Point (de)compression for elliptic curves over highly 2-adic finite fields

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

University of Lleida, Department of Mathematics
Catalonia, Spain

Abstract. This article addresses the issue of efficient and safe (de)compression of $\mathbb{F}_q$-points on an elliptic curve E over a highly 2-adic finite field $\mathbb{F}_q$ of characteristic 5 or greater. The given issue was overlooked by cryptography experts, probably because, until recently, such fields were not in trend. Therefore, there was no difficulty (with rare exceptions) in finding a square $\mathbb{F}_q$-root. However, in our days, fields with large 2-adicities have gained particular popularity in the ZK (zero-knowledge) community, despite the fact that $\sqrt{\cdot} \in \mathbb{F}_q$ should be computed via more sophisticated square-root algorithms such as (Cipolla–Lehmer–)Müller's one. The article explains why the classical x-coordinate (de)compression method based on Müller's algorithm often contains Achilles' heel to successfully perform a novel fault attack, which also fits the definition of a (D)DoS attack. In a nutshell, the trouble stems from the non-deterministic initialization of Müller's algorithm.

Moreover, the article suggests a countermeasure, namely an alternative (still simple) (de)compression method that completely prevents the discovered attack whenever the curve $E/\mathbb{F}_q$ is of even order. In particular, all twisted Edwards (i.e., Montgomery) curves are relevant. The decompression stage of the new method equally suffers from one square-root extraction in $\mathbb{F}_q$. But the corresponding quadratic residue is inherently equipped with additional information, providing an opportunity to launch Müller's algorithm immediately from its main deterministic part. In turn, the compression stage of the new method remains (almost) free as well as for the x-coordinate method.

Keywords: Châtelet surfaces · Müller's square-root algorithm · conic bundles · (D)DoS attacks · elliptic curve cryptography · highly 2-adic finite fields · point (de)compression · quadratic covers · quartic polynomials

---

⋆ https://www.researchgate.net/profile/dimitri-koshelev
This paper is a part of the project "Avances en criptografía post-cuántica aplicados al desarrollo de un sistema de cupones", financed by "European Union NextGeneration–UE, the Recovery Plan, Transformation and Resilience". This research was also partially funded by the Spanish Ministry of Science, Innovation and Universities through the project PID2021-124613OB-I00.

1 Introduction

It is not a secret that ECC (elliptic curve cryptography) plays a significant role in today's digital security architecture. As usual, let $E : y^2 = f(x)$ be an elliptic curve in the Weierstrass form over a finite field $\mathbb{F}_q$ of characteristic > 3. To optimize the performance of cryptographic protocols, one often resorts to compressing $\mathbb{F}_q$-points $(x, y) \in E$. There is the classical method consisting in the projection $pr_x : E \to \mathbb{A}^1_x$ of a point to its x-coordinate. Also, one additional bit is introduced to distinguish the two square roots $y = \pm\sqrt{f(x)}$ in the decompression stage. The presence of this method in the NIST ECC standard [18, Appendix D.2.1] (among others) emphasizes its ubiquity. Analogous square-root (de)compression methods for other popular forms of E are specified, e.g., in [61, Appendix H].

As suggested already in (the last paragraph of) Miller's pioneering paper [46], some elementary cryptographic schemes (like Diffie–Hellman key exchange) can be deployed exclusively on the level of the x-coordinate. This happens when a scheme needs solely a scalar multiplication on E rather than the group operation $+$ itself. Therefore, such protocols can entirely avoid (de)compression. It is only sufficient to check that $f(x)$ is a square in $\mathbb{F}_q$ to thwart a secret key leakage on the quadratic twist of E. More details on the theme are found, e.g., in [26]. Nonetheless, more complicated protocols (even the standard ECDSA) truly require the two coordinates x, y, otherwise $+$ is not defined.

It is worth stressing that the unique bottleneck of the x-coordinate (de)compression method is extracting one square $\mathbb{F}_q$-root. The choice of how to compute this operation should depend on the 2-*adicity* ν of the field $\mathbb{F}_q$, i.e., on the 2-valuation of the number $q - 1$. For many decades real-world elliptic curves were mostly considered over finite fields with $\nu \leqslant 2$. As is well known (see, e.g., [38, Lemma 4]), a square root in $\mathbb{F}_q$ for such ν can be expressed via one exponentiation in $\mathbb{F}_q$ with at least ℓ and at most 2ℓ field multiplications, where $\ell := \lceil \log_2(q) \rceil$. Consequently, implementers of ECC were more or less satisfied by the speed.

Meanwhile, a lot of elliptic curves over *highly 2-adic fields* have emerged in practice in the last years. The point is that such curves are actively utilized in so-called *recursive* (*ZK-*)*SNARKs* (*zero knowledge succinct non-interactive argument of knowledge*) underlying advanced blockchain technologies. In principle, SNARKs can be deployed on conventional elliptic curves, but at the price of catastrophic slowdown. In this regard, curves over fields with large 2-adicities are sometimes called *SNARK-friendly*. An excellent survey of them is represented in [8].

It is impossible not to mention separately the curve NIST P-224 from [18, Section 3.2.1.2] for which $\nu = 96$. As far as the author knows, this is the only standardized curve (from around the world) over a highly 2-adic field. For this curve, finding $\sqrt{f(x)} \in \mathbb{F}_q$ is discussed in the patent [43] (with questionable legal force). It is based on *Müller's square-root algorithm* [49] improving earlier ones of Cipolla [21] and Lehmer [44]. Müller's original algorithm costs $\approx 2\ell$ multiplications in $\mathbb{F}_q$ regardless of ν, which is slightly slower than a general exponentiation in $\mathbb{F}_q$. In fact, a quite folklore trick employed in [33,43] diminishes

the number of multiplications to $\approx 2\ell - \nu$, which is essential speed up in the highly 2-adic setting.

The first step of Müller's algorithm (as opposed to its subsequent steps) has non-deterministic behavior. More precisely, the algorithm starts with searching for an element $u \in \mathbb{F}_q^*$ such that $u^2 - f(x)$ is a non-square in $\mathbb{F}_q$. For a random $x \in \mathbb{F}_q$ outside the roots of f, it is clearly enough (see [39, Lemma 1] to be sure) to iterate u on average solely two times until the desired condition holds. Moreover, quadratic (non-)residuosity of any element from $\mathbb{F}_q$ can be checked with a (sub-)quadratic bit complexity $O(\ell^2)$ according to today's state of the art (see, e.g., [9]). Since x is public information in $99,9\ldots\%$ of realistic scenarios, determinism is not an important property in the context of side-channel attacks. Thus, the x-coordinate method in combination with Müller's algorithm seems at first glance an universal (de)compression solution.

However, the above solution is still insecure concerning fault attacks. First, advanced protocols of ECC mostly operate with huge numbers of elliptic curve $\mathbb{F}_q$-points. Second, a malicious user can try to execute a *DoS* (*denial-of-service*) *attack* [32] by sending to a honest user a series of "poisoned" x-coordinates. More generally, a group of adversaries can cooperate with the aim to perform a *DDoS* (*distributed DoS*) *attack*. It turns out that there is a possibility for such a (D)DoS attack whenever the element u runs over $\mathbb{F}_q$ without strong pseudo-random generation. For example, it is periodically proposed in various sources to simply assign $u := u + 1$ if q is prime and it is even worse when u starts with a fixed value like 1.

Let's suppose that the malicious users know in advance the first $n \in \mathbb{N}$ values $u_i \in \mathbb{F}_q^*$ of the variable u that will be generated by the honest user. The former is invited to send to the latter a value $x \in \mathbb{F}_q$ such that all the $u_i^2 - f(x)$ are conversely squares in $\mathbb{F}_q$ (as well as $f(x)$). By the way, the topic partially resembles that [35] of consecutive tuples of quadratic residues for the aforementioned sequence $u_{i+1} = u_i + 1$. The larger n, the more powerful attack we have in the sense that Müller's algorithm slows down: The receiver of x performs n Legendre symbol $\left(\frac{\cdot}{q}\right)$ computations without suspecting that they are dummy. As said above, $\left(\frac{\cdot}{q}\right)$ is now recognized as a pretty quick primitive, but it is not free. Not to mention that its old implementations (being widely present on today's Internet) could not benefit from the modern Legendre symbol algorithms. Consequently, computing many instances of $\left(\frac{\cdot}{q}\right)$ significantly decelerates a cryptosystem, especially if one deals with low-resource devices. In time-critical situations even a short unexpected delay may be a reason of serious problems.

In the main part of this paper we will rigorously show that for fields $\mathbb{F}_q$ of cryptographic sizes (e.g., with $\ell \approx 128$), there are numerous $\mathbb{F}_q$-values x appropriate for instantiating efficiently the described attack under the condition that n is moderate (say, a few dozens). The difficulty of finding a "poisoned" x-coordinate apparently becomes considerable for large n. Hence, the attack is not catastrophic at first glance. Nevertheless, such destructive x-coordinates can be precomputed upfront and distributed (e.g., in the darknet among hackers) in hope to meet one day a vulnerable implementation of the x-coordinate method

based on Müller's algorithm. The fact is that the parameters of $E/\mathbb{F}_q$ are mainly fixed and published (e.g., in a standard), allowing to create once and for many years the corresponding "prohibited" database.

Of course, there are computer (i.e., not mathematical) instruments of protection against (D)DoS attacks. One of them is evidently an attempt to detect if the majority of incoming data is from the same sender(s) or not. In other words, the receiver can be configured in such a way that it accepts only a limited number of x-coordinates in an indicated period of time. This approach does not work if adversaries possess at least one destructive value x for the big number n found, e.g., on a supercomputer. In any case, an accurately planned (D)DoS attack can additionally exploit the discovered flaw (even for moderate n) in order to increase chances to overflow the receiver. Each security aspect may become decisive, so the new flaw also cannot be ignored.

Apart from Müller's algorithm, there is another classical square-root one invented independently by Tonelli [63] and Shanks [56]. The latter is deterministic in comparison with the former. Furthermore, in the last decades notable improvements of the algorithm occurred in [51,53,62], [57, Section 12.5.1], leading to reduction of its computation complexity. Nevertheless, it still amounts to $\Theta(\ell + g(\nu))$, where a function $g = O(\nu^2)$, but $g \neq O(\nu)$. In other words, TS (*Tonelli–Shanks's*) *algorithm* and even its modifications are an order of magnitude slower (as $\nu \to \ell$) than Müller's one. It is not a secret that TS-type algorithms have table-lookup versions, which follow Bernstein's idea [15] developed for the original TS algorithm and tested on the base field of the curve NIST P-224. This "cheating" helps to make the complexity almost linear in ℓ by introducing an auxiliary parameter μ, but at the price of an exponential growth (as $\mu \to \nu$) of required memory. Thus, table lookups are also not a panacea for very large ν.

The goal of the present article is twofold. On the one hand, it is vitally important to highlight the existence of the described (D)DoS attack. Curiously, it has never been represented in the open literature to the author's knowledge. Before this work the generation problem of the values u_i appeared inessential. On the other hand, we will provide a safe and efficient novel point (de)compression method relevant for a wide class of elliptic curves. In a nutshell, the method makes Müller's algorithm deterministic, noticing the fact that the square root has to be extracted not from an abstract quadratic residue, but from an element having some geometric nature.

The new approach of compressing $P \in E(\mathbb{F}_q)$ is based on a simple proposal to take the image $t := \varphi(P)$ of a certain quadratic $\mathbb{F}_q$-cover $\varphi : E \to \mathbb{A}^1_t$ different from pr_x. By analogy with the x-coordinate method, decompressing, i.e., determining the inverse image $P \in \varphi^{-1}(t)$ requires finding some square $\mathbb{F}_q$-root, namely $\sqrt{h(t)}$ for a quartic polynomial $h \in \mathbb{F}_q[t]$ such that E is birationally $\mathbb{F}_q$-isomorphic to the curve $s^2 = h(t)$. It turns out that for the properly chosen φ (or, equivalently, h), Müller's algorithm obtains for free a necessary value $u \in \mathbb{F}_q$ such that $u^2 - h(t)$ is a non-square in $\mathbb{F}_q$. To be more precise, u is explicitly expressed through t by means of some rational $\mathbb{F}_q$-function. Looking

ahead, the given trick works if and only if the number of all $\mathbb{F}_q$-points on E is even. In particular, this condition is fulfilled for *twisted Edwards* (up to a birational $\mathbb{F}_q$-isomorphism, *Montgomery*) *curves* as well as for *double-odd curves* [52]. Remarkable ones $y^2 = x^3 + ax$ (for $a \in \mathbb{F}_q^*$) of j-invariant 1728 are evidently involved.

Obviously, a cryptographically strong pseudo-random generator for obtaining the values u_i is capable to thwart the aforementioned attack. The literature about such generators is vast (see, e.g., the corresponding NIST standard [11]) and they are regularly utilized in diverse cryptographic situations. Nonetheless, reliance on them has a series of disadvantages in comparison with the new point (de)compression. First, there is no guarantee that a chosen generator is actually pseudo-random, because it may suffer from potential issues listed in [3]. There are (e.g., in [4]) many sad public stories of breaking a cryptosystem not paying due attention to reliable generation of secret information. Second, provably secure generators (inter alia, founded on elliptic curves [58]) are quite slow, possibly even slower than square-root extraction itself. Third, a non-experienced implementer may merely forget to leverage a cryptographic generator. Meanwhile, the current article solution is laconic and apparently immune from any conceivable flaws.

Finally, it is worth adding that highly 2-adic fields are painful for ECC in many aspects, not only during decompression. For instance, [50, Section 3.3] tries to exploit smoothness of $q-1$ to construct a faster index calculus algorithm of solving the DLP (discrete logarithm problem) in $E(\mathbb{F}_q)$. However, the given approach seemingly is not better than general DLP algorithms. Furthermore, the high 2-adicity of $\mathbb{F}_q$ substantially complicates constant-time hashing to elliptic curves, a task dual in a sense to point compression. Hashing of this type for large ν is first discussed in [31] and then in [37,40], [42, Section 4]. Interestingly, the hash function from [40] (valid for all elliptic curves of j-invariant $\neq 0$, 1728) is constructed on mathematical apparatus similar to that of the present article.

2 Algebraic geometry preliminaries

Throughout the current section, occurring formulas are tacitly checked in the computer algebra system Magma by launching the code [41].

2.1 A complete intersection curve

During the whole article, we will deal with a finite field $\mathbb{F}_q$ of characteristic > 3. Given values $c_i \in \mathbb{F}_q^*$ such that $c_i \neq c_j$ for $i < j \leqslant n \in \mathbb{N}$, consider the polynomial system

$$I := \left\{y^2 = f(x)\right\} \cap I_n \quad \subset \quad \mathbb{A}^{n+2}_{(x,y,z_1,\cdots,z_n)}, \tag{1}$$

where $f := x^3 + ax + b \in \mathbb{F}_q[x]$ and

$$I_j := \left\{c_i - y^2 = z_i^2\right\}_{i=1}^{j} \quad \subset \quad \mathbb{A}^{j+1}_{(y,z_1,\cdots,z_j)}$$

is the intersection of $j \leqslant n$ diagonal quadrics of the same type.

Lemma 1. *The algebraic set I is an absolutely (i.e., geometrically) irreducible curve, that is, a complete intersection. Furthermore, I is of arithmetic genus $p_a = 1 + 2^{n-1}3n$.*

Proof. The system I can be imagined as the termination of adjoining recursively (starting with I_1) the new variables $z_2, \cdots, z_n$, and ultimately x. Since $I_1 \subset \mathbb{A}^2_{(y,z_1)}$ is a curve (namely a diagonal conic), so is I as the "top floor" of the curve tower. Moreover, I_j are absolutely irreducible curves, because (under our restrictions on c_j) the conic I_1 is non-degenerate and $c_j - y^2$ is clearly a quadratic non-residue in the function field $\overline{\mathbb{F}_q}(I_{j-1})$. To avoid uncertainty, $\overline{\mathbb{F}_q}$ is the algebraic closure of $\mathbb{F}_q$.

It remains to show irreducibility over $\overline{\mathbb{F}_q}$ of I itself, i.e., irreducibility over $\overline{\mathbb{F}_q}(I_n)$ of the cubic polynomial $F := f - y^2 = x^3 + ax + B$ in the variable x, where $B := b - y^2$. It is suggested to apply *Eisenstein's irreducibility theorem* [60, Proposition 3.1.15.(2)] with respect to some smooth point P_n on the projective closure of I_n. It is convenient to change the current affine chart to take such a point at infinity. In this way,

$$I_j = \left\{g_i(z_0, z_i) = 0\right\}_{i=1}^{j} \quad \subset \quad \mathbb{A}^{j+1}_{(z_0,\cdots,z_j)},$$

where $g_i := c_i z_0^2 - 1 - z_i^2$. On the new chart $y \neq 0$ the function $B = (bz_0^2 - 1)/z_0^2$. Let's pick the point $P_j := (0, \sqrt{-1}, \cdots, \sqrt{-1})$ on the curve I_j.

The points $P_j \in I_j$ are smooth as readily follows from the *Jacobian criterion* [28, Section I.5]. Indeed, the Jacobian matrix of I_j (with j rows and $j+1$ columns) is equal to

$$\left(\frac{\partial g_i}{\partial z_k}\right)_{i,k} = 2 \cdot \begin{pmatrix} c_1 z_0 & -z_1 & 0 & 0 & \cdots & 0 \\ c_2 z_0 & 0 & -z_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ c_j z_0 & 0 & 0 & 0 & \cdots & -z_j \end{pmatrix},$$

where the entry at the intersection of the i-th row and k-th column contains the partial derivative $\dfrac{\partial g_i}{\partial z_k}$. This matrix is of full rank j after the substitution into it of the point P_j.

Due to smoothness of P_j, there is the notion of the valuation (a.k.a. order) $\nu_{P_j} : \overline{\mathbb{F}_q}(I_j)^* \to \mathbb{Z}$ at the given point. It is necessary to compute $\nu_{P_n}(B) = -2 \cdot \nu_{P_n}(z_0)$. First, $\nu_{P_j}(z_0) = \nu_{P_{j-1}}(z_0)$, because P_j is not a ramification point of the quadratic cover $I_j \to I_{j-1}$ (the projection to $z_0, \cdots, z_{j-1}$). Second, $\nu_{P_1}(z_0) = 1$, because the line $L_{z_0} : z_0 = 0$ on $\mathbb{A}^2_{(z_0,z_1)}$ (passing through P_1) is different from the tangent $T_{P_1} : z_1 = \sqrt{-1}$ to the curve I_1 at P_1. In other words, the lines L_{z_0} and T_{P_1} intersect at P_1 transversely. Thus, z_0 is a uniformizing (a.k.a. local) parameter at all the points $P_j \in I_j$ and so $\nu_{P_n}(B) = -2$. Eisenstein's irreducibility theorem is applicable, since a is a constant (i.e., $a = 0$ or $\nu_{P_n}(a) = 0$) and $\deg(F) = 3$ is relatively prime with $\nu_{P_n}(B)$.

Finally, the arithmetic genus of I is computed by means of [55, Equality IV.(18)]. □

2.2 Quartic models of elliptic curves

Consider an elliptic $\mathbb{F}_q$-curve

$$E\colon y^2 = f(x) := x^3 + ax + b \quad \subset \quad \mathbb{A}^2_{(x,y)}$$

equipped with the infinity (i.e., zero) point $\mathcal{O} := (0 : 1 : 0)$. As always, $D_f := -16(4a^3 + 27b^2) \neq 0$ is the discriminant of f (and E). Among other things, we will need the exponent e of the group $E(\mathbb{F}_q)[2^\infty]$, that is, the number

$$e := \max\left\{n := \mathrm{ord}(P) \mid P \in E(\mathbb{F}_q), \log_2(n) \in \mathbb{N}\right\}. \tag{2}$$

The next lemma is folklore, but the author did not encounter a reference, hence its proof is included.

Lemma 2. *An arbitrary $\mathbb{F}_q$-involution ι on E has one of the two incompatible forms:*

1. *the translation map τ_P with $P \in E(\mathbb{F}_q)[2]$,*
2. *the composition $\tau_P \circ [-1]$ with any $P \in E(\mathbb{F}_q)$.*

Moreover, the quotient E/ι is an elliptic $\mathbb{F}_q$-curve in the first case and $\mathbb{P}^1$ in the second one.

Proof. It is not a secret that every $\mathbb{F}_q$-automorphism on E is the composition $\iota = \tau_P \circ \alpha$ of an $\mathbb{F}_q$-automorphism α fixing $\mathcal{O}$ and of the translation τ_P with respect to a point $P \in E(\mathbb{F}_q)$. Undoubtedly, the maps mentioned in the lemma are $\mathbb{F}_q$-involutions on E. Let's prove that there no others. We know that on each elliptic curve there is only one non-trivial involution leaving $\mathcal{O}$ in place, namely $[-1]$. Thereby, it is sufficient to show that α is an involution whenever so is ι. Notice that $\tau_{\alpha(P)} \circ \alpha = \alpha \circ \tau_P$ and hence $[1] = \iota^2 = \tau_{P+\alpha(P)} \circ \alpha^2$. Meanwhile, $\alpha(\mathcal{O}) = \mathcal{O}$, which implies the equality $P + \alpha(P) = \mathcal{O}$, that is, $\tau_{P+\alpha(P)} = [1]$.

Furthermore, $E/\tau_P = E/P$ is just E or a 2-isogenous elliptic $\mathbb{F}_q$-curve if P is a 2-torsion $\mathbb{F}_q$-point. In the second case, ι has exactly 4 fixed points over $\overline{\mathbb{F}_q}$, namely the elements of the set $[2]^{-1}(P)$. They are nothing but the ramification points of the quadratic quotient $\mathbb{F}_q$-cover $E \to E/\iota$. First of all, E/ι is a smooth absolutely irreducible $\mathbb{F}_q$-curve as stated in any detailed source (like [55, Section III.12]) on quotient varieties. Lastly, by virtue of the *Riemann–Hurwitz formula* [28, Corollary IV.2.4], the genus of E/ι is actually zero. □

Theorem 1. *The following conditions are equivalent:*

1. *There is a quartic $\mathbb{F}_q$-polynomial h without $\mathbb{F}_q$-roots such that E is birationally $\mathbb{F}_q$-isomorphic to the curve $C\colon s^2 = h(t)$;*
2. *The curve E admits a quadratic $\mathbb{F}_q$-cover $\varphi\colon E \to \mathbb{P}^1$ without ramification $\mathbb{F}_q$-points;*

3. The group order $\#E(\mathbb{F}_q)$ is even.

Proof. The first equivalence is straightforward if the reader is aware of the basic function field theory (as in the beginning of [60]). Indeed, the presence of a quadratic $\mathbb{F}_q$-cover $\varphi : E \to \mathbb{P}^1 \supset \mathbb{A}^1_t$ precisely means that $\mathbb{F}_q(E) \simeq \mathbb{F}_q(t)(s)$, where $s := \sqrt{h(t)}$ for an appropriate square-free $\mathbb{F}_q$-polynomial h. Moreover, the ($\mathbb{F}_q$-)roots of h naturally correspond to the affine ramification ($\mathbb{F}_q$-)points of φ. Since the genus $g(E) = 1$, the degree of h is either 3 or 4 in accordance with the Riemann–Hurwitz formula. But the case $\deg(h) = 3$ is not admissible, because then $\mathcal{O}$ is a ramification $\mathbb{F}_q$-point of φ over the infinity point $(1 : 0) \in \mathbb{P}^1 \setminus \mathbb{A}^1_t$.

The second equivalence is not much more difficult. Truly, since an arbitrary quadratic $\mathbb{F}_q$-cover $\varphi\colon E \to \mathbb{P}^1$ is Galois, it is necessarily obtained as the quotient cover under the action of some $\mathbb{F}_q$-involution ι on E. The previous lemma says that $\iota = \tau_P \circ [-1]$ for a certain $P \in E(\mathbb{F}_q)$ and

$$\mathrm{Fix}(\iota) = \mathrm{Ram}(\varphi) = [2]^{-1}(P). \tag{3}$$

If the order $\#E(\mathbb{F}_q)$ is odd (i.e., the exponent $e = 1$), then the doubling map [2] is a bijection on $E(\mathbb{F}_q)$. Therefore, the above set contains exactly one $\mathbb{F}_q$-point regardless of P. To put it another way, it is impossible to construct a desired $\mathbb{F}_q$-cover φ. Conversely (i.e., in the case of even group order), for the role of P one should choose any $\mathbb{F}_q$-point on E of order $e > 1$. □

For convenience, an $\mathbb{F}_q$-point on E will be further denoted by $P_0 = (x_0, y_0)$ rather than P as earlier. Let's borrow explicit formulas from [7, Section 2] associated with P_0. We have the quartic $\mathbb{F}_q$-curve

$$C\colon s^2 = h(t) := t^4 - 6x_0t^2 - 8y_0t - (4a + 3x_0^2) \quad \subset \quad \mathbb{A}^2_{(t,s)} \tag{4}$$

and the birational $\mathbb{F}_q$-map

$$\psi\colon E \to C \qquad (x, y) \mapsto \begin{cases} t := \dfrac{y + y_0}{x - x_0}, \\ s := 2x - t^2 + x_0 \end{cases} \tag{5}$$

whose inverse is

$$\psi^{-1}\colon C \to E \qquad (t, s) \mapsto \begin{cases} x := \dfrac{t^2 + s - x_0}{2}, \\ y := (x - x_0)t - y_0. \end{cases} \tag{6}$$

For information, the points P_0, $\mathcal{O}$ are mapped by ψ to the two distinct infinity points $(0 : 1 : \pm 1)$ of the (smooth) closure $\overline{C}\colon S^2 = h^*(R, T)$ of the curve C in the weighted projective plane $\mathbb{P}(1, 1, 2)$ with the variables R, T, S, respectively. Here, h^* is the homogenization of h and $t = T/R$, $s = S/R^2$. In [27, Section 10.1.1] the curve C (or $\overline{C}$) is referred to as a *split* (a.k.a. *real*) *model* of E.

The discriminant of the polynomial h is equal to $D_h = 2^8 D_f$. Surprisingly, it is independent of the point P_0. Inter alia, the Legendre symbol $L := \left(\frac{D_h}{q}\right) =$

$\left(\frac{D_f}{q}\right)$. Owing to *Stickelberger's parity theorem* [22, Section 4], the number of $\mathbb{F}_q$-irreducible factors of h (resp., f) is even (resp., odd) if and only if $L = 1$.

From now on, suppose for simplicity that the cardinality $\#E(\mathbb{F}_q)$ is even, hence we are in the conditions of Theorem 1. This decision loses generality in the below reasoning, but this is enough for our future purposes. The upcoming lemma demonstrates that the quadratic $\mathbb{F}_q$-cover $\varphi = pr_t \circ \psi$ complies with P_0 in the sense that the right-hand equality (3) is satisfied.

Lemma 3. *The function $\theta : t \mapsto (t^2 - x_0)/2$ is surjective from the set of ($\mathbb{F}_q$-)roots of the polynomial h onto the set of the x-coordinates of ($\mathbb{F}_q$-)points in $[2]^{-1}(P_0)$. If on top of that $y_0 \neq 0$, then θ is even bijective.*

Proof. The curve E disposes at least one $\mathbb{F}_q$-point $P_1 = (x_1, 0)$ of order 2. Let's shift it to the origin $O := (0, 0)$ lying on the elliptic curve $E' : y^2 = xg(x)$ whose defining polynomial $g := x^2 + a_2x + a_4$ has the coefficients $a_2 := 3x_1$ and $a_4 := 3x_1^2 + a = f'(x_1)$. We deal with the variable change

$$\begin{aligned} &\tau : E \to E' && (x, y) \mapsto (x - x_1, y), \\ &\tau^{-1} : E' \to E && (x, y) \mapsto (x + x_1, y). \end{aligned}$$

Among other things, $P'_0 := \tau(P_0)$ is the point with the coordinates $x'_0 := x_0 - x_1$ and y_0.

Due to [47, Equation (5)], the x-coordinates of the ($\mathbb{F}_q$-)points from $[2]^{-1}(P'_0)$ are the ($\mathbb{F}_q$-)roots of the $\mathbb{F}_q$-polynomial

$$p(x) := x^4 - 4x'_0x^3 - 2(2a_2x'_0 + a_4)x^2 - 4a_4x'_0x + a_4^2.$$

Hence, the said phrase holds true after the replacement simultaneously of P'_0 by P_0 and of $p(x)$ by $p(x - x_1)$. It turns out that $p(\theta(t_0) - x_1) = 0$ whenever $h(t_0) = 0$ for $t_0 \in \overline{\mathbb{F}_q}$. In addition, θ is obviously defined over $\mathbb{F}_q$ and $f(\theta(t_0))$ is a square in $\mathbb{F}_q$ once $t_0 \in \mathbb{F}_q$, since the coordinate $y_0 \in \mathbb{F}_q$ as well. In other terms, θ is actually a function with the (co)domain declared in the lemma.

In the degenerate case $y_0 = 0$, we can take $P_1 = P_0$, i.e., $x_1 = x_0$ and therefore $P'_0 = O$, i.e., $x'_0 = 0$. As a consequence, $p = x^4 - 2a_4x^2 + a_4^2 = (x^2 - a_4)^2$, that is, the roots of p are $\pm\sqrt{a_4} \neq 0$ and each one is of double multiplicity. In turn, h is a quartic polynomial without multiple roots, because its discriminant $D_h \neq 0$. Since θ is a quadratic function, we thus see that θ is surjective for the declared (co)domain.

It remains to demonstrate that θ is a bijective function if $y_0 \neq 0$, despite the fact that it is quadratic (so non-injective) on abstract elements of $\mathbb{F}_q$ or $\overline{\mathbb{F}_q}$. Assume the contrary: $h(t_0) = h(-t_0) = 0$ for $t_0 \neq 0$. Then, $16y_0t_0 = 0$, that is, $y_0 = 0$, which yields the injectivity of θ. The surjectivity (or, equivalently, bijectivity) of θ follows from the coincidence $\deg(p) = \deg(h)$ and from the absence of multiple roots for h (and hence for p). □

The fundamental cause why we were obliged to treat the case $y_0 = 0$ in another way consists in the fact that the set of the (order-4) points "hanging"

over $P_0 = (x_0, 0)$ is invariant under the canonical involution $[-1] : (x, y) \mapsto (x, -y)$ on E. Evidently, this circumstance happens solely when P_0 is a 2-torsion point.

Corollary 1. *The polynomial h of the equation (4) does not admit $\mathbb{F}_q$-roots if and only if $[2]^{-1}(P_0) \cap E(\mathbb{F}_q) = \emptyset$. For instance, this takes place if the point P_0 is of order $e > 1$.*

Suppose that the premise of this corollary is fulfilled. The cornerstone point P_0 can be found with the help of [47]. By our assumption, f possesses at least one $\mathbb{F}_q$-root (labeled x_1 in the proof of the anterior lemma). At the same time, h does not have $\mathbb{F}_q$-roots. As a result, we get the next remark.

Remark 1. There are two mutually exclusive possibilities:

1. The polynomial h is the product of two $\mathbb{F}_q$-irreducible quadratic ones. Equivalently, $L = 1$, that is, f has three $\mathbb{F}_q$-roots;
2. The polynomial h is $\mathbb{F}_q$-irreducible, i.e., the product of two $\mathbb{F}_{q^2}$-irreducible $\mathbb{F}_q$-conjugate quadratic ones. Equivalently, $L = -1$, that is, f has a unique $\mathbb{F}_q$-root.

To complete the picture, it is worthwhile to analyze the last remark in terms of the quadratic polynomial g from the proof of Lemma 3. As always, $D_g := a_2^2 - 4a_4 \neq 0$ is the discriminant of g. Meanwhile, the discriminant $D_f = 2^4 a_4^2 D_g$ and thereby $L = \left(\frac{D_g}{q}\right)$. This means that g splits over $\mathbb{F}_q$ if and only if we are in the case 1. Vice versa, g is $\mathbb{F}_q$-irreducible if and only if the case 2 is met.

The case $e = 2$. This tiny section aims to illustrate the theory developed above in the most elementary scenario when $e = 2$. We are given an order-2 point $P_0 = P_1 = (x_0, 0) \in E(\mathbb{F}_q)$. The right-hand side of the equation (4) with respect to P_0 has the biquadratic shape $h = t^4 - 2a_2 t^2 + D_g$. Interestingly, the curve $C\colon s^2 = h(t)$ and the maps $\psi^{\pm 1}$, i.e., (5), (6) have the identical formulas as in [17, Section 3] (cf. [52, Section 4.1]), putting there $X = 2$, $Y = s$, and $Z = t$. In that source, C is dubbed as *extended Jacobi quartic* and it is also analyzed in the scope of ECC.

It is readily seen that $16a_4$ is the discriminant of the quadratic polynomial $h(\sqrt{t})$. Therefore, $h(\sqrt{t})$ (and hence $h(t)$) has no $\mathbb{F}_q$-roots provided that $\sqrt{a_4} \notin \mathbb{F}_q$. Although in general, the polynomial h may still have no $\mathbb{F}_q$-roots, that is, $[2]^{-1}(P_0) \cap E(\mathbb{F}_q) = \emptyset$ even if $\sqrt{a_4} \in \mathbb{F}_q$. By virtue of [47, Lemma 2], the equivalence nonetheless holds subject to the restriction $L = -1$. Owing to [22, Theorem 3], we come to exhaustive criterions. The case 1 of Remark 1 takes place if and only if $L = \left(\frac{a_4}{q}\right) = -\left(\frac{d}{q}\right) = 1$, where $d := 2(a_2 - \sqrt{D_g})$, or, as an alternative, $L = -\left(\frac{a_4}{q}\right) = 1$. In turn, the case 2 arises, i.e., E is a double-odd curve if and only if $L = \left(\frac{a_4}{q}\right) = -1$.

The case $b = a_2 = x_0 = 0$ *and* $a = a_4 \neq 0$. This peculiar situation precisely corresponds to the elliptic curve E of j-invariant 1728. In the polynomial language, $g = x^2 + a$, $f = xg$, and $h = t^4 - 4a$. Besides, $D_g = -4a$ and so $L = \left(\frac{-a}{q}\right)$, while $d = -4\sqrt{-a}$. Understandably, the case 1 of Remark 1 appears if and only if $\left(\frac{-1}{q}\right) = \left(\frac{a}{q}\right) = -\left(\frac{\sqrt{-a}}{q}\right) = 1$ or, as the second option, $\left(\frac{-1}{q}\right) = \left(\frac{a}{q}\right) = -1$. In conclusion, the case 2 holds if and only if $\left(\frac{-1}{q}\right) = -\left(\frac{a}{q}\right) = 1$. Recall that $\left(\frac{-1}{q}\right) = 1$ automatically if E is an ordinary (a.k.a. non-supersingular) elliptic curve.

2.3 (Generalized) Châtelet surfaces

Let $c \in \mathbb{F}_q^*$ be a fixed quadratic non-residue and $h \in \mathbb{F}_q[t]$ be a square-free non-zero polynomial. A *generalized Châtelet surface* is given by the equation

$$S_h \colon u^2 - cv^2 = h(t) \quad \subset \quad \mathbb{A}^3_{(u,v,t)}.$$

It possesses the natural *conic bundle* structure $pr_t : S_h \to \mathbb{A}^1_t$. By abuse of notation, S_h can be interpreted as a non-degenerate conic in $\mathbb{A}^2_{(u,v)}$ over the function field $\mathbb{F}_q(t)$.

The next theorem should be well known to someone among algebraic geometers. Nevertheless, its proof is included for convenience of other readers.

Theorem 2. *The conic bundle pr_t enjoys an $\mathbb{F}_q$-section $\mathbb{A}^1_t \to S_h$, that is, $S_h \subset \mathbb{A}^2_{(u,v)}$ does an $\mathbb{F}_q(t)$-point if and only if h does not admit over $\mathbb{F}_q$ divisors of odd degrees.*

Proof. According to [54, Theorem 3.6] (see also [36, Lemma 5]), there is an $\mathbb{F}_q$-section of pr_t if and only if c is a square in $\mathbb{F}_q[t]/h$. Assume that $h = h_1 \cdots h_k$ is the decomposition of h to its $\mathbb{F}_q$-irreducible components of degrees $d_i := \deg(h_i)$. Since the polynomial h is without multiple roots, $h_i \neq h_j$ when $i \neq j$. By virtue of the *Chinese remainder theorem*, we have the ring isomorphisms

$$\mathbb{F}_q[t]/h \;\simeq\; \mathbb{F}_q[t]/h_1 \oplus \cdots \oplus \mathbb{F}_q[t]/h_k \;\simeq\; \mathbb{F}_{q^{d_1}} \oplus \cdots \oplus \mathbb{F}_{q^{d_k}}.$$

Clearly, the image of c in the right-hand side ring is nothing but the vector $(c, \cdots, c)$ of length k. It is a square in that ring if and only if c is a square in every field $\mathbb{F}_{q^{d_i}}$. This is equivalent to the fact that all the degrees d_i are even. Since h_i are the unique "building blocks" for divisors of h over $\mathbb{F}_q$, the theorem is proved. □

By definition, a *Châtelet surface* [20] is any surface of the form S_h provided that the polynomial h is of degree 3 or 4.

Corollary 2. *The conic bundle pr_t on a Châtelet surface enjoys an $\mathbb{F}_q$-section if and only if h is a quartic polynomial without $\mathbb{F}_q$-roots.*

To establish explicit formulas for an $\mathbb{F}_q$-section of pr_t, we lack several auxiliary statements about *monoidal transformations* [28, Section V.3] respecting the conic bundle structure.

Lemma 4. *Suppose that c is conversely a non-zero quadratic residue in $\mathbb{F}_q$. Regardless of an $\mathbb{F}_q$-polynomial $g \neq 0$, we get the blow-up $\mathbb{F}_q$-maps*

$$bl_{g,\pm}\colon S_h \to S_{hg} \qquad (u,v) \mapsto \Big(\frac{(1+g)u \pm \sqrt{c}(1-g)v}{2},\ \frac{(1-g)u \pm \sqrt{c}(1+g)v}{\pm 2\sqrt{c}}\Big)$$

identical on t. They are linear transformations with determinants equal to g.

Proof. Introduce the additional $\mathbb{F}_q$-surfaces

$$\begin{aligned} &S'_h\colon u^2 - v^2 = h(t),\\ &T_h\colon uv = h(t) \end{aligned} \qquad \subset \quad \mathbb{A}^3_{(u,v,t)}.$$

There are the birational $\mathbb{F}_q$-maps

$$\chi_{h,\pm}\colon S_h \to S'_h \qquad (u,v) \mapsto (u, \pm\sqrt{c}\cdot v),$$
$$\chi^{-1}_{h,\pm}\colon S'_h \to S_h \qquad (u,v) \mapsto \Big(u, \frac{v}{\pm\sqrt{c}}\Big)$$

as well as

$$\phi_h\colon S'_h \to T_h \qquad (u,v) \mapsto (u+v, u-v),$$
$$\phi^{-1}_h\colon T_h \to S'_h \qquad (u,v) \mapsto \Big(\frac{u+v}{2}, \frac{u-v}{2}\Big),$$

and ultimately

$$pr_{u,h}\colon T_h \to \mathbb{A}^2_{(u,t)} \qquad (u,v) \mapsto u,$$
$$pr^{-1}_{u,h}\colon \mathbb{A}^2_{(u,t)} \to T_h \qquad u \mapsto \Big(u, \frac{h}{u}\Big).$$

All these maps leave the variable t in place, hence it is omitted in their definitions for conciseness. It is straightforward to verify that the composition

$$bl_{g,\pm} := \chi^{-1}_{hg,\pm} \circ \phi^{-1}_{hg} \circ pr^{-1}_{u,hg} \circ pr_{u,h} \circ \phi_h \circ \chi_{h,\pm}$$

has the indicated formulas and determinant g. $\square$

Corollary 3. *Assume that a quartic $\mathbb{F}_q$-polynomial g is irreducible and monic (for simplicity). Evidently, $g = g_1 g_2$, where $g_i = t^2 - T_i t + N_i$ are irreducible quadratic $\mathbb{F}_{q^2}$-polynomials conjugate over $\mathbb{F}_q$. We have the blow-up $\mathbb{F}_q$-map*

$$bl_g = bl_{g_2,-} \circ bl_{g_1,+}\colon S_h \to S_{hg} \qquad (u,v) \mapsto \Big(\frac{\varrho(t)u + \rho(t)v}{2},\ \frac{\rho(t)u + c\varrho(t)v}{2c}\Big)$$

identical on t, where

$$\rho := \sqrt{c}\big((T_1 - T_2)t - (N_1 - N_2)\big), \qquad \varrho := 2t^2 - (T_1 + T_2)t + (N_1 + N_2).$$

This is a linear transformation with determinant equal to g. Inter alia, it is invertible for every $t \in \mathbb{F}_q$.

Hereafter, h is suggested to be a monic quartic $\mathbb{F}_q$-polynomial without $\mathbb{F}_q$-roots. Thereby, S_h is a (usual) Châtelet surface whose conic bundle pr_t possesses an $\mathbb{F}_q$-section. According to Remark 1, there are the two possibilities 1, 2 for decomposing h. In each of them it is possible to completely eliminate h over $\mathbb{F}_q$, coming to the equation $S_1 : u^2 - cv^2 = 1$ independent of the variable t. In the case 1, one applies twice [40, Lemma 1] to separately get rid of the quadratic $\mathbb{F}_q$-factors h_1, h_2 of the polynomial h, while in the case 2, the previous corollary comes to the fore ($h_i = g_i$). In both cases, we deal with the blow-up $\mathbb{F}_q$-map $bl_h \colon S_1 \to S_h$ of the shape $bl_h = bl_{h_2,-} \circ bl_{h_1,+}$.

Notice that the $\mathbb{F}_q$-conic $S_1 \subset \mathbb{A}^2_{(u,v)}$ enjoys the $\mathbb{F}_q$-point $Q_0 = (1, 0)$. As always, one can involve the projection from this point to some line. It is for example about

$$pr_{Q_0} \colon S_1 \to \mathbb{A}^1_r \qquad (u, v) \mapsto \frac{cv}{1 - u}.$$

It is readily verified (cf. [40, Section 2]) that the map inverse to pr_{Q_0} is given by the formulas

$$pr^{-1}_{Q_0} \colon \mathbb{A}^1_r \to S_1 \qquad r \mapsto \Big(\frac{r^2 + c}{r^2 - c},\ \frac{-2r}{r^2 - c}\Big).$$

By the way, this map is correctly defined for all $r \in \mathbb{F}_q$, because $\sqrt{c} \notin \mathbb{F}_q$ by our assumption. As before, $pr^{-1}_{Q_0} : \mathbb{A}^2_{(r,t)} \to S_1$ can be equally interpreted as a map to the quadratic cone $S_1 \subset \mathbb{A}^3_{(u,v,t)}$.

To sum up, we obtain the rational proper (i.e., invertible) $\mathbb{F}_q$-parametrization

$$\pi := bl_h \circ pr^{-1}_{Q_0} \colon \quad \mathbb{A}^2_{(r,t)} \to S_h$$

without indefiniteness points in $\mathbb{F}_q^2$. And its restriction to the diagonal $\Delta \colon r = t$ gives rise to the desired $\mathbb{F}_q$-section $\sigma := \pi\,|_{\Delta} \colon \mathbb{A}^1_t \to S_h$.

3 Cryptographic applications

We will continue to stick everywhere below to the notation of Section 2.

3.1 The (D)DoS attack on elliptic curves over highly 2-adic fields

As said in the introduction, given disclosed values $u_i \in \mathbb{F}_q^*$ such that $u_i \neq \pm u_j$ for $i < j \leqslant n \in \mathbb{N}$, the malicious users have to find an $\mathbb{F}_q$-solution of the polynomial system I, i.e., (1) with $c_i = u_i^2$. Here, n is a parameter, which is mostly selected by the attackers. Lemma 1 states that (the projective closure $\overline{I} \subset \mathbb{P}^{n+2}$ of) I is an absolutely irreducible curve of arithmetic genus $p_a = 1 + 2^{n-1}3n$. Recall that

the *Weil–Aubry–Perret bound* [10, Corollary 2.4] (valid due to irreducibility of $I/\overline{\mathbb{F}_q}$) is formulated as follows:

$$|\#\overline{I}(\mathbb{F}_q) - (q+1)| \leqslant 2p_a\sqrt{q}.$$

Meanwhile, the set at infinity $S_{z_0} := \overline{I} \setminus I = \overline{I} \cap \big\{z_0 = 0\big\}$ coincides with

$$\big\{x = z_0 = 0\big\} \cap \big\{y^2 + z_i^2 = 0\big\}_{i=1}^{n} \quad \subset \quad \mathbb{P}^{n+2}_{(x:y:z_0:z_1:\cdots:z_n)}.$$

Note that

$$S_{z_0} = \big\{(0:1:0:\pm\sqrt{-1}:\cdots:\pm\sqrt{-1})\big\},$$

where the signs $\pm$ are independently chosen for each coordinate. We live in the highly 2-adic realm, which means that $\nu > 1$ or, equivalently, $\sqrt{-1} \in \mathbb{F}_q$. Otherwise, $\sqrt{\cdot} \in \mathbb{F}_q$ is determined via one field exponentiation (instead of Müller's algorithm) and the (D)DoS attack is meaningless. Consequently, $\#S_{z_0}(\mathbb{F}_q) = \#S_{z_0} = 2^n$ and so

$$|\#I(\mathbb{F}_q) - (q + 1 - 2^n)| \leqslant 2p_a\sqrt{q}.$$

Besides, the degenerate set $S_y := I \cap \big\{y = 0\big\}$ is equal to

$$\big\{y = f(x) = 0\big\} \cap \big\{z_i = \pm u_i\big\}_{i=1}^{n} \quad \subset \quad \mathbb{A}^{n+2}_{(x,y,z_1,\cdots,z_n)}.$$

Generally speaking, f may have 0, 1, or 3 roots in $\mathbb{F}_q$. In this regard, we have to be content only with the inequalities $0 \leqslant \#S_y(\mathbb{F}_q) \leqslant 2^n 3$. The remaining hyperplane sections $S_{z_j} := I \cap \big\{z_j = 0\big\}$ (of course, $j > 0$) have the shape

$$\big\{c_j = y^2 = f(x)\big\} \cap \big\{c_i - c_j = z_i^2\big\}_{i=1}^{n} \quad \subset \quad \mathbb{A}^{n+2}_{(x,y,z_1,\cdots,z_n)}.$$

Analogously, $0 \leqslant \#S_{z_j}(\mathbb{F}_q) \leqslant 2^n 3$.

It is useful to understand that $S_y \cap S_{z_j} = S_{z_i} \cap S_{z_j} = \emptyset$ whenever $i \neq j$, although this circumstance does not play any role in the estimations $0 \leqslant \#S(\mathbb{F}_q) \leqslant (n+1)2^n 3$, where $S := S_y \cup S_{z_1} \cup \cdots \cup S_{z_n}$. We thus see that

$$q - 2p_a\sqrt{q} - (3n+4)2^n + 1 \;\leqslant\; \#(I \setminus S)(\mathbb{F}_q) \;\leqslant\; q + 2p_a\sqrt{q} - 2^n + 1.$$

Therefore, for q of cryptographic size and for the moderate n, there are a lot of $\mathbb{F}_q$-points in $I \setminus S$, i.e., possibilities of instantiating the (D)DoS attack. In particular, the honest user is not capable to store the giant list of all "poisoned" x-coordinates to thwart the attack by searching for the received ones in this list.

It is worth explaining why the set $S(\mathbb{F}_q)$ is ruled out of consideration. The $\mathbb{F}_q$-points of S_y are excluded, since it is immediately detected that an incoming x-coordinate corresponds to an order-2 point on E (inadmissible in the DLP context). Not to mention that the value $\sqrt{0} = 0$ is known a priori without launching Müller's algorithm or any other square-root one. By a similar reason, the adversaries should not send the points of the sets $S_{z_j}(\mathbb{F}_q)$. Otherwise, the receiver encounters the requiring root $u_j = \sqrt{f(x)}$ during the initialization of Müller's

algorithm. It is stopped earlier and so the (D)DoS attack is less productive (especially if j is much smaller than n).

The unique (but essential) obstacle for the attackers consists in difficulty of finding explicitly (the x-coordinate of) an $\mathbb{F}_q$-point in $I \setminus S$ for the quite large n. For instance, nothing prevents from iterating somehow $x_0 \in \mathbb{F}_q$ until the inverse image $pr_x^{-1}(x_0)$ of the projection $pr_x : I \to \mathbb{A}^1_x$ has a non-empty intersection with $(I \setminus S)(\mathbb{F}_q)$. This brute-force method is formalized in Algorithm 1. (Un)fortunately, it is obviously exponential in time as $n \to \infty$. The same strategy (cf. [65, Section 2]) via the projections to the other variables $y, z_1, \cdots, z_n$ does not help the situation. Curiously, the method is also non-deterministic, but it is not dramatic unlike the x-coordinate decompression. Indeed, the adversaries presumably dispose large resources and their work is not performed in real time. So, the symmetric (D)DoS attack on them is not relevant.

It is necessary to remember that the polynomial system I has a concrete type. Hence, nothing contradicts the existence of a clever algorithm of resolving it (in sub-exponential or even polynomial time) if the set $I \setminus S$ admits an $\mathbb{F}_q$-point at least theoretically. To put it another way, n must not exceed some limit, e.g., derived from the above lower bound on $\#(I \setminus S)(\mathbb{F}_q)$. Once such an algorithm is invented, the (D)DoS attack becomes critical, not just annoying. That is why it is paramount to establish the place of the problem at hand in the hierarchy of (hard) computational ones.

Algorithm 1: A brute-force method of finding the x-coordinate of an $\mathbb{F}_q$-point in $I \setminus S$

```
Data: n ∈ ℕ pairwise different values c_i ∈ F_q^*;
Result: The image pr_x(P) of an point P ∈ (I \ S)(F_q);
begin
    finish := false;
    while not finish do
        x := NextElement(F_q);
        f := f(x);
        c_0 := 2·f;
        finish := true;
        for i := 0 to n do
            Z := c_i − f;
            L := (Z/q);
            if L = 0 or L = −1 then
                finish := false;
                break;
            end
        end
    end
    return x.
end
```

3.2 Point (de)compression resistant to the new (D)DoS attack

In this section, the cardinality $\#E(\mathbb{F}_q)$ is assumed to be even. Let $P_0 \in E(\mathbb{F}_q)$ be any point of order $e > 1$ defined by the equality (2). Let's build the compression function

$$com\colon E(\mathbb{F}_q) \setminus \{\pm P_0, \mathcal{O}\} \to \mathbb{F}_q \times \mathbb{F}_2$$

founded on the quadratic $\mathbb{F}_q$-cover $\varphi\colon E \to \mathbb{A}^1_t$ associated with P_0 (see Section 2.2). Recall that one of the two components of φ is the birational $\mathbb{F}_q$-map $\psi : E \to C$ to the quartic curve $C\colon s^2 = h(t)$ with the equation (4). Denote through $M(u, v)$ Müller's algorithm of computing $s = \sqrt{u^2 - cv^2} \in \mathbb{F}_q^*$ given variables u, $v \in \mathbb{F}_q$ and a fixed $c \in \mathbb{F}_q^* \setminus (\mathbb{F}_q^*)^2$. Also, $sign$ will stand for any function $\mathbb{F}_q^* \to \mathbb{F}_2$ such that $sign(-s) \neq sign(s)$. As an illustration, for the prime q, this can be the parity function on the integer interval $[0, q-1]_{\mathbb{N}}$ realizing the field $\mathbb{F}_q$.

The new method of (de)compressing $E(\mathbb{F}_q)$ is specified in Algorithm 2 (resp., 3). The algorithms omit the points $\pm P_0$, $\mathcal{O}$, because the formulas (5) of ψ are unsuitable for them, although those (6) of ψ^{-1} do not have indefiniteness points. On the one hand, $\pm P_0$, $\mathcal{O}$ are out of use in practice, since the DLP is usually considered between prime-order group elements. On the other hand, the algorithms can be easily extended to the whole group $E(\mathbb{F}_q)$ by adding few supplementary bits to the output of com. In Algorithm 3 the $\mathbb{F}_q$-section $\sigma\colon \mathbb{A}^1_t \to S_h$ (from the end of Section 2.3) is leveraged to deterministically obtain an input (u, v) for M. It is nice that σ is correctly defined for each $t \in \mathbb{F}_q$. Lastly, since the roots of the polynomial h do not belong to $\mathbb{F}_q$, the element $s = M(u, v) = \sqrt{h(t)}$ never vanishes and so the function $sign$ does not need to have a value at 0.

Algorithm 2: New point compression

```
Data: A point P ∈ E(F_q) \ {±P_0, O};
Result: The pair com(P) ∈ F_q × F_2;
begin
    (t, s) := ψ(P);
    β := sign(s);
    return (t, β).
end
```

3.3 Alternative (de)compression for elliptic curves of j-invariant 0

Many elliptic curve cryptographers (if they are not conservatively-minded) like curves $E_b : y^2 = x^3 + b$ (for $b \in \mathbb{F}_q^*$) of j-invariant 0. The point is that these curves are the "richest" in the sense that they (and no others) admit a group automorphism of order 3. It has the form $[\omega] : (x, y) \mapsto (\omega x, y)$, where $\omega = (-1 + \sqrt{-3})/2$ is a primitive cubic root of unity. The given automorphism is useful for accelerating diverse cryptographic primitives on E_b. In particular,

Algorithm 3: New point decompression

Data: A pair $(t, \beta) \in \mathbb{F}_q \times \mathbb{F}_2$;
Result: The point $P \in E(\mathbb{F}_q) \setminus \{\pm P_0, \mathcal{O}\}$ such that $com(P) = (t, \beta)$;
begin
  $L := \left(\frac{h(t)}{q}\right)$;
  if $L = -1$ **then**
    return fail.
  end
  else
    $(u, v) := \sigma(t)$;
    $s := M(u, v)$;
    if $sign(s) \neq \beta$ **then**
      $s := -s$;
    end
    return $\psi^{-1}(t, s)$.
  end
end

these curves are paramount in *pairing-based cryptography* [23], since they enjoy (intimately thanks to $[\omega]$) twists of the highest degree 6 as opposed to other elliptic curves. Besides, $[\omega]$ underlies the work [37] (supplemented by [42, Section 4]), which constructs a hash function to $E_b(\mathbb{F}_q)$ friendly to highly 2-adic fields.

In our context, $[\omega]$ is a nice feature, because it gives rise to one more natural (de)compression method specific for $j = 0$ curves. Indeed, the projection $pr_y : E_b \to \mathbb{A}^1_y$ to the y-coordinate is nothing but the quotient $\mathbb{F}_q$-cover of degree 3 associated with the action of $[\omega]$ on E_b. Apart from y, two (rather than one) auxiliary bits are required (generally speaking) to restore the original cubic root $x = \sqrt[3]{y^2 - b}$ in the decompression stage. Note that ω (or, equivalently, $\sqrt{-3}$) does not necessarily lie in $\mathbb{F}_q$, that is, the case $3 \mid q-2$ is also allowed, despite the fact that E_b are supersingular curves for such $\mathbb{F}_q$. As is known, solely ordinary curves are attractive in today's ECC founded on the DLP. In fact, the additional two bits are superfluous when $\omega \in \mathbb{F}_q$, because pr_y becomes a bijective map on the level of $\mathbb{F}_q$-points.

Fortunately, highly 3-adic fields are not utilized (to the author's knowledge) in the real world in contrast to their 2-adic counterparts. Hence, $\sqrt[3]{\cdot} \in \mathbb{F}_q$ can be computed through one exponentiation in $\mathbb{F}_q$ at least if $27 \nmid q - 1$ (see, e.g., [38, Lemma 3]). As a result, for practical curves E_b, there is no difficulty in efficient (de)compression of their $\mathbb{F}_q$-points. Formally, the y-coordinate method continues to properly functionate even if the 3-adicity of $\mathbb{F}_q$ is large. However, there is no any sense to resort to its help in the given scenario, because general cubic-root extraction is substantially slower than square-root one (the latest prominent paper on the theme is perhaps [19]). By the way, the (de)compression method at hand is logically extended to the 2-dimensional setting in [36,38] and cannot be extended to the 3-dimensional one with $\sqrt[3]{\cdot} \in \mathbb{F}_q$ as the unique bottleneck.

3.4 Overview of point (de)compression methods

Table 1 contains all known point (de)compression methods without visible security flaws. First of all, in the column Bottleneck all the roots lie in $\mathbb{F}_q$. In turn, the column Complexity exhibits (except for PRG) the quantity of field multiplications being performed in the bottleneck of a (decompression) method. This comparison approach is workable, since few supplementary multiplications appear in other parts of each method. Besides, the operations $\left(\frac{\cdot}{q}\right)$ and $\cdot^{-1} \in \mathbb{F}_q$ are not taken into account, because they are pretty quick (see, e.g., [16] for the inversion operation) and, at the same time, they arise in all the methods a small number of times (as opposed to the multiplication operation). Finally, every method is undoubtedly economical as concerns used memory.

The abbreviation PRG denotes one execution of a cryptographic pseudo-random generator of an element in $\mathbb{F}_q^*$. Since such generators in practice are built on ad-hoc symmetric (rather than asymmetric) primitives, their running time is typically ignored in estimating the total performance. Therefore, PRG can be thrown out of the table unless an implementer applies a provably secure generator. Unlike the other methods, the third is non-deterministic, hence its indicated complexity is average. In other words, the generator sometimes has to be launched a little more than two times (or once).

As well as in the introduction, the table's g is a sub-quadratic function in the 2-adicity ν. At the moment, the TS-type algorithm with the smallest asymptotics is attributed to Sutherland [62]. More precisely,

$$g(\nu) = \Theta\left(\frac{\nu \log(\nu)}{\log(\log(\nu))}\right)$$

for his modification. As is customary, the best asymptotic behavior is not a guarantee that so is the true running time. Conversely, the latter is frequently much worse than expected. In this regard, it is an obligation of an implementer to select a concrete TS-type algorithm depending on the magnitude of ν. For the sake of conciseness, the table omits the table-lookup versions of TS-type algorithms.

To sum up, the new t-coordinate method seemingly outperforms other point (de)compression methods for very large 2-adicities if it is not about curves E_b for which exists the elegant y-coordinate method. Nonetheless, the author recognizes that for the moderate ν, it is sufficient to leverage the x-coordinate one with (the table-lookup variation of) an appropriate TS-type square-root algorithm. It is hard to derive a more or less exact bound on ν, because the answer depends on many implementation tricks of the mentioned methods. It is impossible to address all of them in the current article, not increasing cardinally its volume. In addition, implementers can use at their own risk the x-coordinate method with a strong pseudo-random generator if non-determinism does not bother them.

In real-life cryptographic applications one typically encounters sequences of elliptic curve points of considerable length $m \in \mathbb{N}$. Consequently, it is fairer to talk about multiple point (de)compression ($m > 1$) in place of single point one ($m = 1$). Surely, an arbitrary method of Table 1 can be exploited separately

Method	Restrictions	Bottleneck	Complexity
Classical with x	$\nu \leqslant 2$, i.e., $8 \nmid q-1$	$\sqrt{\cdot}$ via one exponentiation	$\ell \leqslant \cdot \leqslant 2\ell$
	No	$\sqrt{\cdot}$ via a TS-type algorithm	$\Theta(\ell + g(\nu))$
		$\sqrt{\cdot}$ via Müller's algorithm using a strong pseudo-random generator	$\approx 2\ell - \nu$ $+2 \cdot PRG$
Folklore with y	$j = 0$ and $27 \nmid q-1$	$\sqrt[3]{\cdot}$ via one exponentiation	$\ell \leqslant \cdot \leqslant 2\ell$
New with t	$2 \mid \#E(\mathbb{F}_q)$	$\sqrt{\cdot}$ via Müller's algorithm without the non-deterministic initialization	$\approx 2\ell - \nu$

Table 1. Safe methods of (de)compressing $E(\mathbb{F}_q)$ to (from) $\approx \ell = \lceil \log_2(q) \rceil$ bits

for each point, achieving (almost) ideal memory saving/bandwidth. But to be honest, it is worthwhile to cite the source [25] generalizing the idea of [34] from sequences with $m \in \{2, 3\}$ to longer ones. Remarkably, the given idea allows to compress to $m + 1$ elements of $\mathbb{F}_q$ (as usual, up to few bits), avoiding entirely radicals in $\mathbb{F}_q$, which is excellent news when living in the highly 2-adic realm. Nevertheless, the complexity of this approach apparently grows exponentially as $m \to \infty$ as noted in [38, Section 1.2]. To circumvent this trouble, one can successively employ it to short packets of points. However, this compression solution understandably becomes very far from optimal from the information theory point of view.

3.5 Examples of elliptic curves over highly 2-adic fields

Tables 2, 3, and 4 contain certain practical elliptic curves over fields of large 2-adicities. The first (resp., the second) table in this list is dedicated to twisted Edwards (resp., prime-order) curves of non-zero j-invariants. In turn, the third exhibits $j = 0$ curves independently of whether they have the twisted Edwards form or not.

Curve	Reference	ℓ	ν	e
Starkjub	[6]	252	192	8
Baby Jubjub	[13, Section 6]	254	28	
Jubjub	[24]	255	32	8
Bandersnatch	[45]			2

Table 2. Some real-world twisted Edwards curves of $j \neq 0$ over highly 2-adic fields

Curve	Reference	ℓ	ν	e
NIST P-224	[18, Section 3.2.1.2]	224	96	1
STARK curve	[5]	252	192	
MNT4-298	[14, Section 3.2]	298	17	
MNT6-298			34	
MNT4-753	[1]	753	15	
MNT6-753	[2]		30	

Table 3. Some real-world prime-order curves of $j \neq 0$ over highly 2-adic fields

Curve	Reference	ℓ	ν	e
Pallas-Vesta (Pasta)	[29]	254	32	1
Pluto-Eris	[30]	446		
BLS12-377	[64]	377	46	2^{92}

Table 4. Some real-world curves of $j = 0$ over highly 2-adic fields

The curve NIST P-224 stands out from the rest, because it was generated long time before the de facto use of zero-knowledge proofs. It continues (apart from the other curves of Table 3) to be vulnerable to the discovered (D)DoS attack, since the new (de)compression method is not applicable. Perhaps, the present article will additionally encourage NIST to exclude the given curve from the next version of its standard [18] as that happened at one time with the curve NIST P-192. The author believes that NIST P-224 is used much rarer (maybe, almost never in modern hard/software) than the curve NIST P-256 with the conventional 128-bit security level. So, there are no significant reasons to keep NIST P-224 in the standard. At the same time, the presence of this curve in the standard forces various entities to maintain its functionality. As a consequence, there remains a potential source of insecurity in view of the current paper.

Of course, the malicious users can try to apply the (D)DoS attack, sending x-coordinates of the slowest standardized curve NIST P-521. In this case, the honest user has to perform necessary computations over a larger field of length $\ell = 521$ instead of $\ell = 224$. It is paradoxical at first sight, but the given option apparently has less chances to be successful. Indeed, the attackers are able to send a proportionally shorter series of x-coordinates until the receiver blocks the incoming data from their side. The moral is the following: It is better for the dishonest users to feed "poisoned" x-coordinates for the faster curve NIST P-224 than "usual" x-coordinates for the slower one NIST P-521. In total, the receiving party will spend more time in the first case to process the data.

Speaking more generally, the problem of constructing an efficient deterministic (de)compression method for prime-order curves of $j \neq 0$ over highly 2-adic fields remains open. As is known, 2-*cycles* (a.k.a. *amicable pairs*) [59] of pairing-friendly curves constitute notable examples of such curves. Their advantage over SNARK-friendly curves in the twisted Edwards form is ability to deploy recursive SNARKs of unrestricted length. While the humanity does not yet know instances of pairing-friendly 2-cycles of acceptable performance (only pairs *MNT4-MNT6* [48] are now at the disposal), this is undoubtedly a domain of active research (see more details, e.g., in [12]). Incidentally, 2-cycles in which at least one of the curves is plain are easily realized with the help of $j = 0$ curves as confirmed by the pairs *Pallas-Vesta* (*Pasta*) and *Pluto-Eris* from Table 4.

In conclusion, it is useful to explicitly formulate the problem under consideration.

Problem 1. Let E be a prime-order elliptic curve of non-zero j-invariant over a highly 2-adic field $\mathbb{F}_q$ of length ℓ. Is there a (simple) deterministic (de)compression method for $E(\mathbb{F}_q)$ executing $O(\ell)$ multiplications in $\mathbb{F}_q$ and storing in memory $O(1)$ field elements (with little constants in the O-notation)? As earlier, it is about compression to $\approx \ell$ bits.

Acknowledgements. The author expresses his gratitude to Josep Maria Miret Biosca and Jordi Pujolàs Boix for great hospitality and fruitful mathematical discussions in Lleida. Also, it is impossible not to mention the aid of Laurent Moret-Bailly in establishing Theorem 1. Finally, the author is very grateful to Oleg Taraskin for many valuable talks about purely cryptographic questions of this paper.

References

1. MNT4-753, `https://coinlist.co/build/coda/pages/mnt4753`
2. MNT6-753, `https://coinlist.co/build/coda/pages/mnt6753`
3. Pseudorandom number generator, `https://en.wikipedia.org/wiki/pseudorandom_number_generator`
4. Random number generator attack, `https://en.wikipedia.org/wiki/random_number_generator_attack`
5. STARK curve, `https://docs.starkware.co/starkex/crypto/stark-curve.html`
6. Starkjub (2023), `https://github.com/hashcloak/starkjub`
7. Adams, W.W., Razar, M.J.: Multiples of points on elliptic curves and continued fractions. Proceedings of the London Mathematical Society **s3-41**(3), 481–498 (1980)
8. Aranha, D.F., El Housni, Y., Guillevic, A.: A survey of elliptic curves for proof systems. Designs, Codes and Cryptography **91**(11), 3333–3378 (2023)
9. Aranha, D.F., Salling Hvass, B., Spitters, B., Tibouchi, M.: Faster constant-time evaluation of the Kronecker symbol with application to elliptic curve hashing. In: CCS 2023: ACM SIGSAC Conference on Computer and Communications Security. pp. 3228–3238. Association for Computing Machinery, New York (2023)

10. Aubry, Y., Perret, M.: A Weil theorem for singular curves. In: Pellikaan, R., Perret, M., Vlăduţ, S.G. (eds.) Arithmetic, Geometry, and Coding Theory. pp. 1–7. Proceedings in Mathematics, De Gruyter, Berlin (1996)
11. 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
12. Bellés-Muñoz, M., Urroz, J.J., Silva, J.: Revisiting cycles of pairing-friendly elliptic curves. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. Lecture Notes in Computer Science, vol. 14082, pp. 3–37. Springer, Cham (2023)
13. Bellés-Muñoz, M., Whitehat, B., Baylina, J., Daza, V., Muñoz-Tapia, J.L.: Twisted Edwards elliptic curves for zero-knowledge circuits. Mathematics **9**(23), 3022 (2021)
14. 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)
15. Bernstein, D.J.: Faster square roots in annoying finite fields (2001), https://cr.yp.to/papers.html#sqroot
16. 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)
17. Billet, O., Joye, M.: The Jacobi model of an elliptic curve and side-channel analysis. In: Fossorier, M., Høholdt, T., Poli, A. (eds.) Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 2003. Lecture Notes in Computer Science, vol. 2643, pp. 34–42. Springer, Berlin, Heidelberg (2003)
18. Chen, L., Moody, D., Regenscheid, A., Robinson, A., Randall, K.: Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters (NIST Special Publication 800-186) (2023), https://csrc.nist.gov/publications/detail/sp/800-186/final
19. Cho, G.H., Kwon, S., Lee, H.S.: A refinement of Müller's cube root algorithm. Finite Fields and Their Applications **67**, 101708 (2020)
20. Châtelet, F.: Points rationnels sur certaines courbes et surfaces cubiques. L'Enseignement Mathématique **5**(3), 153–170 (1959)
21. Cipolla, M.: Un metodo per la risolutione della congruenza di secondo grado. Rendiconto dell'Accademia delle Scienze Fisiche e Matematiche **9**, 154–163 (1903)
22. Driver, E., Leonard, P.A., Williams, K.S.: Irreducible quartic polynomials with factorizations modulo p. The American Mathematical Monthly **112**(10), 876–890 (2005)
23. El Mrabet, N., Joye, M. (eds.): Guide to pairing-based cryptography. Cryptography and Network Security Series, Chapman and Hall/CRC, New York (2017)
24. Electric Coin Company: What is Jubjub?, https://bitzecbzc.github.io/technology/jubjub
25. Fan, X., Otemissov, A., Sica, F., Sidorenko, A.: Multiple point compression on elliptic curves. Designs, Codes and Cryptography **83**(3), 565–588 (2017)
26. Fouque, P.A., Lercier, R., Réal, D., Valette, F.: Fault attack on elliptic curve Montgomery ladder implementation. In: 2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography. pp. 92–98. Institute of Electrical and Electronics Engineers, New York (2008)

27. Galbraith, S.D.: Mathematics of public key cryptography. Cambridge University Press, New York (2012)
28. Hartshorne, R.: Algebraic geometry, Graduate Texts in Mathematics, vol. 52. Springer, New York, 8 edn. (1997)
29. Hopwood, D.: The Pasta curves for Halo 2 and beyond (2020), https://electriccoin.co/blog/the-pasta-curves-for-halo-2-and-beyond
30. Hopwood, D.: Pluto/Eris supporting evidence (2021), https://github.com/daira/pluto-eris
31. Icart, T.: How to hash into elliptic curves. In: Halevi, S. (ed.) Advances in Cryptology – CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 303–316. Springer, Berlin, Heidelberg (2009)
32. Internet Architecture Board, Handley, M.J., Rescorla, E.: Internet denial-of-service considerations (RFC 4732) (2006), https://datatracker.ietf.org/doc/rfc4732
33. Joye, M., Quisquater, J.J.: Efficient computation of full Lucas sequences. Electronics Letters **32**(6), 537–538 (1996)
34. 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)
35. Kiritchenko, V., Tsfasman, M., Vlăduţ, S., Zakharevich, I.: Quadratic residue patterns, algebraic curves and a K3 surface (2024), https://arxiv.org/abs/2403.16326
36. Koshelev, D.: New point compression method for elliptic $\mathbb{F}_{q^2}$-curves of j-invariant 0. Finite Fields and Their Applications **69**, 101774 (2021)
37. 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)
38. Koshelev, D.: Batch point compression in the context of advanced pairing-based protocols. Applicable Algebra in Engineering, Communication and Computing (2023), https://link.springer.com/article/10.1007/s00200-023-00625-3
39. Koshelev, D.: Generation of "independent" points on elliptic curves by means of Mordell–Weil lattices. Mathematical Cryptology **4**(1), 11–22 (2024)
40. Koshelev, D.: Hashing to elliptic curves through Cipolla–Lehmer–Müller's square root algorithm. Journal of Cryptology **37**(2), Article 11 (2024)
41. Koshelev, D.: Magma code (2024), https://github.com/dimitri-koshelev/point-de-compression-for-elliptic-curves-over-highly-2-adic-finite-fields
42. Koshelev, D.: Some remarks on how to hash faster onto elliptic curves. Journal of Computer Virology and Hacking Techniques **20**(2) (2024)
43. Lambert, R.J.: Method to calculate square roots for elliptic curve cryptography (2013), https://patents.google.com/patent/US9148282B2/en, United States patent No. 9148282B2
44. Lehmer, D.H.: Computer technology applied to the theory of numbers. In: LeVeque, W.J. (ed.) Studies in Number Theory. Studies in Mathematics, vol. 6, pp. 117–151. Mathematical Association of America, Washington (1969)
45. Masson, S., Sanso, A., Zhang, Z.: Bandersnatch: a fast elliptic curve built over the BLS12-381 scalar field (2021), https://eprint.iacr.org/2021/1152
46. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) Advances in Cryptology – CRYPTO 1985. Lecture Notes in Computer Science, vol. 218, pp. 417–426. Springer, Berlin, Heidelberg (1986)
47. Miret, J., Moreno, R., Rio, A., Valls, M.: Determining the 2-sylow subgroup of an elliptic curve over a finite field. Mathematics of Computation **74**(249), 411–427 (2005)

48. Miyaji, A., Nakabayashi, M., Takano, S.: New explicit conditions of elliptic curve traces for FR-reduction. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E84-A**(5), 1234–1243 (2001)
49. Müller, S.: On the computation of square roots in finite fields. Designs, Codes and Cryptography **31**(3), 301–312 (2004)
50. Petit, C., Kosters, M., Messeng, A.: Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) Public-Key Cryptography – PKC 2016. Lecture Notes in Computer Science, vol. 9615, pp. 3–18. Springer, Berlin, Heidelberg (2016)
51. Pornin, T.: Optimized discrete logarithm computation for faster square roots in finite fields (2023), `https://eprint.iacr.org/2023/828`
52. Pornin, T.: A prime-order group with complete formulas from even-order elliptic curves. IACR Communications in Cryptology **1**(1) (2024)
53. Sarkar, P.: Computing square roots faster than the Tonelli–Shanks/Bernstein algorithm. Advances in Mathematics of Communications **18**(1), 141–162 (2024)
54. Schicho, J.: The parameterization problem for algebraic surfaces. ACM SIGSAM Bulletin **33**(3) (1999)
55. Serre, J.P.: Algebraic groups and class fields, Graduate Texts in Mathematics, vol. 117. Springer, New York (1988)
56. Shanks, D.: Five number-theoretic algorithms. In: Thomas, R.S.D., Williams, H.C. (eds.) Proceedings of the Second Manitoba Conference on Numerical Mathematics. Congressus Numerantium, vol. 7, pp. 51–70. Utilitas Mathematica Publishing Inc., Winnipeg (1973)
57. Shoup, V.: A computational introduction to number theory and algebra. Cambridge University Press, Cambridge, 2 edn. (2008)
58. Shparlinski, I.E.: Pseudorandom number generators from elliptic curves. In: Luengo, I. (ed.) Recent Trends in Cryptography. Contemporary Mathematics, vol. 477, pp. 121–141. American Mathematical Society, Providence (2009)
59. Silverman, J.H., Stange, K.E.: Amicable pairs and aliquot cycles for elliptic curves. Experimental Mathematics **20**(3), 329–357 (2011)
60. Stichtenoth, H.: Algebraic function fields and codes, Graduate Texts in Mathematics, vol. 254. Springer, Berlin, Heidelberg, 2 edn. (2009)
61. Struik, R.: Alternative elliptic curve representations (2024), `https://datatracker.ietf.org/doc/draft-ietf-lwig-curve-representations/23`
62. Sutherland, A.V.: Structure computation and discrete logarithms in finite abelian p-groups. Mathematics of Computation **80**(273), 477–500 (2011)
63. Tonelli, A.: Bemerkung über die auflösung quadratischer congruenzen. Nachrichten von der Königlichen Gesellschaft der Wissenschaften und der Georg-Augusts-Universität zu Göttingen pp. 344–346 (1891)
64. Vlasov, A.: EIP-2539: BLS12-377 curve operations (2020), `https://eips.ethereum.org/EIPS/eip-2539`
65. Von zur Gathen, J., Shparlinski, I., Sinclair, A.: Finding points on curves over finite fields. SIAM Journal on Computing **32**(6), 1436–1448 (2003)