphism

$$\tau\colon GK_b \xrightarrow{\sim} \mathbb{A}^2_{(z_0,z_1)} \qquad (t, y_0, y_1) \mapsto \Big(\frac{\mathrm{num}_{z_0}}{\mathrm{den}_z}, \frac{\mathrm{num}_{z_1}}{\mathrm{den}_z}\Big),$$

where

$$\mathrm{num}_{z_0} := f_0(t)y_0 + f_1(t)y_1, \qquad \mathrm{num}_{z_1} := -\sqrt{6}\cdot\alpha(t)(t^2 - 4t + 1),$$
$$\mathrm{den}_z := g_0(t)y_0 + g_1(t)y_1,$$

and

$$\begin{aligned}
f_0(t) &:= 6\big(\,(7\sqrt{6} - 13)t^3 - 13t^2 + (3\sqrt{6} - 1)t - 1\,\big),\\
f_1(t) &:= 3\sqrt{6}\cdot\alpha(t)\big(\,(\sqrt{6} - 3)t^2 + \sqrt{6}\cdot t - 1\,\big),\\
g_0(t) &:= 3\big(\,(\sqrt{6} + 2)t^4 + 2t^3 - 2(4\sqrt{6} - 5)t^2 + 10t - \sqrt{6}\,\big),\\
g_1(t) &:= 6\alpha(t)\big(\,(\sqrt{6} - 1)t - 1\,\big).
\end{aligned}$$

It turns out that

$$\tau^{-1}\colon \mathbb{A}^2_{(z_0,z_1)} \xrightarrow{\sim} GK_b \qquad (z_0, z_1) \mapsto \Big(\frac{\mathrm{num}_t}{\mathrm{den}_t}, \frac{\mathrm{num}_{y_0}}{\mathrm{den}_y}, \frac{\mathrm{num}_{y_1}}{\mathrm{den}_y}\Big),$$

where

$$\mathrm{num}_t := z_0^2 + 12z_1^2 - 1, \qquad \mathrm{den}_t := -2(z_0 + 6z_1^2), \qquad \mathrm{den}_y := -\sqrt{6}\cdot\alpha(t)(t^2+1),$$

$$\mathrm{num}_{y_0} := \alpha(t)\big(F_0(t)Z_0 + F_1(t)Z_1\big), \qquad \mathrm{num}_{y_1} := G_0(t)Z_0 + G_1(t)Z_1,$$

$$Z_0 := \frac{z_0\cdot\mathrm{den}_t + \mathrm{num}_t}{z_1\cdot\mathrm{den}_t}, \qquad Z_1 := \frac{1}{z_1}$$

and

$$\begin{aligned} F_0(t) &:= 2\big(\,(\sqrt{6}-1)t - 1\,\big), \qquad F_1(t) := (\sqrt{6}-4)t^2 - 4t + \sqrt{6},\\ G_0(t) &:= -(\sqrt{6}+2)t^4 - 2t^3 + 2(4\sqrt{6}-5)t^2 - 10t + \sqrt{6},\\ G_1(t) &:= (\sqrt{6}+2)t^5 + 2t^4 + 2(3\sqrt{6}-8)t^3 - 16t^2 + (5\sqrt{6}-2)t - 2. \end{aligned}$$

All the written formulas are checked in Magma, namely in [31]. As usual, to compress any points from $E_b(\mathbb{F}_{q^2})$ it remains to process the degenerate cases when the denominators equal zero. In order not to complicate the text this is left as an elementary exercise.

Appendix B. Compressing $E_b(\mathbb{F}_{q^2})$ sub-optimally in such a way that decompressing is for free

Let's stick to the notation of the previous section. This one contains formulas obtained in the same way as in [25, Section 3.1] for compressing $E_b^2(\mathbb{F}_q)$ to $\approx 3\lceil\log_2(q)\rceil$ bits. The new formulas are very simple and important, but the author did not find them anywhere else. So, the appendix is a good place to write out them. Probably, the similar approach from [25, Section 3.2] in the 3-dimensional case may be also adapted for compressing $E_b(\mathbb{F}_{q^2})\times E_{b_2}(\mathbb{F}_q)$ to $\approx 4\lceil\log_2(q)\rceil$ bits.

Given a non-zero point $P = (x, y) \in E_b(\mathbb{F}_{q^2})$, consider the $\mathbb{F}_q$-elements

$$Y := y_0 + y_1, \qquad Y_1 := \frac{2(\rho_0 - Y^2) + (\gamma+1)\rho_1}{2(\gamma-1)Y}.$$

Obviously, $\gamma \neq 1$. By looking at the defining equations of R_b, it is readily checked (see also [31]) that $y_1 = Y_1$ whenever $Y \neq 0$. Therefore, we get the compression map

$$\mathrm{com}\colon E_b(\mathbb{F}_{q^2}) \setminus \{\mathcal{O}\} \;\hookrightarrow\; \mathbb{F}_q^3 \times \{0,1\}$$

$$\mathrm{com}(P) := \begin{cases} (x_0, x_1, y_1, 0) & \text{if} \quad Y = 0,\\ (x_0, x_1, Y, 1) & \text{otherwise.} \end{cases}$$

The corresponding decompression map has the form

$$\mathrm{com}^{-1}\colon \mathrm{Im}(\mathrm{com}) \;\xrightarrow{\sim}\; E_b(\mathbb{F}_{q^2}) \setminus \{\mathcal{O}\}$$

$$\mathrm{com}^{-1}(x_0, x_1, Y', \mathrm{bit}) = \begin{cases} (x_0, x_1, -Y', Y') & \text{if} \quad \mathrm{bit} = 0, \\ (x_0, x_1, Y' - Y_1, Y_1) & \text{if} \quad \mathrm{bit} = 1. \end{cases}$$

Table 5 exhibits a complexity comparison (all the operations are carried out in $\mathbb{F}_q$) of the compression-decompression methods for $\mathbb{F}_{q^2}$-points on E_b. It is worth noting that all the remarks given for Table 4 still hold for the new one.

method	compression	decompression
classical with x_0, x_1	one inversion	two $\sqrt{\cdot}$
from [27]		one $\sqrt[3]{\cdot}$
from [26, Section 4] with $\frac{x_0}{x_1}$, $\frac{y_0}{y_1}$		one $\sqrt[6]{\cdot}$
new with x_0, x_1, Y		for free

Table 5. Worst-case complexity for compressing $\overline{E_b}(\mathbb{F}_{q^2})$ (with respect to the projective or Jacobian coordinates). The last method compresses to $\approx 3\lceil \log_2(q) \rceil$ bits and the other ones compress to $\approx 2\lceil \log_2(q) \rceil$ bits.