Some remarks on how to hash faster onto elliptic curves

Dmitrii Koshelev [1]

Parallel Computation Laboratory, École Normale Supérieure de Lyon, France

Abstract. This article proposes four optimizations of indifferentiable hashing onto (prime order subgroups of) ordinary elliptic curves over finite fields $\mathbb{F}_q$. One of them is dedicated to elliptic curves E provided that $q \equiv 2 \pmod 3$. The second deals with $q \equiv 2, 4 \pmod 7$ and an elliptic curve E_7 of j-invariant $-3^3 5^3$. The corresponding section plays a rather theoretical role, because (the quadratic twist of) E_7 is not used in real-world cryptography. The other two optimizations take place for the subgroups $\mathbb{G}_1$, $\mathbb{G}_2$ of some pairing-friendly curves. The performance gain comes from the smaller number of required exponentiations in $\mathbb{F}_q$ for hashing to $E(\mathbb{F}_q)$, $E_7(\mathbb{F}_q)$, and $\mathbb{G}_2$ as well as from the absence of necessity to hash directly onto $\mathbb{G}_1$. In particular, the new results affect the pairing-friendly curve BLS12-381 (the most popular in practice at the moment) and a few ones from the American standard NIST SP 800-186. Among other things, a taxonomy of state-of-the-art hash functions to elliptic curves will be presented. Finally, it will be discussed how to hash over highly 2-adic fields $\mathbb{F}_q$.

Key words: clearing cofactor, highly 2-adic fields, Icart-like encodings, hashing to elliptic curves, Klein quartic, optimal ate pairings.

1 How to hash onto pairing-friendly curves

There are a lot of articles (including recent ones) on how to hash into or onto elliptic curves over finite fields. So, with the reader's permission, a detailed introduction is not provided in order to avoid repetition. Good surveys are represented in [1, Section 8], [2]. It is worth emphasizing that throughout this text we will mean *hashing indifferentiable from a random oracle* (in the sense of [3, Section 2.2]).

Let E_1 be an ordinary pairing-friendly elliptic curve of embedding degree $k > 1$ over a finite field $\mathbb{F}_q$. Besides, let E_2 be a twist of E_1 of degree $d := \#\mathrm{Aut}(E_1)$ over the field $\mathbb{F}_{q^e}$, where $e := k/d \in \mathbb{N}$. As is customary, for a common prime divisor r of the orders $N_1 := \#E_1(\mathbb{F}_q)$ and $N_2 := \#E_2(\mathbb{F}_{q^e})$ denote by $\mathbb{G}_1 \subset E_1(\mathbb{F}_q)$ and $\mathbb{G}_2 \hookrightarrow E_2(\mathbb{F}_{q^e})$ the eigenspaces of the Frobenius endomorphism on $E_1[r] \subset E_1(\mathbb{F}_{q^k})$, associated with the eigenvalues 1, q, respectively. Note that the condition $e \in \mathbb{N}$ is not automatically met, i.e., this is our assumption. It is claimed (e.g., in [1, Theorem 3.3.5]) that for any prime divisor $r \mid N_1$ there is always a unique non-trivial $\mathbb{F}_{q^e}$-twist E_2 (of degree d) such that $r \mid N_2$. By abuse of notation, we will identify the order r subgroup $\mathbb{G}_2 \subset E_1(\mathbb{F}_{q^k})$ with its image under an $\mathbb{F}_{q^e}$-isomorphism $E_1 \to E_2$. Thus, $\mathbb{G}_1 = E_1(\mathbb{F}_q)[r]$ and $\mathbb{G}_2 = E_2(\mathbb{F}_{q^e})[r]$. Besides, $d \in \{2, 4, 6\}$ and $d = 2$ if and only if $j(E_i) \neq 0, 1728$ (resp., $d = 4$ iff $j(E_i) = 1728$, and $d = 6$ iff $j(E_i) = 0$).

This section explains how to hash onto $\mathbb{G}_2$ more efficiently and why we do not need to hash directly onto $\mathbb{G}_1$. In the first case, we will significantly exploit the presence of clearing

[1] Email: dimitri.koshelev@gmail.com

The author was supported by Ethereum Foundation

the cofactor $c_2 := N_2/r$. In the second one, on the contrary, clearing the cofactor $c_1 := N_1/r$ can be fully avoided. The fact is that *optimal ate pairings* $a\colon \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r \subset \mathbb{F}_{q^k}^*$ [1, Theorem 3.3.4] can be painlessly (unlike $E_2(\mathbb{F}_{q^e}) \times \mathbb{G}_1$) extended to $\mathbb{G}_2 \times E_1(\mathbb{F}_q)$, at least in main pairing-based protocols.

At the moment, due to [4, Table 1], the curve BLS12-381 is a de facto standard in pairing-based cryptography. More generally, the *Barreto–Lynn–Scott family* with $k = 12$ and $d = 6$ (see, e.g., [5, Section 3.1]) possesses the parameters

$$r(z) = z^4 - z^2 + 1, \qquad q(z) = (z-1)^2 r(z)/3 + z.$$

By definition, BLS12-381 is generated by $z := -0xd201000000010000$ and hence

$$\lceil \log_2(-z) \rceil = 64, \qquad \lceil \log_2(r) \rceil = 255, \qquad \lceil \log_2(q) \rceil = 381.$$

Notice that $r \ll q$ in contrast to the *Barreto–Naehrig family* [1, Example 4.2].

Recall that almost all known hash functions $\mathcal{H}_i\colon \{0,1\}^* \to \mathbb{G}_i$ are the compositions $\mathcal{H}_i = [c_i'] \circ h_i \circ \eta_i$. Here, $\eta_i\colon \{0,1\}^* \to S_i$ are hash functions to some finite sets, $h_1\colon S_1 \to E_1(\mathbb{F}_q)$ and $h_2\colon S_2 \to E_2(\mathbb{F}_{q^e})$ are just maps traditionally called *encodings*, and finally $c_i' \in \mathbb{N}$ such that $c_i \mid c_i'$, $r \nmid c_i'$. The scalar multiplication $[c_i']$ on the curve E_i is said to be *clearing cofactor*. Surprisingly, due to Fuentes-Castaneda et al. [6], it is more efficient to multiply points by scalars c_i' greater than c_i. The sets S_i are usually very simple, hence it is easy to combine η_i from existing hash functions $\{0,1\}^* \to \{0,1\}^\ell$ for $\ell \in \mathbb{N}$. The most complicated component of $\mathcal{H}_i$ is no doubt h_i, because its essence is based on high-dimensional algebraic geometry.

The majority of pairing-based protocols require a hash function to at most one group $\mathbb{G}_1$ or $\mathbb{G}_2$. Of course, any such protocol can be equivalently implemented for hashing to the other group. Without using point compression-decompression methods, elements of $\mathbb{G}_1$ (resp., $\mathbb{G}_2$) are obviously represented by $2\lceil \log_2(q) \rceil$ (resp., $2e\lceil \log_2(q) \rceil$) bits. Therefore, the choice often depends on whether a hash value should be more compact than the second pairing argument or vice versa. Besides, there are rare protocols, for example the Scott identity-based key agreement [7], where both hash functions $\mathcal{H}_i$ are necessary. Thus, the more cumbersome hashing to $\mathbb{G}_2$ cannot be replaced by hashing to $\mathbb{G}_1$ in all situations.

1.1 How not to hash onto $\mathbb{G}_1$

As far as the author knows, (non-degenerate) optimal ate pairings $a\colon \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r \subset \mathbb{F}_{q^k}^*$ are only used in today's real-world cryptography. The fact is that the corresponding Miller loop has the hypothetically smallest length $\approx \log_2(r)/\varphi(k)$, where φ is Euler's totient function. However, it is more practical to take the whole group $E_1(\mathbb{F}_q)$ instead of $\mathbb{G}_1$. In this case, the pairing $a\colon \mathbb{G}_2 \times E_1(\mathbb{F}_q) \to \mu_r$ becomes degenerate, but this is not important. A similar trick is done in [8, Section 5] for the Tate pairing in the context of isogeny-based cryptography, where, on the contrary, $\mathbb{G}_2$ is replaced by $E_1(\mathbb{F}_{q^k})$ in our notation.

Indeed, first, the length of the Miller loop depends only on the order of $\mathbb{G}_2$. Second, if for points $P \in E_1(\mathbb{F}_q)$ and $Q \in \mathbb{G}_2$ we have $a(Q,P) = 1$, then a fortiori $a(Q, c_1'P) = a(Q,P)^{c_1'} = 1$. Further, many popular protocols (such as the aggregated BLS signature [9]) work correctly whether the order of P equals r or not. It should be borne in mind that the *strong unforgeability property* (unlike the usual *existential* one) is not satisfied anymore for this signature as

emphasized in [9, Section 5.2]. Nevertheless, in the opinion of [10, Section 7], the existential unforgeability is sufficient in practice. Finally, the complexity of computing $a(Q, P)$ remains the same as that of computing $a(Q, c'_1 P)$, because P, $c'_1 P$ are equally defined over $\mathbb{F}_q$.

In [11] an encoding $h_1 \colon \mathbb{F}_q^2 \to E_1(\mathbb{F}_q)$ is constructed for elliptic curves $E_1 \colon y^2 = x^3 + b$ (of j-invariant 0) provided that $\sqrt{b} \in \mathbb{F}_q$. There, it is proved that h_1 is *admissible* in the sense of [3, Definition 4], which leads (in compliance with [3, Theorem 1]) to the indifferentiable hash function $h_1 \circ \eta_1$. Moreover, the only bottleneck of h_1 consists in extracting one cubic root in $\mathbb{F}_q$. For $q \not\equiv 1 \pmod{27}$ the latter can be implemented in constant time of raising to some power $n_1 \in \mathbb{N}$ in the field $\mathbb{F}_q$. In particular, the encoding is applicable to the curve BLS12-381 for which $b = 4$ and $n_1 = (q - 10)/27$.

Recall that famous *(indirect) Wahby–Boneh's encoding* h_{WB} [12, Section 4] (based on the *simplified SWU* one [3, Section 7]) is also valid for BLS12-381. It requires to extract one square root in $\mathbb{F}_q$, which for that curve is equivalent to raising in $\mathbb{F}_q$ to the power $n_2 := (q - 3)/4 \in \mathbb{N}$. The hash function H_2 from [12, Section 5] twice applies h_{WB} in order to act as a random oracle. By the way, the other indifferentiable hash function H_3 is even less performant than H_2 by virtue of [12, Figure 1].

To be exact, the Hamming weight $w(n_1) = 192$ and $w(n_2) = 228$. Denote by $\ell(n_i)$ the length of a shortest addition chain for n_i. In accordance with [13, Section 9.2.1], we establish the inequalities

$$382 \leq \ell(n_1) \lesssim 419, \qquad 385 \leq \ell(n_2) \lesssim 422.$$

We cannot claim that these upper bounds are mathematically correct, because $o(1)$ is omitted in contrast to the original inequality. However, in any case, the sought bounds are very close (probably equal) to ours.

On the other hand, following the *sliding window method* [13, Section 9.1.3] (with $k = 5$), the author explicitly derives in Magma [14] an addition chain for n_1 (resp., n_2) whose the length equals 449 (resp., 458). Curiously, a similar chain for n_2 of the same length 458, obtained by means of more advanced methods, appears in the optimized library *blst* [15]. Thus, the encoding h_{WB} applied twice is much slower than the one h_1 applied once. Indeed, $2 \cdot 458 - 449 = 467$ is a significant amount of multiplications in $\mathbb{F}_q$ that can be eliminated by giving priority to h_1.

The author provides in [16] a general reference implementation of h_1 in Sage. The corresponding Rust implementation and benchmarks for BLS12-381 (and BLS12-377 [17]) are given in [18] by Zhang. He uses the famous library *arkworks* as a base. His low-level comparison of running time shows that the new encoding is actually more efficient than h_{WB}, not to mention universal *Shallue–van de Woestijne's (SW) encoding* [19].

1.2 How to hash onto $\mathbb{G}_2$

To the author's knowledge, optimal ate pairings do not have a natural extension to $E_2(\mathbb{F}_{q^e}) \times \mathbb{G}_1$. Conversely, (non-degenerate) *twisted optimal ate pairings* [1, Theorem 3.3.8] of the form $\mathbb{G}_1 \times \mathbb{G}_2 \to \mu_r$ are readily extended to $\mathbb{G}_1 \times E_2(\mathbb{F}_{q^e})$. But for them the Miller loop is of a larger length than for (usual) optimal ate pairings. It is widely recognized that a pairing is a more laborious operation than an elliptic curve scalar multiplication. Therefore, reducing the Miller loop seems a better solution than avoiding the multiplication by c'_2.

For the sake of convenience, introduce so-called *tensor multiplication* of any two maps $h\colon S \to G$, $g\colon T \to G$ from sets S, T to the same group $(G, +)$:

$$h \otimes g\colon S \times T \to G \qquad (s,t) \mapsto h(s) + g(t).$$

We know (e.g., from [1, Theorem 2.11]) that $E_2(\mathbb{F}_{q^e}) \simeq \mathbb{Z}/(mr) \times \mathbb{Z}/\ell$, where $\ell \mid m$ and $m\ell = c_2$. Pick any independent points $P_0, P_1 \in E_2(\mathbb{F}_{q^e})$ of orders m and ℓ, respectively. The independency means that $P_1 \in E_2(\mathbb{F}_{q^e}) \setminus \langle P_0 \rangle$ if $\ell > 1$, and $P_1 = (0:1:0)$ if $\ell = 1$. Consider the set $V := [0, m) \times [0, \ell)$ and the maps

$$g\colon V \to E_2(\mathbb{F}_{q^e}) \qquad (v_0, v_1) \mapsto v_0 P_0 + v_1 P_1,$$
$$F\colon \mathbb{F}_{q^e} \times V \to \mathbb{G}_2 \qquad F := [c_2'] \circ (h_2 \otimes g).$$

For the next theorem we need the notions of (B-)*well-distributed encoding* [20, Definitions 5] and (ϵ-)*regular map* [20, Definition 3] (with respect to the uniform distribution on its domain).

Theorem 1. *Assume that $h_2\colon \mathbb{F}_{q^e} \to E_2(\mathbb{F}_{q^e})$ is a B-well-distributed encoding (for $B \in \mathbb{R}_{>0}$). Then, the map F is ϵ-regular, where $\epsilon := B\sqrt{r/q^e}$. As a result, ϵ is negligible whenever $e > 1$.*

This is an immediate consequence of [20, Corollary 1] and [3, Lemma 13].

Note that F is a *samplable map* (in the sense of [3, Definition 4]) if, as is often the case, h_2 enjoys a large image, that is, $\#\mathrm{Im}(h_2) = \Theta(q^e)$. Indeed, this property follows from [3, Lemma 13] and [20, Algorithm 1]. Eventually, we establish a series of corollaries.

Corollary 1. *The map F is admissible.*

Corollary 2. *If a hash function $\eta\colon \{0,1\}^* \to \mathbb{F}_{q^e}$ is indifferentiable from a random oracle, then so is the hash function $[c_2'] \circ h_2 \circ \eta\colon \{0,1\}^* \to \mathbb{G}_2$ (denoted by H_4 in [12, Section 5]).*

Proof. Take another random oracle $\theta\colon \{0,1\}^* \to V$. Therefore, the functions $(\eta, \theta)(s) := \big(\eta(s), \theta(s)\big)$ and hence $F \circ (\eta, \theta)\colon \{0,1\}^* \to \mathbb{G}_2$ also act as a random oracle (the second fact is [3, Theorem 1]). Finally, obviously, $\mathrm{H}_4 = F \circ (\eta, \theta)$. □

For the BLS12-381 curve $E_2\colon y^2 = x^3 + 4(1+i)$ (where $i := \sqrt{-1} \notin \mathbb{F}_q$) in the role of h_2 the article [12, Section 5] proposes Wahby–Boneh's encoding. However, that article does not notice the indifferentiability of H_4. By the way, the other (indifferentiable) hash functions H_5, H_6 are even slower than H_4 by virtue of [12, Figure 1].

2 Batching two "relatively prime" encodings

Hash functions to classical (i.e., non-pairing-friendly) elliptic curves have become more and more in demand. Indeed, according to [21, Table I], they are actively used in many *PAKE (Password-Authenticated Key Exchange) protocols*. Incidentally, several years ago CFRG (Crypto Forum Research Group) conducted the PAKE selection process [22] in which the protocols CPace [23] and OPAQUE [24] won. Besides, such hash functions are necessary for some *blind signatures* (e.g., from [25, Section 3.3]), which serve as a basis of modern electronic voting systems. It is also worth mentioning that hashing to elliptic curves is applied in *OPRFs (Oblivious Pseudorandom Functions)* [26], among others, in the 2HashDH scheme [27, Section 3.1], [28, Section 3].

2.1 How to hash onto $E(\mathbb{F}_q)$ if $q \equiv 2 \pmod 3$

Consider an elliptic curve $E\colon y^2 = x^3 + ax + b$ defined over a finite field $\mathbb{F}_q$. Under the condition $q \equiv 2 \pmod 3$ (resp., $j(E) \neq 0, 1728$), *Icart's encoding* h_I [29] (resp., the simplified SWU one h_{sSWU}) is available. In accordance with [29, Lemma 4], [3, Lemma 6], for any $P \in E(\mathbb{F}_q)$ we have $\#h_I^{-1}(P) \leqslant 4$ and $\#h_{sSWU}^{-1}(P) \leqslant 8$. In fact, if an implementation of h_{sSWU} takes into account the sign of the y-coordinate, then $\#h_{sSWU}^{-1}(P) \leqslant 4$. At the same time, by virtue of [30, Section 5], the encoding h_I (resp., h_{sSWU}) is B-well-distributed with $B = 13$ (resp., $B = 53$) at least for q of a cryptographic size. Applying [20, Corollary 1], we thus get the next statement.

Lemma 1. *Suppose that $q \equiv 2 \pmod 3$ and $j(E) \neq 0, 1728$. Then, the map $F := h_I \otimes h_{sSWU}\colon \mathbb{F}_q^2 \to E(\mathbb{F}_q)$ is ϵ-regular for the negligible value $\epsilon := 26\sqrt{N}/q$, where $N := \#E(\mathbb{F}_q)$.*

From now on we assume in addition that $q \equiv 3 \pmod 4$. Obviously,

$$q \equiv 2 \pmod 3,\ q \equiv 3 \pmod 4 \qquad \Leftrightarrow \qquad q \equiv 11 \pmod{12}.$$

For the sake of compactness, we put $e := (q+1)/4$ and $k := (q+1)/12$. Given $Z = n/d$ such that $n, d \in \mathbb{F}_q^*$, we obtain:

$$z := Z^k = n^k \cdot d^{q-1-k} = n^k \cdot d^{(11q-13)/12} = nd^9 \cdot (nd^{11})^{(q-11)/12}, \qquad z^6 = Z^{(q+1)/2} = \Big(\frac{Z}{q}\Big) Z,$$

where $\big(\frac{Z}{q}\big)$ is the Legendre symbol. In particular, $z = \sqrt[6]{Z}$ whenever Z is a quadratic residue in $\mathbb{F}_q$.

Given $(t, s) \in \mathbb{F}_q^2$, we need to evaluate $h_I(t)$ and $h_{sSWU}(s)$. As is known, separately each of these points can be computed in constant time of one exponentiation in $\mathbb{F}_q$ (see the case of h_{sSWU} in [12, Section 4.2]). Let's show that this is also possible simultaneously for the two points (and hence for $F(t, s)$). The only cumbersome part of h_I (resp., h_{sSWU}) consists in the exponentiation $\sqrt[3]{f} = f^{(2q-1)/3}$ (resp., $\pm g^e$ such that $(g^e)^2 = \big(\frac{g}{q}\big) g$), where

$$f := \left(\frac{3a - t^4}{6t}\right)^2 - b - \frac{t^6}{27}, \qquad g := -\frac{b}{a}\left(1 + \frac{1}{s^4 - s^2}\right).$$

Evidently, $\sqrt[3]{f}$ is the unique cubic root of f in $\mathbb{F}_q$ and for our purpose it is sufficient to find g^e up to a sign. For the sake of simplicity, we exclude from consideration the zeros and poles of the functions f, g. As usual, they can be processed individually.

It is suggested to act in a similar way as in [31, Section 3], that is, for $Z := f^2 g^3$ to compute $z = Z^k$ (almost $\sqrt[6]{Z}$) instead of computing separately $\sqrt[3]{f}$ and $\pm g^e$ (almost $\sqrt{g}$). Note that

$$z = f^{(q+1)/6} \cdot g^e = \Big(\frac{f}{q}\Big) \sqrt[3]{f} \cdot g^e, \qquad z^2 = \sqrt[3]{f^2} \cdot \Big(\frac{g}{q}\Big) g.$$

Introducing the auxiliary notation $\theta := fg/z^2$, we get the equalities

$$\sqrt[3]{f} = \frac{\big(\frac{g}{q}\big) fg}{z^2} = \Big(\frac{g}{q}\Big)\theta, \qquad g^e = \frac{z}{\big(\frac{f}{q}\big)\sqrt[3]{f}} = \frac{z}{\big(\frac{fg}{q}\big)\theta}.$$

We see that $\theta^3 = \left(\frac{g}{q}\right) f$ and $z^6 = \left(\frac{g}{q}\right) Z$. Therefore, the symbol $\left(\frac{g}{q}\right)$ can be determined for free. More formally,

$$\left(\sqrt[3]{f},\, \pm g^e\right) = \begin{cases} \left(\theta,\, z/\theta\right) & \text{if} \quad \theta^3 = f,\ \text{i.e.,}\ z^6 = Z, \\ \left(-\theta,\, z/\theta\right) & \text{otherwise.} \end{cases}$$

Bearing in mind the formula above for $(n/d)^k$ without the inversion operation, we completely justify the next remark.

Remark 1. *The map F (in contrast to $h_I^{\otimes 2}$ and $h_{sSWU}^{\otimes 2}$) can be computed in constant time of one exponentiation in $\mathbb{F}_q$.*

By analogy with [14], given q, it is of course not difficult to derive explicit short addition chains for raising to the power k. Besides, F is a samplable map due to [20, Algorithm 1], which eventually leads to the following result.

Corollary 3. *The map $F\colon \mathbb{F}_q^2 \to E(\mathbb{F}_q)$ is admissible.*

Remark 1 is still valid when h_{sSWU} is replaced by any encoding implementable with the cost of extracting one square root in $\mathbb{F}_q$. We chose h_{sSWU}, because it is the most universal among such encodings known in the literature. In particular, this encoding is relevant even if N is a prime (that is, the cofactor equals 1), which is the case for many classical elliptic curves. Note that for $q \equiv 11 \pmod{12}$ curves of j-invariants 0, 1728 are supersingular in compliance with [13, Section 24.2.1.c]. Since such curves pose special challenges for security by virtue of [1, Remark 2.22], the map h_{sSWU} does not have restrictions in the current context.

There are a lot of standardized elliptic curves over fields $\mathbb{F}_q$ such that $q \equiv 11 \pmod{12}$. It is readily checked that this condition is fulfilled, e.g., for the (unique) French curve FRP256v1 [32], for the curves P-192, P-384, and Curve448-Goldilocks from NIST SP 800-186 [33, Section 4.2.1] as well as for all Russian curves [34, Appendix B] except for id-GostR3410-2001-CryptoPro-B-ParamSet. Remark 1 remains true in the case $q \equiv 2 \pmod 3$, $q \equiv 5 \pmod 8$ when a square root is still expressed via one exponentiation (see, e.g., [2, Appendix I.2]). However, the author did not find standardized curves over such fields, hence he decided to stop in order not to complicate the text.

2.2 Generalization of Icart's encoding

The given section answers the following curious question.

Question 1. *Do we know an example of a superelliptic $\mathbb{F}_q$-curve $C\colon y^m = f(x)$ and an $\mathbb{F}_q$-cover $\chi\colon C \to E$ (of small degree) onto some ordinary elliptic $\mathbb{F}_q$-curve E provided that $m \in \mathbb{N}$ is relatively prime with $3(q-1)$?*

The case $m = 2$ is obviously impossible for odd q we deal with in this article. In turn, the case $m = 3$ is deliberately excluded, because it is treated by Icart to the full extent. If the question has a positive answer, then we have yet another encoding $h_\chi := \chi \circ pr_x^{-1}\colon \mathbb{F}_q \to E(\mathbb{F}_q)$, where pr_x is the projection from C to the x-coordinate. Indeed, since $\mathrm{GCD}(m, q-1) = 1$, for every $x \in \mathbb{F}_q$ there is a unique $\mathbb{F}_q$-root $y = \sqrt[m]{f(x)} = f(x)^{m^{-1} \pmod{q-1}}$. Of course, from

the practical point of view, h_χ is only useful if Icart's encoding is not applicable, that is, $q \equiv 1 \pmod 3$. However, in this section the latter condition is unnecessary.

By analogy with Section 2.1, we also possess the map $F_\chi := h_\chi \otimes h_{sSWU} \colon \mathbb{F}_q^2 \to E(\mathbb{F}_q)$. Note that $\#h_\chi^{-1}(P) \leqslant \deg(\chi)$ for each point $P \in E(\mathbb{F}_q)$. This property is sufficient for F_χ to be regular (and hence admissible), because h_{sSWU} is already well-distributed. Moreover, since m is odd, we can extract the roots $\sqrt[m]{f}$ and $\sqrt{g}$ at the cost of extracting the root $z := \sqrt[2m]{Z}$, where $Z := f^2 g^m$. Indeed, $z = \sqrt[m]{f}\sqrt{g}$, hence $\sqrt[m]{f} = fg^{(m-1)/2}/z^{m-1}$ and $\sqrt{g} = z/\sqrt[m]{f}$. Furthermore, z is expressed via one exponentiation in $\mathbb{F}_q$ whenever this is true for $\sqrt{g}$, e.g., in the case $q \equiv 3 \pmod 4$, because $z = \sqrt{\sqrt[m]{Z}}$. As well as for $m = 3$, the reasoning is readily extended, taking into account the sign $\left(\frac{Z}{q}\right) = \left(\frac{g}{q}\right)$ in the equality $z^{2m} = \left(\frac{g}{q}\right)Z$.

Denote by $\overline{C} \subset \mathbb{P}^2$ and $\overline{pr_x} \colon \overline{C} \to \mathbb{P}^1$ the projective closure of C and pr_x, respectively. It is quite evident that $\overline{pr_x}$ (resp., $\overline{C}$) fits the definition of an *exceptional cover* (resp., *median value curve*) in the sense of [35] no matter $m \geqslant \deg(f)$ or not. By definition, $\overline{pr_x} \colon \overline{C}(\mathbb{F}_q) \to \mathbb{P}^1(\mathbb{F}_q)$ is a bijection and $\#\overline{C}(\mathbb{F}_q) = q + 1$. As a result, h_χ is an *Icart-like encoding* in the terminology of [36]. So h_χ is not "almost surjective", that is, $\#h_\chi(\mathbb{F}_q) \neq q + o(q)$. Therefore, this encoding itself cannot be admissible. Nevertheless, it is not required due to the availability of F_χ. Incidentally, another type of Icart-like encodings is studied in [37] for elliptic curves of j-invariants 0, 1728 whose Frobenius trace has a small divisor.

Interestingly, the author found only one desired example (with $m = 7$) in the existing literature on elliptic curves. Consider the *Klein quartic* $K := X^3Y + Y^3Z + Z^3X$ on the projective plane $\mathbb{P}^2_{(X:Y:Z)}$. It is clearly a non-singular curve of genus 3. The detailed information about the Klein quartic is represented in the book [38] and especially in its chapter "The Klein quartic in number theory". In particular, there is a birational isomorphism between K and $C_7 \colon y^7 = x^2(x+1)$ of the form

$$\tau \colon K \dashrightarrow C_7 \qquad (X : Y : Z) \mapsto \left(\frac{Y^3}{Z^2X}, -\frac{Y}{Z}\right),$$

$$\tau^{-1} \colon C_7 \dashrightarrow K \qquad (x, y) \mapsto \left(-\frac{y^3}{x} : -y : 1\right).$$

Furthermore, we have a cover

$$\chi \colon C_7 \to E_7 \qquad (x, y) \mapsto \left(\frac{\mathrm{num}_x}{\mathrm{den}}, \frac{\mathrm{num}_y}{\mathrm{den}}\right)$$

onto the elliptic curve $E_7 \colon y^2 = x^3 + (21/4)x^2 + 7x$, where

$$\mathrm{num}_x := 2y\big(x^2(-y^7 - y^5 - y^2 + y + 1) + x(-y^2 + y + 1)y^7 + (y^5 + y^2 - y - 1)y^7\big),$$

$$\mathrm{num}_y := x^2(3y^8 + 2y^7 + 3y^6 + 2y^5 + 2y^4 - y^3 - 3y^2 - 3y - 2) + x(2y^4 - y^3 - 3y^2 - 3y - 2)y^7$$

$$+(-3y^6 - 2y^5 - 2y^4 + y^3 + 3y^2 + 3y + 2)y^7, \qquad \mathrm{den} := 2y^3(x^2 + xy^7 - y^7).$$

The correctness of these formulas is checked in Magma [14]. Similar formulas (with respect to another model of E_7) are given in [39]. The cover χ is nothing but the composition of τ^{-1} and the canonical map $K \to K/\phi \simeq E_7$, where ϕ is the cyclic shift $(X : Y : Z) \mapsto (Y : Z : X)$ on K. Inter alia, $\deg(\chi) = 3$. Finally, it does not seem that χ can be easily modified to cover over $\mathbb{F}_q$ the quadratic twist of $E_7/\mathbb{F}_q$.

Put ζ to be a fixed primitive 7-th root of unity and $\alpha := (-1+\sqrt{-7})/2 = \zeta^4+\zeta^2+\zeta$. It is known that $j(E_7) = -3375 = -3^3 5^3$ and the natural lift $E_7/\mathbb{Q}$ has complex multiplication by $\mathbb{Z}[\alpha]$ in the sense of [13, Section 18.1.1]. Therefore, the curve $E_7/\mathbb{F}_q$ is ordinary if and only if $\left(\frac{-7}{p}\right) = 1$ (see, e.g., [1, Section 4.3]), where $p > 7$ denotes the characteristic of the field $\mathbb{F}_q$. According to the quadratic reciprocity law [13, Section 2.3.4.a], this is equivalent to the condition $\left(\frac{p}{7}\right) = 1$, that is, $p \equiv 1, 2, 4 \pmod 7$. We need to exclude the case $p \equiv 1 \pmod 7$ with regard to Question 1. In other words, $\sqrt{-7} \in \mathbb{F}_p$, while $\zeta \notin \mathbb{F}_p$. More precisely, $\mathbb{F}_p(\zeta) = \mathbb{F}_{p^3}$, because the extension degree $[\mathbb{Q}(\zeta) : \mathbb{Q}] = \varphi(7) = 6$. Thus, we are interested in prime powers $q \equiv 2, 4 \pmod 7$.

For $\ell := 7^{-1} \pmod{q-1}$ and $Z = n/d$ such that $n, d \in \mathbb{F}_q^*$ we obtain:

The case $q \equiv 2 \pmod 7$**:**

$$\ell = \frac{6q-5}{7} \in \mathbb{N}, \qquad \sqrt[7]{Z} = Z^\ell = n^\ell \cdot d^{q-1-\ell} = n^\ell \cdot d^{(q-2)/7} = n \cdot (n^6 d)^{(q-2)/7};$$

The case $q \equiv 4 \pmod 7$**:**

$$\ell = \frac{2q-1}{7} \in \mathbb{N}, \qquad \sqrt[7]{Z} = Z^\ell = n^\ell \cdot d^{q-1-\ell} = n^\ell \cdot d^{(5q-6)/7} = nd^2 \cdot (n^2 d^5)^{(q-4)/7}.$$

Besides, for $e := (q+1)/4 \in \mathbb{N}$ and $k := \ell e \pmod{q-1}$, and $L := \left(\frac{Z}{q}\right)$ we have:

The case $q \equiv 2 \pmod 7$, $q \equiv 3 \pmod 4$ **or, equivalently,** $q \equiv 23 \pmod{28}$**:**

$$k = \frac{5q-3}{28} \in \mathbb{N}, \qquad \sqrt[14]{LZ} = Z^k = n^k \cdot d^{q-1-k} = n^k \cdot d^{(23q-25)/28} = n^4 d^{18} \cdot (n^5 d^{23})^{(q-23)/28};$$

The case $q \equiv 4 \pmod 7$, $q \equiv 3 \pmod 4$ **or, equivalently,** $q \equiv 11 \pmod{28}$**:**

$$k = \frac{11q-9}{28} \in \mathbb{N}, \qquad \sqrt[14]{LZ} = Z^k = n^k \cdot d^{q-1-k} = n^k \cdot d^{(17q-19)/28} = n^4 d^6 \cdot (n^{11} d^{17})^{(q-11)/28}.$$

3 Taxonomy of hash functions to elliptic curves

This section aims to systematize known results on hashing to elliptic $\mathbb{F}_q$-curves E. Table 1 contains state-of-the-art admissible encodings of the form $\mathbb{F}_q^2 \to E(\mathbb{F}_q)$. For the sake of completeness, Table 2 exhibiting encodings $\mathbb{F}_q \to E(\mathbb{F}_q)$ is also included. The latter are not regular, because their full images are only proportions of the whole group $E(\mathbb{F}_q)$. Nevertheless, in a series of situations they are more efficient than the former. The point is that some protocols remain secure whether a used hash function $\{0,1\}^* \to E(\mathbb{F}_q)$ acts as a random oracle or not. In the literature there are a lot of other encodings to elliptic curves. The tables demonstrate only those, which arose earlier and which are at the same time the best at least for certain E and $\mathbb{F}_q$.

The only exception is *Skałba's encoding* [40]. Indeed, in contrast to Shallue–van de Woestijne's encoding, it does not cover curves of j-invariant 0, not to mention that Skałba's formulas are quite awkward. So it is widely recognized that the former is worse than the latter.

Year	Authors	References	Complexity	Conditions
2005	Skałba	[40]	$\sqrt{\cdot}$ and two $\left(\frac{\cdot}{q}\right)$	$a \neq 0$
2006	Shallue, van de Woestijne (modification)	[19]		
2022	Chávez-Saab, Rodriguez-Henriquez, Tibouchi (SwiftEC)	[41]		[41, Theorem 3]
2009-2010	Icart (combination with the simplified SWU map)	[3, Section 7], [29], Section 2.1	$\sqrt[6]{\cdot}$	$q \equiv 2 \pmod 3$, $ab \neq 0$
2022	K.	[11]	$\sqrt[3]{\cdot}$	$a = 0$, $\sqrt{b} \in \mathbb{F}_q$
		[42]	$\sqrt[4]{\cdot}$	$b = 0$
2023	K. (combination with the simplified SWU map)	[3, Section 7], Section 2.2	$\sqrt[14]{\cdot}$	$q \equiv 2, 4 \pmod 7$, j-invariant $-3^3 5^3$

Table 1: Taxonomy of admissible encodings $\mathbb{F}_q^2 \to E(\mathbb{F}_q)$ to elliptic $\mathbb{F}_q$-curves $E\colon y^2 = x^3 + ax + b$

Year	Authors	Reference	Complexity	Conditions
2009	Icart	[29]	$\sqrt[3]{\cdot}$	$q \equiv 2 \pmod 3$
2010	Brier et al. (the simplified SWU map)	[3, Section 7]	$\sqrt{\cdot}$	$ab \neq 0$
2019	Wahby, Boneh	[12]		$ab = 0$, E has a vertical $\mathbb{F}_q$-isogeny of small degree
2022	K.	[37]		$ab = 0$, the trace of E has a small divisor
2023		Section 2.2	$\sqrt[7]{\cdot}$	$q \equiv 2, 4 \pmod 7$, j-invariant $-3^3 5^3$

Table 2: Taxonomy of (non-admissible) encodings $\mathbb{F}_q \to E(\mathbb{F}_q)$ to elliptic $\mathbb{F}_q$-curves $E\colon y^2 = x^3 + ax + b$

Nonetheless, seminal Skałba's work merits to be cited, because it is historically the first in this research area. Further, both encodings need to determine the values of two Legendre symbols. However, we have to bear in mind the recent breakthrhough [43, 44] in fast constant-time implementations of the Legendre symbol. In other words, the computational complexity of the encodings is close to that of one square root extraction.

It is necessary to explain why Skałba's encoding is considered to be admissible and what is meant by modification of the SW one, which appears to be equally admissible. The original encodings (of the form $\mathbb{F}_q \to E(\mathbb{F}_q)$) are of course not so, because they are far from surjective. First of all, recall that the threefold

$$T\colon y^2 = f(x_1)f(x_2)f(x_3) \quad \subset \quad \mathbb{A}^4_{(x_1,x_2,x_3,y)},$$

where $E\colon y^2 = f(x) := x^3 + ax + b$, is at the core of the encodings under discussion. To be precise, we have the auxiliary map

$$h'\colon T(\mathbb{F}_q) \to E(\mathbb{F}_q) \qquad h'(x_1,x_2,x_3,y) := \begin{cases} \big(x_1, \sqrt{f(x_1)}\big) & \text{if} \quad \big(\frac{f(x_1)}{q}\big) \in \{0,1\}, \\ \big(x_2, \sqrt{f(x_2)}\big) & \text{if} \quad \big(\frac{f(x_2)}{q}\big) \in \{0,1\}, \\ \big(x_3, \sqrt{f(x_3)}\big) & \text{otherwise, i.e., } \big(\frac{f(x_3)}{q}\big) \in \{0,1\}. \end{cases}$$

Skałba's encoding is based on $\mathbb{F}_q$-*unirationality* of the *Châtelet surface* (see, e.g., [45, Sections 1-2]). More concretely, one deals with the surface

$$S\colon y_1^2 + 12ay_2^2 = f(x) \quad \subset \quad \mathbb{A}^3_{(x,y_1,y_2)}. \qquad \text{[40, Equation (3)]}$$

By definition, there is a *dominant* $\mathbb{F}_q$-*map* $\psi\colon \mathbb{A}^2_{(t_1,t_2)} \dashrightarrow S$ in the sense of [13, Definition 4.43]. Such a map is given in [40, Lemma 3] and yet another rational $\mathbb{F}_q$-map $\varphi\colon S \dashrightarrow T$ is from [40, Lemma 2]. Besides, introduce the following notation:

$$\varphi \circ \psi = (X_1, X_2, X_3, Y)\colon \quad \mathbb{A}^2_{(t_1,t_2)} \dashrightarrow T$$

with irreducible fractions

$$X_i = \frac{\mathrm{num}_i}{\mathrm{den}_i}, \qquad \mathrm{num}_i, \mathrm{den}_i \in \mathbb{F}_q[t_1,t_2], \qquad \text{and} \qquad Y \in \mathbb{F}_q(t_1,t_2).$$

Finally, for $\alpha \in \mathbb{F}_q$ let's define the curves

$$C_{i,\alpha} := \alpha\cdot\mathrm{den}_i - \mathrm{num}_i, \qquad C_{i,\infty} := \mathrm{den}_i \qquad \subset \qquad \mathbb{A}^2_{(t_1,t_2)}.$$

A "right" version of Skałba's encoding is given as follows:

$$h\colon \mathbb{F}_q^2 \to E(\mathbb{F}_q) \qquad h(t_1,t_2) := \begin{cases} (0:1:0) & \text{if} \quad \exists i\colon (t_1,t_2) \in C_{i,\infty}, \\ \big(h' \circ \varphi \circ \psi\big)(t_1,t_2) & \text{otherwise.} \end{cases}$$

In order to shorten a bit his formulas Skałba restricts h to the line $t_1 = t_2$, because he does not worry about the regularity property. Using the intuition confirmed by [11, Theorem 3], [42, Theorem 1], the author suggests that the curves $C_{i,\alpha}$ are probably absolutely irreducible

except for few α. He does not possess sufficient computer resources to prove this statement, since Skalba's formulas are fairly cumbersome. If the assumption is true, then, by analogy with [11, Corollary 2], [42, Corollary 2], it follows that the encoding h is regular. As usual, h is also efficiently computable and samplable in a clear way, which implies its admissibility.

The SW encoding is obtained in almost the same way. The difference consists in the surface

$$S = y^2 + (3x^2 + 4a)t^2 + f(x) \quad \subset \quad \mathbb{A}^3_{(x,y,t)} \qquad \text{[19, Equation (15)]}$$

or, equivalently,

$$S\colon -y^2 = x^3 + 3t^2x^2 + ax + 4at^2 + b \quad \subset \quad \mathbb{A}^3_{(x,y,t)}.$$

Note that the projection $\pi\colon S \to \mathbb{A}^1_x$ is a *conic bundle* [46, Definition 6]. The original SW encoding just picks a non-degenerate $\mathbb{F}_q$-fiber $\pi^{-1}(\beta) \subset \mathbb{A}^2_{(y,t)}$ (for some $\beta \in \mathbb{F}_q$) whose an $\mathbb{F}_q$-parametrization $\mathbb{A}^1 \dashrightarrow \pi^{-1}(\beta)$ is taken in the role of ψ.

According to the quite constructive result [46, Theorem 1], the surface S is also $\mathbb{F}_q$-unirational, alhought in general it is not $\mathbb{F}_q$-*rational* as stressed in [19, Section 5]. As before, it is proposed to chose a dominant $\mathbb{F}_q$-map $\psi\colon \mathbb{A}^2 \dashrightarrow S$ of degree as little as possible. Once again, if the resulting curves $C_{i,\alpha}$ (with respect to the new functions X_i) are absolutely irreducible, then the modified SW encoding h is admissible. Unfortunately, explicit formulas of ψ heavily depend on the specified a, b, q, hence we are not able to write out them generally.

Recently, Chávez-Saab, Rodriguez-Henriquez, and Tibouchi [41] completely studied the case when S is an $\mathbb{F}_q(x)$-rational conic, that is, it has an $\mathbb{F}_q(x)$-point. Thereby, they constructed the birational $\mathbb{F}_q$-map ψ (of degree one). Be careful that $\mathbb{F}_q$-rationality of the surface S does not imply $\mathbb{F}_q(x)$-rationality of S as a conic. The corresponding encoding h was called *SwiftEC*. It is relevant for many elliptic curves arising in practice. Inter alia, all ordinary curves of j-invariant 0 are covered.

Nevertheless, the applicability conditions of SwiftEC are too restrictive for a series of interesting curves among which $E\colon y^2 = x^3 + ax$ (of j-invariant 1728). That is why the work [42] does not lose significance. Moreover, the encoding h_1 invented in [11] is still much faster than SwiftEC over *highly* 2*-adic fields* (see Section 4). Lots of modern curves [47] (including BLS12-377) are defined over such fields. The point is that h_1 extracts a cubic root in $\mathbb{F}_q$ rather than a square one, not to mention two Legendre symbols.

In conclusion, the last four rows of Table 1 encourage to formulate a very beautiful and practically useful conjecture. Among others, it is related to Question 1.

Conjecture 1. *For any elliptic $\mathbb{F}_q$-curve E there is an admissible encoding $\mathbb{F}_q^2 \to E(\mathbb{F}_q)$ at the cost of one radical $\sqrt[n]{\cdot}$ in $\mathbb{F}_q$ for some $n \in \mathbb{N}$ (in particular, without computing additional power residue symbols $\left(\frac{\gamma}{q}\right)_m := \gamma^{(q-1)/m}$, where $\gamma \in \mathbb{F}_q$ and $m \mid q-1$).*

4 How to hash over highly 2-adic fields

As said in the title, this section dwells on the problem of hashing to an elliptic curve E defined over a finite field $\mathbb{F}_q$ such that $2^\nu \mid\mid q-1$ for a non-small $\nu \in \mathbb{N}$. According to Tables 1, 2, the majority of curves have only hash functions computing a root of even degree. Therefore, we cannot expect to represent such a root as one or at least several exponentiations in $\mathbb{F}_q$.

For simplicity, let's restrict to the case of a square root, but the arguments below are also true in the general case.

Undoubtedly, $\sqrt{\cdot} \in \mathbb{F}_q$ can be found through (constant-time) *Tonelli–Shanks's algorithm* [2, Appendix I.4]. However, it requires $O(\log(q) + \nu^2)$ operations in $\mathbb{F}_q$. The given drawback can be mitigated to $O\big(\log(q) + (\nu/\mu)^2\big)$ ones with $O(2^\mu \nu/\mu)$ storage (where μ is a parameter) by means of *Bernstein's table-lookup variant* [48]. In practice, this memory overhead is not significant for moderate 2-adicities such as the popular choice $\nu \approx 32$. These words are justified, for example, by the square root implementation [49] aimed at the classical point decompression [50]. By contrast, in the context of hashing to elliptic curves (assuming a secret input as earlier) Bernstein's approach is vulnerable to *cache-timing attacks* (cf. [51]). There is a vast literature about secure table lookups (see, e.g., [52]). However, it is desirable to completely avoid them if possible in order to be more confident in reliability.

Whenever j-invariant of E is different from 0, 1728, we can resort to the recent solution from [53]. Its novelty consists in an unexpected possibility to batch *Müller's square root algorithm* [54] and some encoding $h_M \colon \mathbb{F}_q \to E(\mathbb{F}_q)$, including $\sqrt{\cdot}$, in such a way that Müller's algorithm does not contain anymore the non-deterministic subroutine. Recall that the unique bottleneck of Müller's algorithm (and thereby of h_M) is computing the n-th element of a certain non-full *Lucas sequence* $V_i(\cdot, 1)$, where $n := (q-1)/4$. These sequences periodically appear in various areas of cryptography (see, e.g., [55, Section 6.3.2]).

For determining $V_n(\cdot, 1)$ we possess *Postl's algorithm* [56], which performs $\approx 2\log_2(q)$ multiplications in $\mathbb{F}_q$. In fact, there is a folklore trick [57, 58] reducing the running time to $\approx 2\log_2(q) - \nu$ ones. Taking it into account, Postl's algorithm may be faster than (or at least of the same performance as) the exponentiation in $\mathbb{F}_q$ to some power m of length $\approx \log_2(q)$ and of Hamming weight $\omega \lesssim \log_2(q)$. This happens when

$$2\log_2(q) - \nu \lesssim \log_2(q) + \omega \qquad \Leftrightarrow \qquad \log_2(q) \lesssim \nu + \omega.$$

In this estimation the conventional binary exponentiation method is applied for the sake of simplicity. In particular, the encodings from [53, Table 1] sometimes become slower than h_M.

In the author's opinion, the task of hashing in time $O(\log(q))$ to all elliptic $\mathbb{F}_q$-curves of j-invariant 1728 is most likely solvable. For instance, one of prospective approaches is mentioned in [53, Section 4]. Nonetheless, it is suggested to omit the case of $j = 1728$ curves in the present section, because they are not considered over highly 2-adic fields in today's real-world cryptography. The situation is radically opposite for $j = 0$ curves. So, it is worth explaining how an implementer should act if (s)he encounters the Weierstrass form $E \colon y^2 = x^3 + b$ such that $\sqrt{b} \notin \mathbb{F}_q$.

As usual, one first needs to check the existence of an $\mathbb{F}_q$-isogeny $\varphi \colon E' \to E$ of small (prime) degree ℓ from another elliptic curve E'. This can be done by means of [55, Theorem 25.4.6]. In this case, nothing prevents to employ the indirect encoding $\varphi \circ h_M$. To evaluate φ we are able to use classical *Vélu's formulas* [55, Section 25.1.1] requiring $O(\ell)$ operations in $\mathbb{F}_q$. Alternatively, there is in [59] the method called *square-root Vélu* and denoted by $\sqrt{\text{élu}}$ or $\sqrt{}$élu. It has the asymptotic complexity $\widetilde{O}\big(\sqrt{\ell}\big)$, that is, $O\big(\sqrt{\ell}\big)$ up to polylogarithmic factors. In turn, the work [60] establishes that the complexity is closer to $O(\ell^{\log_2(3)/2})$. Owing to those sources, $\sqrt{\text{élu}}$ becomes more efficient for $\ell \approx 100$.

Now, we proceed to the most painful remaining case. Obviously, $\sqrt{b} \in \mathbb{F}_{q^2}$, hence we can

enjoy the admissible encoding $h_1 \colon \mathbb{F}_{q^2}^2 \to E(\mathbb{F}_{q^2})$. Further, the trace map

$$\mathrm{Tr} \colon E(\mathbb{F}_{q^2}) \to E(\mathbb{F}_q) \qquad P \mapsto P + P^q$$

comes to the fore, where P^q is the point $\mathbb{F}_q$-conjugate to P. Eventually, we get the composition $\mathrm{Tr} \circ h_1 \colon \mathbb{F}_{q^2}^2 \to E(\mathbb{F}_q)$. By the way, Tr is also leveraged in *Sato–Hakuta's encoding* [61] relevant conversely for all curves of $j \neq 0$. It is cute, but useless (regardless of ν) if we bear in mind Tables 1, 2.

The trace map is known to be a surjective homomorphism (as confirmed in [62, Lemma 1]), hence it is regular. At the same time, Tr is realized as an algebraic $\mathbb{F}_q$-map with quite clear formulas (see, e.g., [13, Section 7.4.2]), which implies its samplability. Thus, the trace map is admissible and, in fact, so is $\mathrm{Tr} \circ h_1$ as it can be readily seen. One of disadvantages of this construction is that its domain has the bit length $4\lceil \log_2(q) \rceil$ instead of $2\lceil \log_2(q) \rceil$. Consequently, the execution time of a supplementary hash function $\eta \colon \{0,1\}^* \to \mathbb{F}_{q^2}^2$ is doubled as well.

Besides, the cubic root arising in h_1 takes place over $\mathbb{F}_{q^2}$ rather than $\mathbb{F}_q$. In contrast to a square $\mathbb{F}_{q^2}$-root [1, Algorithm 5.18], the author does not know how to express $\sqrt[3]{\cdot} \in \mathbb{F}_{q^2}$ through a few $\mathbb{F}_q$-roots. Since E is supposed to be an ordinary curve, $q \equiv 1 \pmod 3$ and hence the 3-adicities of $\mathbb{F}_q$ and $\mathbb{F}_{q^2}$ coincide. This means that $\sqrt[3]{\cdot} \in \mathbb{F}_{q^2}$ is represented as an exponentiation in $\mathbb{F}_{q^2}$ if and only if the same holds over $\mathbb{F}_q$, that is, $q \not\equiv 1 \pmod{27}$ due to [11, Lemma 6]. As above, denote by $\omega \lesssim 2\log_2(q)$ the Hamming weight of the corresponding power. By virtue of *Karatsuba's trick* [1, Algorithm 5.16], let's assume that 1 multiplication in $\mathbb{F}_{q^2}$ costs 3 ones in $\mathbb{F}_q$. As a result, a cubic $\mathbb{F}_{q^2}$-root is found after $\approx 2\log_2(q) + \omega$ multiplications in $\mathbb{F}_{q^2}$, which amounts to $\approx 6\log_2(q) + 3\omega$ ones in $\mathbb{F}_q$. To sum up, the encoding $\mathrm{Tr} \circ h_1$ has the linear complexity $O(\log(q))$ as desired unless the field $\mathbb{F}_q$ is highly 3-adic. Fortunately, such fields do not occur in practice to the author's knowledge.

Acknowledgements. The author expresses his gratitude to Antonio Sanso, Daira Hopwood, Evgeny Alekseev, Gottfried Herold, Jeffrey Burdges, Justin Drake, Oleg Taraskin, and Sergey Vasilyev for useful comments on the role of hashing to elliptic curves in real-world cryptography. Besides, it is impossible not to note the financial support provided by the Web3 Foundation (W3F) grant "Implementation of the new hash function to BLS12 curves".

References

[1] *Guide to pairing-based cryptography*, Cryptography and Network Security Series, ed. El Mrabet N., Joye M., Chapman and Hall/CRC, New York, 2017.

[2] Faz-Hernandez A., Scott S., Sullivan N., Wahby R. S., Wood C. A., *Hashing to elliptic curves*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-hash-to-curve, 2022.

[3] Brier E., Coron J.-S., Icart T., Madore D., Randriam H., Tibouchi M., "Efficient indifferentiable hashing into ordinary elliptic curves", Advances in Cryptology – CRYPTO 2010, LNCS, **6223**, ed. Rabin T., Springer, Berlin, Heidelberg, 2010, 237–254.

[4] Sakemi Y., Kobayashi T., Saito T., Wahby R. S., *Pairing-friendly curves*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-pairing-friendly-curves, 2022.

[5] Budroni A., Pintore F., "Efficient hash maps to $\mathbb{G}_2$ on BLS curves", *Applicable Algebra in Engineering, Communication and Computing*, 2020, 1–21.

[6] Fuentes-Castaneda L., Knapp E., Rodríguez-Henríquez F., "Faster hashing to $\mathbb{G}_2$", Selected Areas in Cryptography. SAC 2011, LNCS, **7118**, eds. Miri A., Vaudenay S., Springer, Berlin, Heidelberg, 2012, 412–430.

[7] Scott M., *Authenticated ID-based key exchange and remote log-in with simple token and PIN number*, https://eprint.iacr.org/2002/164, 2002.

[8] Pereira G., Doliskani J., Jao D., "x-only point addition formula and faster compressed SIKE", *Journal of Cryptographic Engineering*, **11**:1 (2021), 57–69.

[9] Boneh D., Gorbunov S., Wahby R. S., Wee H., Wood C. A., Zhang Z., *BLS signatures*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-bls-signature, 2022.

[10] Galbraith S. D., *CRYPTREC review of EdDSA*, https://www.cryptrec.go.jp/exreport/cryptrec-ex-3003-2020.pdf, 2020.

[11] 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 (2022), 801–812.

[12] 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 (2019), 154–179.

[13] *Handbook of elliptic and hyperelliptic curve cryptography*, Discrete Mathematics and Its Applications, **34**, eds. Cohen H., Frey G., Chapman and Hall/CRC, New York, 2005.

[14] Koshelev D., *Magma code*, https://github.com/dishport/Some-remarks-on-how-to-hash-faster-onto-elliptic-curves, 2022.

[15] Supranational, *blst/src/sqrt-addchain.h*, https://github.com/supranational/blst/blob/c76b5ac69a0044432d16cfd2cce60c93c8b01872/src/sqrt-addchain.h, 2020.

[16] Koshelev D., *Sage code*, https://github.com/dishport/Indifferentiable-hashing-to-ordinary-elliptic-curves-of-j-0-with-the-cost-of-one-exponen 2022.

[17] Vlasov A., *EIP-2539: BLS12-377 curve operations*, https://eips.ethereum.org/EIPS/eip-2539, 2020.

[18] Zhang Z., *Rust code*, https://github.com/zhenfeizhang/indifferentiable-hashing, 2023.

[19] Shallue A., van de Woestijne C. E., "Construction of rational points on elliptic curves over finite fields", Algorithmic Number Theory. ANTS 2006, LNCS, **4076**, eds. Hess F., Pauli S., Pohst M., Springer, Berlin, Heidelberg, 2006, 510–524.

[20] Tibouchi M., Kim T., "Improved elliptic curve hashing and point representation", *Designs, Codes and Cryptography*, **82**:1–2 (2017), 161–177.

[21] Hao F., "Prudent practices in security standardization", *IEEE Communications Standards Magazine*, **5**:3 (2021), 40–47.

[22] Crypto Forum Research Group (CFRG), *PAKE selection process*, https://github.com/cfrg/pake-selection, 2020.

[23] Abdalla M., Haase B., Hesse J., *CPace, a balanced composable PAKE*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-cpace, 2022.

[24] Bourdrez D., Krawczyk H., Lewi K., Wood C. A., *The OPAQUE asymmetric PAKE protocol*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-opaque, 2022.

[25] Abe M., Okamoto T., "Provably secure partially blind signatures", Advances in Cryptology – CRYPTO 2000, LNCS, **1880**, eds. Bellare M., Springer, Berlin, Heidelberg, 2000, 271–286.

[26] Davidson A., Faz-Hernández A., Sullivan N., Wood C. A., *Oblivious pseudorandom functions (OPRFs) using prime-order groups*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-voprf, 2022.

[27] Jarecki S., Kiayias A., Krawczyk H., "Round-optimal password-protected secret sharing and T-PAKE in the password-only model", Advances in Cryptology – ASIACRYPT 2014, LNCS, **8874**, eds. Sarkar P., Iwata T., Springer, Berlin, Heidelberg, 2014, 233–253.

[28] Jarecki S., Kiayias A., Krawczyk H., Xu J., "Highly-efficient and composable password-protected secret sharing (or: How to protect your Bitcoin wallet online)", 2016 IEEE European Symposium on Security and Privacy (EuroS&P), 2016, 276–291.

[29] Icart T., "How to hash into elliptic curves", Advances in Cryptology – CRYPTO 2009, LNCS, **5677**, eds. Halevi S., Springer, Berlin, Heidelberg, 2009, 303–316.

[30] Farashahi R. R., Fouque P.-A., Shparlinski I. E., Tibouchi M., Voloch J. F., "Indifferentiable deterministic hashing to elliptic and hyperelliptic curves", *Mathematics of Computation*, **82**:281 (2013), 491–512.

[31] Koshelev D., "Faster point compression for elliptic curves of j-invariant 0", *Mathematical Aspects of Cryptography*, **12**:4 (2021), 115–123.

[32] Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI), *Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français*, https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000024668816, 2011.

[33] 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)*, https://csrc.nist.gov/publications/detail/sp/800-186/final, 2023.

[34] Alekseev E. K., Nikolaev V. D., Smyshlyaev S. V., "On the security properties of Russian standardized elliptic curves", *Mathematical Aspects of Cryptography*, **9**:3 (2018), 5–32.

[35] Fried M. D., "Global construction of general exceptional covers, with motivation for applications to encoding", Finite Fields: Theory, Applications, and Algorithms, Contemporary Mathematics, **168**, eds. Mullen G. L., Shiue P. J., American Mathematical Society, Providence, 1994, 69–100.

[36] Tibouchi M., "Impossibility of surjective Icart-like encodings", Provable Security. ProvSec 2014, LNCS, **8782**, eds. Chow S. S. M., Liu J. K., Hui L. C. K., Yiu S. M., Springer, Cham, 2014, 29–39.

[37] Koshelev D., "Optimal encodings to elliptic curves of j-invariants 0, 1728", *SIAM Journal on Applied Algebra and Geometry*, **6**:4 (2022), 600–617.

[38] *The eightfold way: The beauty of Klein's quartic curve*, Mathematical Sciences Research Institute Publications, **35**, eds. Levi S., Cambridge University Press, Cambridge, 1999.

[39] Magma group, *Automorphism groups of curves*, https://magma.maths.usyd.edu.au/magma/handbook/text/1415#16022.

[40] Skałba M., "Points on elliptic curves over finite fields", *Acta Arithmetica*, **117**:3 (2005), 293–301.

[41] Chávez-Saab J., Rodriguez-Henriquez F., Tibouchi M., "SWIFTEC: Shallue–van de Woestijne indifferentiable function to elliptic curves", Advances in Cryptology – ASIACRYPT 2022, LNCS, **13791**, eds. Agrawal S., Lin D., Springer, Cham, 2022, 63–92.

[42] Koshelev D., "The most efficient indifferentiable hashing to elliptic curves of j-invariant 1728", *Journal of Mathematical Cryptology*, **16**:1 (2022), 298–309.

[43] Pornin T., *X25519 implementation for ARM Cortex-M0/M0+*, https://github.com/pornin/x25519-cm0, 2020.

[44] Hamburg M., *Computing the Jacobi symbol using Bernstein–Yang*, https://eprint.iacr.org/2021/1271, 2021.

[45] Moret-Bailly L., "Variétés stablement rationnelles non rationnelles", Séminaire Bourbaki: volume 1984/85, report no. 643, *Astérisque*, **133-134** (1986), 223–236.

[46] Kollár J., Mella M., "Quadratic families of elliptic curves and unirationality of degree 1 conic bundles", *American Journal of Mathematics*, **139**:4 (2017), 915–936.

[47] Aranha D. F., El Housni Y., Guillevic A., "A survey of elliptic curves for proof systems", https://link.springer.com/article/10.1007/s10623-022-01135-y, *Designs, Codes and Cryptography*, 2022.

[48] Bernstein D. J., *Faster square roots in annoying finite fields*, https://cr.yp.to/papers.html#sqroot, 2001.

[49] Herold G., *field_element_square_root.go*, https://github.com/GottfriedHerold/Bandersnatch/blob/main/bandersnatch/fieldElements/field_element_square_root.go, 2023.

[50] Hagopian I., *Bandersnatch sqrt optimization notes*, https://hackmd.io/@jsign/bandersnatch-optimized-sqrt-notes, 2023.

[51] Bernstein D. J., *Cache-timing attacks on AES*, https://cr.yp.to/papers.html#cachetiming, 2005.

[52] Tromer E., Osvik D. A., Shamir A., "Efficient cache attacks on AES, and countermeasures", *Journal of Cryptology*, **23** (2010), 37–71.

[53] Koshelev D., *Batching Cipolla–Lehmer–Müller's square root algorithm with hashing to elliptic curves*, https://eprint.iacr.org/2023/390, 2023.

[54] Müller S., "On the computation of square roots in finite fields", *Designs, Codes and Cryptography*, **31**:3 (2004), 301–312.

[55] Galbraith S. D., *Mathematics of public key cryptography*, Cambridge University Press, New York, 2012.

[56] Postl H., "Fast evaluation of Dickson polynomials", *Contributions to General Algebra*, **6** (1988), 223–225.

[57] Joye M., Quisquater J.-J., "Efficient computation of full Lucas sequences", *Electronics Letters*, **32**:6 (1996), 537–538.

[58] Lambert R. J., *Method to calculate square roots for elliptic curve cryptography*, United States patent No. 9148282B2, https://patents.google.com/patent/US9148282B2/en, 2013.

[59] Bernstein D. J., De Feo L., Leroux A., Smith B., "Faster computation of isogenies of large prime degree", *The Open Book Series*, **4**:1 (2020), 39–55.

[60] Adj G., Chi-Domínguez J. J., Rodríguez-Henríquez F., "Karatsuba-based square-root Vélu's formulas applied to two isogeny-based protocols", *Journal of Cryptographic Engineering*, **13**:1 (2023), 89–106.

[61] Sato H., Hakuta K., "An efficient method of generating rational points on elliptic curves", *Journal of Math-for-Industry*, **1**:A (2009), 33–44.

[62] Shparlinski I. E., Voloch J. F., "Generators of elliptic curves over finite fields", *Bulletin of the Institute of Mathematics, Academia Sinica (New Series)*, **9**:4 (2014), 657–670.