Batching Cipolla–Lehmer–Müller's square root algorithm with hashing to elliptic curves

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

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

Abstract. The present article provides a novel hash function $\mathcal{H}$ to any elliptic curve of j-invariant $\neq 0, 1728$ over a finite field $\mathbb{F}_q$ of large characteristic. The unique bottleneck of $\mathcal{H}$ consists in extracting a square root in $\mathbb{F}_q$ as well as for most hash functions. However, $\mathcal{H}$ is designed in such a way that the root can be found by (Cipolla–Lehmer–)Müller's algorithm in constant time. Violation of this security condition is known to be the only obstacle to applying the given algorithm in the cryptographic context. When the field $\mathbb{F}_q$ is highly 2-adic and $q \equiv 1 \pmod 3$, the new batching technique is the state-of-the-art hashing solution except for some sporadic curves. Indeed, Müller's algorithm costs $\approx 2\log_2(q)$ multiplications in $\mathbb{F}_q$. In turn, original Tonelli–Shanks's square root algorithm and all of its subsequent modifications have the algebraic complexity $\Theta(\log(q) + g(\nu))$, where ν is the 2-adicity of $\mathbb{F}_q$ and a function $g(\nu) \neq O(\nu)$. As an example, it is shown that Müller's algorithm actually needs several times fewer multiplications in the field $\mathbb{F}_q$ (whose $\nu = 96$) of the standardized curve NIST P-224.

Keywords: Cipolla–Lehmer–Müller's algorithm · conic bundles · generalized Châtelet surfaces · genus 2 curves of zero trace · gluing elliptic curves · hashing to elliptic curves · highly 2-adic fields · unirationality problem

1 Introduction

The idea of this paper came to the author when he was working on the other recent one [32] on the same topic. There and here one addresses the problem of efficient hashing to elliptic curves $E\colon y^2 = f(x) := x^3 + ax + b$ over *highly* 2*-adic fields* $\mathbb{F}_q$ of characteristic $p > 3$. By definition, $q - 1 = 2^\nu m$ for $\nu, m \in \mathbb{N}$ and ν is quite large. A lot of modern elliptic curves are defined over such finite fields (see, e.g., [4]), because they are the most suitable for the fast Fourier transform (FFT). At the same time, it is a favourite tool of developers belonging to the zero-knowledge (ZK) community.

Over highly 2-adic fields for extraction of a square root $\sqrt{\cdot} \in \mathbb{F}_q$ one usually prefers *Müller's algorithm* [37], which is an enhancement of the classical *Cipolla–Lehmer's algorithm* [13,35]. Unfortunately, the first stage of the algorithm is not deterministic in contrast to its subsequent one. So, applying the given algorithm in cryptography is often not safe with regards to timing attacks. That is why in the RFC [17, Appendix I], devoted to hashing to elliptic curves, Müller's algorithm is not contained.

The authors of the mentioned RFC are content only with a constant-time version of *Tonelli–Shanks's algorithm* [41,48]. The point is that the probabilistic part of the latter (unlike Müller's algorithm) does not depend on an input quadratic residue, but only on q. Meanwhile, at least in elliptic curve cryptography the field $\mathbb{F}_q$ is fixed all the time. The problem is that (original) Tonelli–Shanks's algorithm becomes extremely slow for 2-adicity $\nu \to \log_2(q)$, namely it carries out $\Theta(\log(q) + \nu^2)$ operations in $\mathbb{F}_q$.

There are enhancements of the given algorithm based on faster discrete logarithm computation in the 2-power subgroup $(\mathbb{F}_q^*)^m = \mu_{2^\nu}$. The sources [38], [42, Section 12.5.1] employ the divide-and-conquer strategy to obtain the subquadratic complexity $\Theta(\log(q) + \nu\log(\nu))$. Finally, Sutherland [44] invents a variant working in time

$$\Theta\biggl(\log(q) + \frac{\nu\log(\nu)}{\log(\log(\nu))}\biggr).$$

In any case, Sutherland's algorithm asymptotically loses to Müller's one as ν (and so $\log_2(q)$) tending to infinity.

It is impossible not to remark that all the Tonelli–Shanks-type methods can be even more accelerated with the help of precomputed tables containing in total $\Theta(2^\omega\nu/\omega)$ finite field elements, where $\omega \in \mathbb{N}$ is an auxiliary parameter. In addition to exponential growth in ω of occupied memory, this approach is in fact not a panacea in the constant-time setting. Indeed, as said in [38], *"All table lookups must use a constant-time process which reads all entries and combines them using Boolean operations to retain only the value of the correct entry. The cost of a lookup is then proportional to the table size* (2^ω)*, which disfavours large tables."*.

In view of all the above, in [32] the author tries to bypass painful square root computation during hashing to E. For this purpose, he provides some cubic $\mathbb{F}_q$-polynomial in one variable having a unique $\mathbb{F}_q$-root. Since its coefficients depend on an element of the field $\mathbb{F}_q$, this eventually results in a desired hash function. The approach of this article is cardinally opposite. Instead of computing $\mathbb{F}_q$-roots of higher-degree $\mathbb{F}_q$-polynomials, we will make Müller's algorithm completely deterministic. This turns out to be possible, because we are free to generate specific quadratic residues in $\mathbb{F}_q$ equipped with additional data.

Let's pick once and for all any quadratic non-residue $v \in \mathbb{F}_q$. Suppose that we possess a quadratic residue z^2 with the unknown square root $z \in \mathbb{F}_q$. Recall that Cipolla–Lehmer–Müller's algorithm of determining z starts with searching for an element $x \in \mathbb{F}_q$ such that $x^2 - z^2$ is a non-square in $\mathbb{F}_q$. Put another way, $x^2 - z^2 = vy^2$ for some $y \in \mathbb{F}_q$. In Müller's paper [37] the expression $z^2x^2 - 4$ is instead chosen, but this of course does not play any role. There is a long-standing open problem about how to find x in constant polynomial time and without assuming unproven conjectures of number theory. By the way, the more malleable problem of constructing an arbitrary quadratic non-residue in $\mathbb{F}_q$ is solved in [46].

Substituting the separable cubic $\mathbb{F}_q$-polynomial $f(t)$ to the place of z^2, we get the so-called *Châtelet surface* $S_f : x^2 - vy^2 = f(t)$ originating from [12]. This equation can be imagined as an example of the function field analogue of *general(ized) Pell's equation* [3, Chapter 4]. We deal with an absolutely irreducible cubic surface different from the cone over a plane cubic curve. Moreover, $S_f(\mathbb{F}_q) \neq \emptyset$ as the field $\mathbb{F}_q$ is always large in cryptography. As a result, S_f is $\mathbb{F}_q$-*unirational* according to [24], that is, there is a rational (not necessarily proper) $\mathbb{F}_q$-parametrization $\pi\colon \mathbb{A}^2 \to S_f$.

Thereby, we are able to generate for free points $(x, y, t) \in S_f(\mathbb{F}_q)$ to execute Müller's algorithm of finding $\sqrt{f(t)}$. Only the element x is essentially necessary in the algorithm, but y will not hurt in its low-level optimizations. The trouble is that $f(t)$ may be a non-square in $\mathbb{F}_q$. So, the parametrization π does not give a hash function to $E(\mathbb{F}_q)$, but just to $E(\mathbb{F}_q) \cup E^T(\mathbb{F}_q)$, where $E^T : vy^2 = f(x)$ is the (unique) quadratic twist of E.

To fix the given imperfection, it is suggested to consider a genus 2 curve $H : s^2 = h(t)$ having two (quadratic) $\mathbb{F}_q$-covers $\varphi\colon H \to E$ and $\varphi^T : H \to E^T$. We will derive the desired H for all curves E of j-invariants $\neq 0, 1728$, i.e., of the coefficients $a, b \neq 0$. In addition, introduce $H^T : vs^2 = h(t)$, the quadratic hyperelliptic twist of H. Up to the isomorphism $(x, y) \mapsto (x, vy)$, formulas of φ^T equally define an $\mathbb{F}_q$-cover $H^T \to E$. By abuse of notation, it will be also denoted by φ^T. It turns out that the *generalized Châtelet surface* $S_h : x^2 - vy^2 = h(t)$ is still $\mathbb{F}_q$-unirational for our degree 6 polynomials $h(t)$. And the corresponding formulas are elementary. Thus, we will get a way to map into $E(\mathbb{F}_q)$ through $H(\mathbb{F}_q) \cup H^T(\mathbb{F}_q)$.

The equation of Châtelet surfaces is very similar to that of elliptic curves. So, it is no wonder that these surfaces occur in the context of elliptic curve cryptography. In Skałba's seminal work [43, Lemma 2] a certain Châtelet surface is also utilized (but in another way) for hashing to elliptic curves provided that $a \neq 0$. Curiously, the other remarkable sources [11, Section 3.1], [40, Section 5] on the topic are based on a surface resembling a Châtelet one. Furthermore, whenever $a = 0$, the former becomes the latter. This is a rough explanation why Chavez-Saab et al.'s (indifferentiable) hash function *SwiftEC* from [11] is always valid for curves of j-invariant 0.

1.1 Alternative hash-to-curve functions

In the literature, there exist numerous hash functions of the wanted form $\{0, 1\}^* \to E(\mathbb{F}_q)$. The author collects state-of-the-art ones in [28, Tables 1, 2], because the RFC [17] is rather outdated as this regularly happens with RFCs. Inter alia, the hash-to-curve RFC misses SwiftEC, although this hash function is recognized to be the best for the majority of elliptic curves defined over non-highly 2-adic fields. Recall that finding $\sqrt{\cdot} \in \mathbb{F}_q$ (via an exponentiation in $\mathbb{F}_q$) is the only bottleneck of SwiftEC as well as most hash functions, especially to curves of j-invariants $\neq 0, 1728$. As an example, SwiftEC is appropriate for the popular standardized curves NIST P-256 and Curve25519 (a.k.a. Edwards25519) from [10, Section 3.2].

Let's now talk about existing hash functions friendly to fields of large 2-adicities ν. First of all, the hash function $\mathcal{H}_{old}$ from [32] is relevant only for curves having an $\mathbb{F}_q$-isogeny of degree 3, which is a pretty restrictive condition. Moreover, it requires $\gtrsim 4\log_2(q) - \nu$ multiplications in $\mathbb{F}_q$, which is a fairly large number. We will construct a new hash function $\mathcal{H}$ improving the former on the both indicators. It is applicable to all elliptic $\mathbb{F}_q$-curves of j-invariants $\neq 0, 1728$. At the same time, its running time amounts to that of Müller's algorithm, namely to $\approx 2\log_2(q)$ multiplications in $\mathbb{F}_q$. As seen, $\mathcal{H}$ has to perform $\gtrsim \log_2(q)$ fewer field multiplications than $\mathcal{H}_{old}$ even for the largest $\nu \approx \log_2(q)$.

Whenever $q \equiv 2 \pmod 3$, we can utilize *Icart's hash function* $\mathcal{H}_I$ [23], which extracts a (unique) cubic root in $\mathbb{F}_q$ instead of a square one. The solution of Icart is thereby optimal for the given case. Nevertheless, the opposite case $q \equiv 1 \pmod 3$ arises quite often. For instance, this is known to be a necessary condition for the ordinariness of curves $E_b\colon y^2 = x^3 + b$ of j-invariant 0. Therefore, $\mathcal{H}_I$ is absolutely useless for them. Meanwhile, ordinary (a.k.a non-supersingular) curves E_b are very attractive, especially in pairing-based cryptography, because they (and only they) enjoy order 6 automorphisms and degree 6 twists. This positively influences the efficiency of diverse operations on E_b.

The work [29] succeeds in obtaining an indifferentiable hash function $\mathcal{H}_3$ to E_b provided that $\sqrt{b} \in \mathbb{F}_q$ and hence $3 \mid \#E_b(\mathbb{F}_q)$. Surprisingly, it also extracts $\sqrt[3]{\cdot} \in \mathbb{F}_q$, but in the desired case $q \equiv 1 \pmod 3$. Since highly 3-adic fields are not so popular in practice as their 2-adic counterparts, $\mathcal{H}_3$ costs one exponentiation in $\mathbb{F}_q$, at any rate for $q \not\equiv 1 \pmod{27}$. The order 3 automorphism $[\omega](x, y) := (\omega x, y)$ on E_b, where $\omega := \sqrt[3]{1} \neq 1$ and $\omega \in \mathbb{F}_q$, underlies the established result. Unfortunately, other elliptic curves do not possess a non-trivial automorphism of odd order. Consequently, the result cannot be generalized, at least staying within elliptic curves.

One more hashing method unsaid earlier is the naive one from [15, Section 8.1], i.e., scalar multiplication on E with a fixed $\mathbb{F}_q$-point, but with a variable scalar. This hashing can obviously work in constant linear time $\Theta(\log(q))$ as we wish, but it is often insecure. That source demonstrates how to forge a signature without a secret key in the *BLS (Boneh–Lynn–Shacham) signature scheme* [15, Section 1.4.3]. More generally, the so-called *Pedersen hash function* $\mathcal{H}_{Ped}$ (see, e.g., [6]) with $n \in \mathbb{N}$ "independent" points $P_k \in E(\mathbb{F}_q)$ is likewise unsafe. It is easily shown that, given n known pairs (message, signature), a malicious entity is able to sign any message by solving a system of linear equations. And it is not even required to know a non-trivial linear relation between P_k. The same conclusions can be found at the end of [23, Section 1] for the *Boneh–Franklin IBE (identity-based encryption)* [15, Section 1.6.4].

In both mentioned schemes the input numerical arguments m_k of $\mathcal{H}_{Ped}$ are public. The attack seemingly does not work whenever m_k are secret as in the setting of the current article. However, there is a danger that some inputs m_k will be leaked when their relevance is outdated. An adversary may try to somehow find any n of them, because over time, less and less effort will be made to ensure the secrecy of old m_k. Even if they are assumed to be destroyed, there is a chance that copies of this data will be saved on one of devices by accident or by intent. The attack can be mitigated by taking large values n. But in this case, the hashing complexity becomes equal to $\Theta(n\log(q))$, i.e., it is not linear anymore. It is also worth noting that $\mathcal{H}_{Ped}$ is not a random oracle regardless of n, hence many protocols using $\mathcal{H}_{Ped}$ can no longer be considered provably secure.

2 Algebraic geometry preliminaries

Let $\mathbb{F}_q$ be a finite field of characteristic $p > 3$ and 2-adicity $\nu > 1$. The last assumption means that $\sqrt{-1} \in \mathbb{F}_q$. For our objectives, it will be more convenient to work with the more general form

$$E\colon y^2 = f(x) := x^3 + a_2x^2 + a_4x + a_6$$

of an elliptic $\mathbb{F}_q$-curve. It still has the unique infinity point $\infty := (0:1:0) \in \mathbb{P}^2$. It is helpful to have before our eyes the expression of the j-invariant

$$j(E) = \frac{-2^8(a_2^2 - 3a_4)^3}{4a_2^3a_6 - a_2^2a_4^2 - 18a_2a_4a_6 + 4a_4^3 + 27a_6^2}. \tag{1}$$

Besides, denote by r_0, r_1, r_2 the (pairwise distinct) roots of the polynomial $f(x)$. As usual,

$$a_2 = -(r_0 + r_1 + r_2), \qquad a_4 = r_0r_1 + r_0r_2 + r_1r_2, \qquad a_6 = -r_0r_1r_2.$$

Fix once and forever a quadratic non-residue $v \in \mathbb{F}_q$. Consider the quadratic twist $E^T : vy^2 = f(x)$ having the Weierstrass form

$$y^2 = f^T(x) := x^3 + a_2vx^2 + a_4v^2x + a_6v^3.$$

By abuse of notation, E^T will also stand for this form. There is the $\mathbb{F}_{q^2}$-isomorphism

$$\theta\colon E \to E^T \qquad (x, y) \mapsto (vx, v\sqrt{v}\cdot y).$$

Obviously, $\theta(r_k, 0) = (vr_k, 0)$, that is, vr_k are roots of $f^T(x)$.

As is clear from the introduction, generalized Châtelet surfaces

$$S_h\colon x^2 - vy^2 = h(t) \quad \subset \quad \mathbb{A}^3_{(x,y,t)},$$

with separable $\mathbb{F}_q$-polynomials $h(t)$, are main geometric objects of the current article. They are ones of the simplest examples of *conic bundles* or, alternatively, of conics over the function field $\mathbb{F}_q(t)$. It is useful to remember that we are primarily interested in elements $t \in \mathbb{F}_q$ for which trivially $\mathbb{F}_q(t) = \mathbb{F}_q$. Conic bundles are a fairly common tool in applied mathematics. For instance, as seen in [26,27], they appear in the context of compressing points on elliptic curves of j-invariant 0.

Below, we will tacitly use the program code [33] written in Magma to verify underlying formulas. Among other things, we need the following folklore result about *blowing up and down* [19, Section V.3]. It will be later actively utilized to successively simplify the surfaces S_h by reducing polynomials $h(t)$.

Lemma 1. *Assume that a quadratic $\mathbb{F}_q$-polynomial $Q(t) = t^2 - Tt + N$ is irreducible, i.e., its discriminant $D := T^2 - 4N$ is a quadratic non-residue in $\mathbb{F}_q$. Then, we have the blow-up $\mathbb{F}_q$-maps*

$$bl_{Q,\pm}\colon S_h \to S_{hQ} \qquad (x,y) \mapsto \Big(\Big(t - \frac{T}{2}\Big)x \pm \frac{\sqrt{Dv}}{2}y,\ \pm\frac{\sqrt{Dv}}{2v}x + \Big(t - \frac{T}{2}\Big)y\Big),$$

identical on t. They are linear transformations whose determinant is equal to $Q(t)$. In particular, the maps $bl_{Q,\pm}$ are invertible for every $t \in \mathbb{F}_q$.

Corollary 1. *For $T = 0$ and the non-square $d := -N$ the blow-up maps from the previous lemma take the form*

$$bl_{Q,\pm}\colon S_h \to S_{hQ} \qquad (x,y) \mapsto \Big(tx \pm \sqrt{dv}\cdot y,\ \pm\sqrt{\frac{d}{v}}\cdot x + ty\Big).$$

By default, put $bl_Q := bl_{Q,+}$. The notation $S_h(\alpha)$ will mean the fiber of S_h over an arbitrary element $\alpha \in \overline{\mathbb{F}_q}$. Evidently, it is degenerate if and only if α is a root of $h(t)$. In this circumstance,

$$S_h(\alpha) = L_+(\alpha) \cup L_-(\alpha), \qquad \text{where} \qquad L_\pm(\alpha) := \begin{cases} x = \pm\sqrt{v}\cdot y, \\ t = \alpha. \end{cases}$$

More concretely, let $\alpha_\pm := (T \pm \sqrt{D})/2$ be the roots of $Q(t)$. In geometric terms, the blow-down, i.e., inverse map $bl^{-1}_{Q,+}\colon S_{hQ} \to S_h$ (respectively, $bl^{-1}_{Q,-}\colon S_{hQ} \to S_h$) contracts the two $\mathbb{F}_q$-conjugate lines $L_\pm(\alpha_\pm)$ (respectively, $L_\pm(\alpha_\mp)$) on the surface S_{hQ} to two $\mathbb{F}_q$-conjugate points on the one S_h.

Throughout the section, we will encounter the quadratic cone $S_c \subset \mathbb{A}^3_{(x,y,t)}$ over the plane conic $C\colon x^2 - vy^2 = c$ with $c \in \mathbb{F}_q^*$. The latter has the $\mathbb{F}_q$-point

$$P_0 := \begin{cases} (\sqrt{c}, 0) & \text{if} \quad \sqrt{c} \in \mathbb{F}_q, \\ \Big(0, \sqrt{\dfrac{-c}{v}}\Big) & \text{if} \quad \sqrt{c} \notin \mathbb{F}_q. \end{cases}$$

It is a classical fact (see, e.g., [14, Section 3.1]) that, given an abstract conic $C : ax^2 + by^2 + 1 = 0$ having a point $P_0 = (x_0, y_0)$, the map inverse to the projection of C from P_0 has the form

$$pr^{-1}_{P_0}\colon \mathbb{A}^1_u \to C \qquad u \mapsto \Big(\frac{ax_0u^2 + 2by_0u - bx_0}{au^2 + b},\ \frac{ay_0u^2 - 2ax_0u - by_0}{au^2 + b}\Big).$$

In our situation, $a = -1/c$ and $b = v/c$. As a result, acting identically on t, we obtain the map

$$pr_{P_0}^{-1}\colon \mathbb{A}^2_{(u,t)} \to S_c \qquad u \mapsto \begin{cases} \Big(\sqrt{c}\cdot\dfrac{u^2+v}{u^2-v},\ \sqrt{c}\cdot\dfrac{-2u}{u^2-v}\Big) & \text{if} \quad \sqrt{c} \in \mathbb{F}_q, \\ \Big(\sqrt{\dfrac{-c}{v}}\cdot\dfrac{-2vu}{u^2-v},\ \sqrt{\dfrac{-c}{v}}\cdot\dfrac{u^2+v}{u^2-v}\Big) & \text{if} \quad \sqrt{c} \notin \mathbb{F}_q \end{cases}$$

with the same notation. Conveniently, the denominator $u^2 - v$ does not vanish for $u \in \mathbb{F}_q$.

Hereafter, we proceed to analyzing step by step several separate cases. First, in Section 2.1 we will consider the scenario in which one of the roots r_k, e.g., r_0 belongs to $\mathbb{F}_q$ or, equivalently, $r_0 = 0$. There are two possible subcases when the remaining roots r_1, $r_2 \in \mathbb{F}_q$ (Section 2.1.1) and conversely r_1, $r_2 \notin \mathbb{F}_q$ (Section 2.1.2). Finally, the most complicated situation arises in Section 2.2, where none of the roots r_k lies in $\mathbb{F}_q$.

2.1 The case when $r_0 \in \mathbb{F}_q$

Without loss of generality, put $r_0 = a_6 = 0$ and $O := (0, 0)$. Under this premise, $a_2 = -(r_1 + r_2)$ and $a_4 = r_1 r_2$. Let's glue the curves E, E^T along their 2-torsion subgroups as follows:

$$\psi\colon E[2] \to E^T[2] \qquad O \mapsto O, \quad (r_1, 0) \mapsto (vr_2, 0), \quad (r_2, 0) \mapsto (vr_1, 0).$$

No matter $r_1, r_2 \in \mathbb{F}_q$ or not, the map ψ respects the Frobenius action on $E[2]$ and $E^T[2]$. In addition, note that $\psi \neq \theta|_{E[2]}$.

Owing to [22, Section 3.2], there are two quadratic $\mathbb{F}_q$-covers

$$\begin{aligned} \varphi\colon H \to E \qquad & (t,s) \mapsto \Big(\frac{a_4(vt^2+1)}{-va_2t^2},\ \frac{a_4}{a_2^2t^3}\cdot s\Big), \\ \varphi^T\colon H \to E^T \qquad & (t,s) \mapsto \Big(\frac{a_4(vt^2+1)}{-a_2},\ \frac{va_4}{a_2^2}\cdot s\Big) \end{aligned} \tag{2}$$

from a genus 2 curve $H\colon s^2 = h(t)$. Here,

$$h(t) := c\cdot Q_0(t)\cdot Q_1(t)\cdot Q_2(t) = c(t^6 + b_2t^4 + b_4t^2 + b_6), \tag{3}$$

where $Q_k(t) := t^2 - \delta_k$ and

$$c := -a_2a_4, \qquad \delta_0 := -\frac{1}{v}, \qquad \delta_1 := \frac{r_1}{vr_2}, \qquad \delta_2 := \frac{r_2}{vr_1}, \tag{4}$$

$$b_2 := -(\delta_0 + \delta_1 + \delta_2), \qquad b_4 := \delta_0\delta_1 + \delta_0\delta_2 + \delta_1\delta_2, \qquad b_6 := -\delta_0\delta_1\delta_2 = \frac{1}{v^3}.$$

Whenever $j(E) \neq 1728$ (as supposed), it is clear that $r_1 \neq \pm r_2$, i.e., $a_2 \neq 0$. As a consequence, $c \neq 0$ and the covers φ, φ^T are correctly defined. Keep in mind that $\varphi(0, \sqrt{cb_6}) = \infty$ and besides $\sqrt{cb_6} \in \mathbb{F}_q \Leftrightarrow \sqrt{c} \notin \mathbb{F}_q$.

For the sake of convenience, put $\gamma_k := \sqrt{\delta_k}$. For our polynomial $h(t)$ the generalized Châtelet surface S_h fits with that discussed by Swinnerton-Dyer [45]. Furthermore, the polynomial $Q_0(t)$ is irreducible over $\mathbb{F}_q$, i.e., $\gamma_0 \not\in \mathbb{F}_q$, hence we are able to eliminate it by virtue of Corollary 1. Thereby, we get an (ordinary) Châtelet surface S_{h_0} for which

$$h_0(t) := c{\cdot}Q_1(t){\cdot}Q_2(t) = c(t^4 + d_2t^2 + d_4),$$

$$d_2 := -(\delta_1 + \delta_2), \qquad d_4 := \delta_1\delta_2 = \frac{1}{v^2}.$$

2.1.1 The case when all $r_k \in \mathbb{F}_q$

The subcase $\sqrt{a_4} \in \mathbb{F}_q$ (*still* $r_0 = 0$). If so, then $\gamma_1, \gamma_2 \not\in \mathbb{F}_q$ or, equivalently, the polynomials $Q_1(t)$, $Q_2(t)$ are irreducible over $\mathbb{F}_q$. Therefore, nothing prevents to likewise eliminate the remaining degenerate fibers of S_{h_0}, arriving at the quadratic cone S_c.

The subcase $\sqrt{a_4} \not\in \mathbb{F}_q$ (*still* $r_0 = 0$). It is the most difficult, because the degenerate fibers of the surface S_{h_0}, viz. $S_{h_0}(\pm\gamma_1)$, $S_{h_0}(\pm\gamma_2)$ cannot be liquidated over $\mathbb{F}_q$. Indeed, $\gamma_1, \gamma_2 \in \mathbb{F}_q$, hence every of them consists of a pair of $\mathbb{F}_q$-conjugate lines. Shifting, e.g., γ_1 to the infinity point $(1:0) \in \mathbb{P}^1$, we immediately obtain a cubic surface birationally $\mathbb{F}_q$-isomorphic to S_{h_0}. Due to [24], we thus have a constructive proof of $\mathbb{F}_q$-unirationality of S_{h_0}.

In fact, one can reduce the subcase under consideration to the case from Section 2.1.2 with the help of the next lemma. In this way, we are able to hash into E through a transitional elliptic $\mathbb{F}_q$-curve by analogy with [11, Section 4.3], [49, Section 4.3].

Lemma 2. *Whenever we are in the subcase conditions* (*and* $j(E) \neq 1728$), *there is an elliptic* $\mathbb{F}_q$*-curve* $E' \colon y^2 = x(x^2 + A_2x + A_4)$ (*also of* j*-invariant* $\neq 1728$) *such that* E, E' *are* 2*-isogenous over* $\mathbb{F}_q$ *and* O *is the only* $\mathbb{F}_q$*-point of order* 2 *on* E'.

Proof. As seen in [18, Example 9.6.9], the quotient curve $E' := E/O$ possesses the coefficients $A_2 = -2a_2$ and $A_4 = a_2^2 - 4a_4$. In accordance with the formula (1) applied to E',

$$j(E') = 1728 \qquad \Leftrightarrow \qquad A_2 = 0 \text{ or } A_4 = \frac{2A_2^2}{9} \qquad \Leftrightarrow \qquad a_2 = 0 \text{ or } a_4 = \frac{a_2^2}{6^2}.$$

The second equality is never fulfilled, since $\sqrt{a_4} \not\in \mathbb{F}_q$. So, we do not hit a $j = 1728$ curve if the initial curve E is not. By the same reason, the discriminant $A_2^2 - 4A_4 = 4^2a_4$ is a non-square. This is nothing but the lemma's statement. □

2.1.2 The case when $r_0 \in \mathbb{F}_q$, but $r_1, r_2 \not\in \mathbb{F}_q$.

If so, then $\delta_1, \delta_2 \not\in \mathbb{F}_q$ as well. However, $\gamma_1, \gamma_2 \in \mathbb{F}_{q^2}$, because the norm $\mathrm{N}_{\mathbb{F}_{q^2}/\mathbb{F}_q}(\delta_k) = d_4$ is a quadratic residue in $\mathbb{F}_q$. Not losing generality, let $\gamma_1^q = \gamma_2$. We have the factorization $h_0(t) = c{\cdot}Q_+(t){\cdot}Q_-(t)$ into two irreducible quadratic $\mathbb{F}_q$-polynomials

$$Q_\pm(t) := (t \mp \gamma_1)(t \mp \gamma_2) = t^2 \mp (\gamma_1 + \gamma_2)t + \frac{1}{v}.$$

Applying twice Lemma 1, we again come to the quadratic cone S_c.

2.2 The case when all $r_k \not\in \mathbb{F}_q$

Suppose that $r_k^q = r_{k+1}$, where the index k is taken modulo 3. Let's glue the curves E, E^T along their 2-torsion subgroups in the following way:

$$\psi\colon E[2] \to E^T[2] \qquad (r_k, 0) \mapsto (vr_{k+1}, 0).$$

The map ψ respects the Frobenius action on $E[2]$ and $E^T[2]$. Furthermore, note that $\psi \neq \theta|_{E[2]}$. For the sake of compactness, it is worth introducing the new $\mathbb{F}_q$-values

$$num_y := (r_0 - r_1)(r_0 - r_2)(r_1 - r_2), \qquad R := r_0r_1^2 + r_1r_2^2 + r_2r_0^2 + 3a_6,$$
$$den_x^T := a_2^2 - 3a_4, \qquad R^T := r_0r_2^2 + r_1r_0^2 + r_2r_1^2 + 3a_6.$$

Owing to [22, Section 3.2], there are two quadratic $\mathbb{F}_q$-covers

$$\begin{aligned} \varphi\colon H \to E \qquad & (t,s) \mapsto \left(\frac{vR{\cdot}t^2 - num_y}{v{\cdot}den_x^T{\cdot}t^2},\ \frac{num_y}{(den_x^T)^2{\cdot}t^3}{\cdot}s \right), \\ \varphi^T\colon H \to E^T \qquad & (t,s) \mapsto \left(\frac{v{\cdot}num_y{\cdot}t^2 + R^T}{den_x^T},\ \frac{v{\cdot}num_y}{(den_x^T)^2}{\cdot}s \right) \end{aligned} \tag{5}$$

from a genus 2 curve $H : s^2 = h(t)$. Here, $h(t)$ has the same shape as the polynomial (3) except that

$$c := num_y{\cdot}den_x^T, \qquad \delta_k := \frac{r_{k-1} - r_k}{v(r_k - r_{k+1})}, \tag{6}$$

and $b_6 = -1/v^3$. In addition, put $\widehat{h}(t) := h(t)/c$. Due to the formula (1), the covers φ, φ^T are correctly defined (equivalently, $c \neq 0$) if and only if $j(E) \neq 0$ as assumed. Keep in mind that $\varphi(0, \sqrt{cb_6}) = \infty$ and besides $\sqrt{cb_6} \in \mathbb{F}_q \Leftrightarrow \sqrt{c} \not\in \mathbb{F}_q$.

It is readily seen that $\delta_k^q = \delta_{k+1} \not\in \mathbb{F}_q$. In turn, $\gamma_k := \sqrt{\delta_k} \not\in \mathbb{F}_{q^3}$, because the norm $\mathrm{N}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\delta_k) = -b_6$ is a quadratic non-residue in $\mathbb{F}_q$. Consequently, the polynomial $h(t)$ is $\mathbb{F}_q$-irreducible. Without loss of generality, let $\gamma_0^q = \gamma_1$, $\gamma_1^q = \gamma_2$, and $\gamma_2^q = -\gamma_0$. The components of the degenerate fibers on the surface S_h constitute two Frobenius orbits, namely

$$\Big\{ L_+\big((-1)^k\gamma_k\big),\ L_-\big((-1)^{k+1}\gamma_k\big) \Big\}_{k=0}^2, \qquad \Big\{ L_+\big((-1)^{k+1}\gamma_k\big),\ L_-\big((-1)^k\gamma_k\big) \Big\}_{k=0}^2.$$

We can contract over $\mathbb{F}_q$ any of them, obtaining the quadratic cone S_c as before. As a consequence, the composition

$$bl_{\widehat{h},\pm} := bl_{Q_0,\pm} \circ bl_{Q_1,\mp} \circ bl_{Q_2,\pm} \colon \quad S_c \to S_h$$

is defined over $\mathbb{F}_q$ (unlike $bl_{Q_k,\pm}$). More precisely,

$$bl_{\widehat{h},\pm} \colon S_c \to S_h \qquad (x,y) \mapsto \Big(\rho(t)\cdot x \pm \sqrt{v}\cdot\varrho(t)\cdot y,\ \pm\frac{\varrho(t)}{\sqrt{v}}\cdot x + \rho(t)\cdot y\Big),$$

where

$$\rho(t) := t^3 + (-\gamma_0\gamma_1 + \gamma_0\gamma_2 - \gamma_1\gamma_2)t, \qquad \varrho(t) := (\gamma_0 - \gamma_1 + \gamma_2)t^2 - \gamma_0\gamma_1\gamma_2.$$

By default, put $bl_{\widehat{h}} := bl_{\widehat{h},+}$.

3 New hash function

Let's stick to the symbolism of Section 2. In it we established the following theorem.

Theorem 1. *Take the polynomial $h(t)$ of the form (3) with the values (6) or (4) except for the case $r_0 = 0$, $r_1, r_2 \in \mathbb{F}_q$, but $\sqrt{a_4} \not\in \mathbb{F}_q$. Then, there is a birational $\mathbb{F}_q$-parametrization $\pi\colon \mathbb{A}^2_{(u,t)} \to S_h$ of the generalized Châtelet surface S_h. Moreover, π is well defined on the whole set $\mathbb{F}_q^2$.*

To be more precise,

$$\pi = \begin{cases} bl_{Q_0} \circ bl_{Q_1} \circ bl_{Q_2} \circ pr_{P_0}^{-1} & \text{if} \quad r_0 = 0 \text{ and } r_1, r_2, \sqrt{a_4} \in \mathbb{F}_q, \\ bl_{Q_0} \circ bl_{Q_+} \circ bl_{Q_-} \circ pr_{P_0}^{-1} & \text{if} \quad r_0 \in \mathbb{F}_q, \text{ but } r_1, r_2 \not\in \mathbb{F}_q, \\ bl_{\widehat{h}} \circ pr_{P_0}^{-1} & \text{if} \quad r_k \not\in \mathbb{F}_q. \end{cases}$$

The exceptional case of the theorem is treated by means of Lemma 2, hence it is excluded from our discussion. For uniformity of notation, $S := S_h$ henceforth. In fact, the restriction of the map π to the line $u = t$ gives rise to an $\mathbb{F}_q$-section $\sigma\colon \mathbb{A}^1_t \to S$ of the conic bundle $pr_t\colon S \to \mathbb{A}^1_t$ or, alternatively, to an $\mathbb{F}_q(t)$-point of S as a conic. To further simplify the formulas of π it is reasonable to actually put $u = t$ as it is originally done for Skałba's map [43].

Denote by $H^T\colon vs^2 = h(t)$ the hyperelliptic quadratic twist of H. Any cover $\varphi^T : H \to E^T$ is clearly can be interpreted (up to an $\mathbb{F}_q$-isomorphism) as the cover $\varphi^T : H^T \to E$. Since the curves E, E^T possess opposite traces and H is obtained by gluing them, H (and hence H^T) is a curve of zero trace, that is,

$$\#H(\mathbb{F}_q) = \#H^T(\mathbb{F}_q) = q + 1.$$

The polynomial $h(t)$ with the values (4) fits [31, Section 5], because $d := b_4/b_2 = 1/v$ is a quadratic non-residue and the coefficient $b_6 = d^3$. Therefore, we enjoy bijective maps $\mathbb{P}^1(\mathbb{F}_q) \to H(\mathbb{F}_q)$ and $\mathbb{P}^1(\mathbb{F}_q) \to H^T(\mathbb{F}_q)$ extracting a square root in $\mathbb{F}_q$. These maps are based on a non-hyperelliptic involution of H, H^T defined over $\mathbb{F}_{q^2}$, but not over $\mathbb{F}_q$. It is not hard to prove that the geometric automorphism group $\mathrm{Aut}(\overline{H})$ of the general H is isomorphic to the dihedral group D_8 of order 8. Interestingly, the hash function from [32] is built in a similar way on other genus 2 curves H having $\mathrm{Aut}(\overline{H}) \simeq \mathrm{D}_{12}$. By the way, there are no other dihedral groups $\mathrm{Aut}(\overline{H})$ for genus 2 curves (see details in [8,9]).

The facts of the previous paragraph are wrong if we talk about the values (6). The point is that the (geometric) automorphism group of the general H is just isomorphic to $(\mathbb{Z}/2)^2$. Roughly speaking, the curve H is not sufficiently "symmetric". That is why instead of mapping separately to $H(\mathbb{F}_q)$ and $H^T(\mathbb{F}_q)$ it is suggested to map onto $U(\mathbb{F}_q)$ from two copies of $\mathbb{P}^1(\mathbb{F}_q)$, where

$$U := H \sqcup H^T \quad \subset \quad \mathbb{A}^2_{(t,s)} \times \{0,1\}$$

for compactness. We purposely introduce the disjoint union, because the curves H, H^T intersect at the points $(\pm\gamma_k, 0)$. Unless stated otherwise, the subsequent exposition is carried out for the both suites (4), (6).

Given $x, y \in \mathbb{F}_q$ such that $x^2 - vy^2 = z^2$ for $z \in \mathbb{F}_q$, denote by $M(x,y)$ Müller's algorithm returning z by using the values x, y. It should be noted that $x = y = 0$ is the only possible situation for $z = 0$. Consider the twisted surface $S^T : v(x^2 - vy^2) = h(t)$. In contrast to the twisted curves E^T, H^T, there is the $\mathbb{F}_q$-isomorphism

$$\iota\colon S \to S^T \qquad (x,y) \mapsto \Big(iy, \frac{ix}{v}\Big),$$

where $i := \sqrt{-1} \in \mathbb{F}_q$.

Eventually, we get the map

$$\tau\colon S(\mathbb{F}_q) \times \{0,1\} \to U(\mathbb{F}_q)$$

$$\tau(x,y,t,b) := \begin{cases} (t, 0, b) & \text{if} \quad \big(\frac{h(t)}{q}\big) = 0, \\ \Big(t, (-1)^b M(x,y), 0\Big) & \text{if} \quad \big(\frac{h(t)}{q}\big) = 1, \\ \Big(t, (-1)^b M\big(\iota(x,y)\big), 1\Big) & \text{if} \quad \big(\frac{h(t)}{q}\big) = -1. \end{cases}$$

Here, $\big(\frac{\cdot}{q}\big)$ is nothing but the Legendre symbol in $\mathbb{F}_q$ supplemented by the equality $\big(\frac{0}{q}\big) = 0$. It is worth emphasizing that it (as well as the inversion in $\mathbb{F}_q^*$) can be implemented in fast constant time in compliance with [5], [11, Section 2.1]. On the same subject, an implementer must call $M(0,0)$ in the first case to achieve a deterministic execution of τ for all input arguments.

We also lack the auxiliary map

$$\Phi\colon U(\mathbb{F}_q) \to E(\mathbb{F}_q) \qquad \Phi(P,b) := \begin{cases} \varphi(P) & \text{if} \quad b = 0, \\ \varphi^T(P) & \text{if} \quad b = 1. \end{cases}$$

Lastly, we obtain the map

$$e := \varPhi \circ \tau \circ \sigma_{\mathrm{id}} \colon \quad \mathbb{F}_q \times \{0,1\} \to E(\mathbb{F}_q),$$

where $\sigma_{\mathrm{id}} := \sigma \times \mathrm{id}$. It can be extended to $\mathbb{P}^1(\mathbb{F}_q) \times \{0,1\}$ by tinkering with the $\mathbb{F}_q$-points of $\mathbb{P}^1$, S, H, H^T, and E at infinity. Nonetheless, this is unnecessary in practice.

In cryptographic language, we also have the hash function $\mathcal{H} := e \circ \eta$, picking any one $\eta\colon \{0,1\}^* \to \mathbb{F}_q \times \{0,1\}$. For convenience, a step-by-step description of $\mathcal{H}$ containing all its components is represented in Algorithm 1. Among other things, the dummy multiplications of x, y by 0, 1 are included in the algorithm to respect its constant-timeness.

Example 1. As far as the author knows, the 2-adicity $\nu = 96$ is maximal among the basic fields of standardized elliptic curves (around the world). It is attained by the curve NIST P-224 from the standard [10, Section 3.2.1.2] recently updated. As the name indicates, the curve is defined over a field $\mathbb{F}_q$ of length $\lceil \log_2(q) \rceil = 224$. Its order $q \equiv 1 \pmod 3$, hence Icart's hash function $\mathcal{H}_I$ is not applicable to the curve as opposed to $\mathcal{H}$ and $\mathcal{H}_{old}$ from the former work [32]. Before $\mathcal{H}_{old}$, the so-called *simplified SWU hash function* $\mathcal{H}_{sSWU}$ (see, e.g., [49, Section 4.1]) was the best for NIST P-224.

Recall that $\mathcal{H}_{sSWU}$ extracts a square root in $\mathbb{F}_q$ as well as $\mathcal{H}$. The constant-time implementation of [38, Algorithm 4] has a running time close to $\approx \log_2(q) + \nu(2\lceil \log_2(\nu) \rceil - 1)$ finite field multiplications. This amounts to $\approx 224 + 96(2 \cdot 7 - 1) = 1472$ ones in the field $\mathbb{F}_q$ under consideration. In turn, Müller's algorithm performs $\approx 2 \cdot 224 = 448$ ones (see a further optimization in the patent [34]). Finally, $\mathcal{H}_{old}$ has to compute ≈ 865 field multiplications in accordance with [32, Table 1]. To sum up, the new hash function $\mathcal{H}$ carries out ≈ 417 (respectively, ≈ 1024) fewer multiplications than $\mathcal{H}_{old}$ (respectively, $\mathcal{H}_{sSWU}$). In other terms, there is an acceleration of about 2 and 3.25 times, respectively.

The reader may wonder what is the approximate number of multiplications in Sutherland's square root algorithm. At the moment, it is difficult to give an answer to this question, because there is no article yet (similar in style to [38]) that likewise analyzes all the implementation details. Sutherland does not care about constant-timeness, hence the average estimates from [44, Tables 1, 2] cannot be taken as a basis for comparison with Müller's algorithm. On the other hand, investigating practicality of Sutherland's approach goes beyond the scope of the present article.

Example 2. There may be an objection that the curve NIST P-224 is obsolete and is maintained in the standard only for the sake of compatibility with old software. In this connection, the relevance of this paper may seem overpriced at first glance. This is not so, since there are at least two relatively new elliptic curves dubbed *stark curve* [1] and *starkjub* [2]. They are defined over the same 252-bit field $\mathbb{F}_q$ with $\nu = 192 \gg 96$. For such finite fields, the supremacy of Müller's algorithm compared to other square root algorithms should be even clearer.

Algorithm 1: The new hash function $\mathcal{H}\colon \{0,1\}^* \to E(\mathbb{F}_q)$

```
Data: a finite field $\mathbb{F}_q$ of characteristic $p > 3$ such that $i = \sqrt{-1} \in \mathbb{F}_q$,
    an elliptic $\mathbb{F}_q$-curve $E$ of $j$-invariant $\neq 0, 1728$,
    a hash function $\eta\colon \{0,1\}^* \to \mathbb{F}_q \times \{0,1\}$ and a string $str \in \{0,1\}^*$,
    the genus two $\mathbb{F}_q$-curves $H: s^2 = h(t)$ and $H^T: vs^2 = h(t)$, where $\sqrt{v} \not\in \mathbb{F}_q$,
    the $\mathbb{F}_q$-covers $\varphi\colon H \to E$ and $\varphi^T: H^T \to E$,
    the surface $S\colon x^2 - vy^2 = h(t)$ and the $\mathbb{F}_q$-section $\sigma\colon \mathbb{A}^1_t \to S$,
    Müller's algorithm $M$;
Result: the point $\mathcal{H}(str) \in E(\mathbb{F}_q)$;
begin
    $(t, b) := \eta(str)$;
    $(x, y, t) := \sigma(t)$;
    $L := \left(\frac{h(t)}{q}\right)$;
    if $L = 0$ then
        $X := 0 \cdot x$;
        $Y := 0 \cdot y$;
        $b' := b$;
    end
    else if $L = 1$ then
        $X := 1 \cdot x$;
        $Y := 1 \cdot y$;
        $b' := 0$;
    end
    else
        $X := i \cdot y$;
        $Y := (i/v) \cdot x$;
        $b' := 1$;
    end
    $s := (-1)^b M(X, Y)$;
    $b := b'$;
    if $b = 0$ then
        return $\varphi(t, s)$.
    end
    else
        return $\varphi^T(t, s)$.
    end
end
```

It is important to realize that the magnitude of ν for the the mentioned curves is not accidental. The point is that in the case of the high 2-adicity of $\mathbb{F}_q$ one can apply the fast Fourier transform (FFT) to speed up the arithmetic of $\mathbb{F}_q$-polynomials. This becomes in demand more and more in advanced protocols of elliptic curve cryptography. A confirmation of the given words can be seen in the up-to-date survey [4]. It also contains a lot of modern real-world curves, but frankly speaking, $\nu < 96$ concretely for those curves.

To be honest, $\mathcal{H}$ is not the best hash function for the stark(jub) curves, because $q \equiv 2 \pmod 3$ for them and thereby Icart's hash function (extracting $\sqrt[3]{\cdot} \in \mathbb{F}_q$) is relevant. Hence, it makes little sense to analyze (as in the previous example) the multiplication counts for $\mathcal{H}$, $\mathcal{H}_{old}$ (if applicable), and $\mathcal{H}_{sSWU}$. However, it is quite possible that alternative *STARK-friendly curves* will emerge in the near future, for which $\nu \gtrsim 192$ and conversely $q \equiv 1 \pmod 3$. Any redundant condition (including on the reminder of q modulo 3) no doubt leads to complication of curve generation. According to [39], producing truly transparent curves (even without looking at ν) is not as simple as may appear at first sight.

3.1 Indifferentiability from a random oracle

In this section we will encounter some statistical notions, which are common in the current research area. They can be found, e.g., in [7, Sections 2, 3].

Lemma 3. *For the covers (2) and any affine point $P = (x, y) \in E(\mathbb{F}_q)$ there is the criterion*

$$\varphi^{-1}(P) \cap H(\mathbb{F}_q) = \emptyset \qquad \Leftrightarrow \qquad (\varphi^T)^{-1}(P) \cap H^T(\mathbb{F}_q) = \emptyset.$$

The lemma immediately follows from the simple equality

$$(pr_x \circ \varphi^T)(t) = (pr_x \circ \varphi)\Big(\frac{1}{vt}\Big).$$

For the other pair of covers (5) the given lemma is false. In particular, the situation $\Phi^{-1}(P) \cap U(\mathbb{F}_q) = \emptyset$ occurs quite often. A counterexample can be easily found by sampling randomly the appropriate parameters q, r_k, and P.

We see that the map Φ (and hence e) is itself far from surjective. This implies non-*regularity* of the maps Φ, e. By this reason, we are forced to resort to the tensor squares

$$\Phi^{\otimes 2} := [+] \circ \Phi^{\times 2}\colon \quad U^2(\mathbb{F}_q) \to E(\mathbb{F}_q),$$
$$e^{\otimes 2} := \Phi^{\otimes 2} \circ \tau^{\times 2} \circ \sigma_{\mathrm{id}}^{\times 2}\colon \quad \mathbb{F}_q^2 \times \{0,1\}^2 \to E(\mathbb{F}_q),$$

where

$$[+]\colon E^2 \to E \qquad (P, P') \mapsto P + P'.$$

Despite the fact that the original map π acts from the whole plane $\mathbb{A}^2_{(u,t)}$, we cannot benefit from this circumstance. We conclude that restricting π to the diagonal $u = t$ is actually justified. Otherwise, the output length (and hence the running time) of the auxiliary hash function η would be doubled without any advantage. For the provably secure η this would significantly slow down a cryptosystem. In comparison, certain maps from $\mathbb{F}_q^2$ in the recent works [11,29,30] lead to indifferentiable hash functions requiring only one root extraction.

Theorem 2. *The map $\varPhi^{\otimes 2}$ is admissible.*

Proof. We lack the quantities

$$\mathrm{WS}(\phi, \chi) := \sum_{P \in \mathbb{S}} (\chi \circ \phi)(P), \qquad \varDelta(\phi) := \sum_{P \in E(\mathbb{F}_q)} \left| \frac{\#\phi^{-1}(P)}{\#\mathbb{S}} - \frac{1}{\#E(\mathbb{F}_q)} \right|,$$

where $\chi\colon E(\mathbb{F}_q) \to \mathbb{C}^*$ is a complex character and $\phi\colon \mathbb{S} \to E(\mathbb{F}_q)$ is any map from a finite set $\mathbb{S}$. The first quantity is an analogue of *Weil sum* [36, Section 5.4]. The second is the *statistical distance* between the uniform distribution on $E(\mathbb{F}_q)$ and that induced by ϕ (provided that the distribution on $\mathbb{S}$ is also uniform).

Due to [16, Theorem 7], the cover φ is 2-*well-distributed*, i.e., $|\mathrm{WS}(\varphi, \chi)| \leqslant 2\sqrt{q}$ for every non-trivial character χ. Besides, since φ is a quadratic cover, $\varphi^{-1}(P)$ contains at most two $\mathbb{F}_q$-points for each $P \in E(\mathbb{F}_q)$. The same properties are true for φ^T. We have the right to suppose (for simplicity) the near-equality $\#E(\mathbb{F}_q) \approx q$. So, in accordance with [47, Corollary 1], the tensor products

$$\varphi^{\otimes 2}, \qquad (\varphi^T)^{\otimes 2}, \qquad \varphi \otimes \varphi^T, \qquad \varphi^T \otimes \varphi$$

are ϵ-regular, where the value $\epsilon \approx 2\sqrt{2/q}$ is negligible.

Note that

$$U^2 \ = \ H^2 \ \sqcup \ (H^T)^2 \ \sqcup \ H \times H^T \ \sqcup \ H^T \times H$$

and thereby $\#U^2(\mathbb{F}_q) = 4(q+1)^2$. It is readily seen that

$$\varDelta(\varPhi^{\otimes 2}) \ \leqslant \ \frac{\varDelta(\varphi^{\otimes 2}) + \varDelta((\varphi^T)^{\otimes 2}) + \varDelta(\varphi \otimes \varphi^T) + \varDelta(\varphi^T \otimes \varphi)}{4} \ \leqslant \ \epsilon.$$

By definition, the map $\varPhi^{\otimes 2}$ is also ϵ-regular. Formally speaking, we established regularity when the domain of $\varPhi^{\otimes 2}$ includes all pairs of $\mathbb{F}_q$-points on H, H^T (together with two bits) such that at least one of them lies at infinity. Nonetheless, restricting to $U^2(\mathbb{F}_q) \subset \mathbb{F}_q^4 \times \{0,1\}^2$ remains regular, because we discard a negligible number of points, viz. $O(q)$, with respect to $4(q+1)^2$.

Further, the map $\varPhi^{\otimes 2}$ is computable in constant time as the "basic" maps φ, φ^T are of the same degree (two) and have similar formulas. That is why evaluating them can be easily implemented without time difference. Lastly, their pairwise tensor products are *samplable* according to [47, Algorithm 1]. This entails samplability of $\varPhi^{\otimes 2}$, because nothing prevents to choose uniformly at random the pairs of φ, φ^T. Eventually, all the admissibility characteristics are proved. □

Let $\Sigma \subset S$ stand for the image of the section σ. The restriction $\tau\colon \Sigma(\mathbb{F}_q) \times \{0,1\} \to U(\mathbb{F}_q)$ is bijective. Indeed, it is effortlessly checked that the inverse map to τ has the form

$$\tau^{-1}\colon U(\mathbb{F}_q) \to \Sigma(\mathbb{F}_q) \times \{0,1\}$$

$$\tau^{-1}(t,s,b) = \begin{cases} (0,0,t,b) & \text{if} \quad s = 0, \\ (x,y,t,0) & \text{if} \quad s = M\big(\iota^b(x,y)\big) \neq 0, \\ (x,y,t,1) & \text{if} \quad s = -M\big(\iota^b(x,y)\big) \neq 0, \end{cases}$$

where $(x,y,t) = \sigma(t)$ and $\iota^0 := \mathrm{id}$.

In general, the composition operation leads beyond the class of admissible maps as said in [7, Appendix C.1]. However, the bijective maps τ, σ_{id} (and hence $\tau^{\times 2}$, $\sigma_{\mathrm{id}}^{\times 2}$) admit a deterministic evaluation along with their inverses. Consequently, we arrive at the next statement.

Theorem 3. *The map $e^{\otimes 2}$ is admissible.*

Corollary 2. *Whenever $\eta^{\times 2}\colon \{0,1\}^* \to \mathbb{F}_q^2 \times \{0,1\}^2$ is an indifferentiable hash function, so is the composition $\mathcal{H}^{\otimes 2} := e^{\otimes 2} \circ \eta^{\times 2}\colon \{0,1\}^* \to E(\mathbb{F}_q)$.*

The output length of $\eta^{\times 2}$ is only two bits longer than $2\lceil \log_2(q) \rceil$, hence the executing time of $\eta^{\times 2}$ is (almost) identical to that of hash functions $\{0,1\}^* \to \mathbb{F}_q^2$ of a more classical kind. At the same time, the operating time for $\mathcal{H}^{\otimes 2}$ is obviously two times greater than for $\mathcal{H}$ from Section 3, that is, Müller's algorithm is executed twice. Likewise, the hash functions $\mathcal{H}_{old}$, $\mathcal{H}_{sSWU}$ from Example 1 are not indifferentiable themselves as opposed to $\mathcal{H}_{old}^{\otimes 2}$, $\mathcal{H}_{sSWU}^{\otimes 2}$. Therefore, all the multiplication counts of that example should be doubled in the random oracle setting.

4 Conclusion

The hashing approach of the present article can be extended to elliptic $\mathbb{F}_q$-curves of j-invariants 0, 1728 whose Frobenius trace has a small divisor. To this end, one should study the generalized Châtelet surfaces S_h with polynomials $h(t)$ (of degrees 5, 6) written out in [31, Sections 3, 4]. The only potential obstacle on the path might be non-unirationality of S_h over the field $\mathbb{F}_q$. Fortunately, [25, Section 8] concludes that any conic $\mathbb{F}_q$-bundle with $\leqslant 6$ degenerate fibers (in particular, S_h) is actually $\mathbb{F}_q$-unirational provided that the surface has a smooth $\mathbb{F}_q$-point. And this condition automatically holds over finite fields of cryptographic sizes.

Meanwhile, (most) modern elliptic curves of j-invariants 0, 1728 over highly 2-adic fields are initially equipped with an $\mathbb{F}_q$-isogeny χ of small degree to another elliptic curve. The SNARK-friendly $j = 0$ curves from the web pages [20,21] can serve as a confirmation of the given words. Therefore, indirect hashing via χ takes place. Let's repeat again that the hash function $\mathcal{H}_{old}$ is relevant only if $\deg(\chi) = 3$. So, $\mathcal{H}_{old}$ does not cover any curve of $j = 0, 1728$ that $\mathcal{H}$ could not cover indirectly. At this stage of development of elliptic curve cryptography we hereby handled (almost) all real-world elliptic curves over fields of large 2-adicity ν.

Finally, it is worth stressing one more time that there are even more efficient hash functions $\mathcal{H}_I$, $\mathcal{H}_3$, and $\mathcal{H}_7$ represented in Table 1 (in addition to $\mathcal{H}$). Their bottleneck consists in finding a radical $\sqrt[\ell]{\cdot} \in \mathbb{F}_q$ of odd degree $\ell \in \mathbb{N}$. For most fields $\mathbb{F}_q$ this is nothing but one exponentiation in $\mathbb{F}_q$ requiring $n \in \mathbb{N}$ field multiplications, where $\log_2(q) \lesssim n \lesssim 2\log_2(q)$. Nevertheless, as opposed to $\mathcal{H}$, the other table hash functions suffer from specific limitations on E and $\mathbb{F}_q$. Surprisingly, $\mathcal{H}_3$ behaves as a random oracle unlike $\mathcal{H}_I$, $\mathcal{H}_7$, though the tensor squares $\mathcal{H}_I^{\otimes 2}$, $\mathcal{H}_7^{\otimes 2}$ as usual become random oracles (cf. Corollary 2).

Hash function	Year	Author	Reference	Bottleneck	Conditions	Is indiff.?
$\mathcal{H}_I$	2009	Icart	[23]	$\sqrt[3]{\cdot}$	$q \equiv 2 \pmod 3$	no
$\mathcal{H}_3$	2022	K.	[29]		$q \equiv 1 \pmod 3$, $a = 0$, $\sqrt{b} \in \mathbb{F}_q$	yes
$\mathcal{H}_7$	2023		[28, Section 2.2]	$\sqrt[7]{\cdot}$	$q \equiv 2, 4 \pmod 7$, j-invariant $-3^3 5^3$	no
$\mathcal{H}$			Section 3	$\sqrt{\cdot} = M(\cdot, \cdot)$	$ab \neq 0$	

Table 1. State-of-the-art hash functions to elliptic curves $E\colon y^2 = x^3 + ax + b$ over highly 2-adic fields $\mathbb{F}_q$

Acknowledgements. The author expresses his gratitude to Damien Stehlé for hiring him as a postdoc at École Normale Supérieure de Lyon.

References

1. stark curve, `https://docs.starkware.co/starkex/crypto/stark-curve.html`
2. starkjub (2023), `https://github.com/hashcloak/starkjub`
3. Andreescu, T., Andrica, D.: Quadratic diophantine equations, Developments in Mathematics, vol. 40. Springer, New York (2015)
4. Aranha, D.F., El Housni, Y., Guillevic, A.: A survey of elliptic curves for proof systems. Designs, Codes and Cryptography (2022), `https://link.springer.com/article/10.1007/s10623-022-01135-y`
5. Aranha, D.F., Salling Hvass, B., Spitters, B., Tibouchi, M.: Faster constant-time evaluation of the Kronecker symbol with application to elliptic curve hashing (2023), `https://eprint.iacr.org/2023/1261`
6. Bottinelli, P.: Breaking Pedersen hashes in practice (2023), `https://research.nccgroup.com/2023/03/22/breaking-pedersen-hashes-in-practice`

7. Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M.: Efficient indifferentiable hashing into ordinary elliptic curves. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 237–254. Springer, Berlin, Heidelberg (2010)
8. Cardona, G.: $\mathbb{Q}$-curves and abelian varieties of GL_2-type from dihedral genus 2 curves. In: Cremona, J.E., Lario, J.C., Quer, J., Ribet, K.A. (eds.) Modular Curves and Abelian Varieties. Progress in Mathematics, vol. 224, pp. 45–52. Birkhäuser, Basel (2004)
9. Cardona, G., Quer, J.: Curves of genus 2 with group of automorphisms isomorphic to D_8 or D_{12}. Transactions of the American Mathematical Society **359**(6), 2831–2849 (2007)
10. 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`
11. Chávez-Saab, J., Rodríguez-Henríquez, F., Tibouchi, M.: SWIFTEC: Shallue-van de Woestijne indifferentiable function to elliptic curves. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology – ASIACRYPT 2022. Lecture Notes in Computer Science, vol. 13791, pp. 63–92. Springer, Cham (2022)
12. Châtelet, F.: Points rationnels sur certaines courbes et surfaces cubiques. L'Enseignement Mathématique **5**(3), 153–170 (1959)
13. Cipolla, M.: Un metodo per la risolutione della congruenza di secondo grado. Rendiconto dell'Accademia delle Scienze Fisiche e Matematiche **9**, 154–163 (1903)
14. Cremona, J., Rusin, D.: Efficient solution of rational conics. Mathematics of Computation **72**(243), 1417–1441 (2003)
15. El Mrabet, N., Joye, M. (eds.): Guide to pairing-based cryptography. Cryptography and Network Security Series, Chapman and Hall/CRC, New York (2017)
16. Farashahi, R.R., Fouque, P.A., Shparlinski, I.E., Tibouchi, M., Voloch, J.F.: Indifferentiable deterministic hashing to elliptic and hyperelliptic curves. Mathematics of Computation **82**(281), 491–512 (2013)
17. Faz-Hernandez, A., Scott, S., Sullivan, N., Wahby, R.S., Wood, C.A.: Hashing to elliptic curves (RFC 9380) (2023), `https://datatracker.ietf.org/doc/draft-irtf-cfrg-hash-to-curve`
18. Galbraith, S.D.: Mathematics of public key cryptography. Cambridge University Press, New York (2012)
19. Hartshorne, R.: Algebraic geometry, Graduate Texts in Mathematics, vol. 52. Springer, New York, 8 edn. (1997)
20. Hopwood, D.: The Pasta curves for Halo 2 and beyond (2020), `https://electriccoin.co/blog/the-pasta-curves-for-halo-2-and-beyond`
21. Hopwood, D.: Pluto/Eris supporting evidence (2021), `https://github.com/daira/pluto-eris`
22. Howe, E.W., Leprévost, F., Poonen, B.: Large torsion subgroups of split Jacobians of curves of genus two or three. Forum Mathematicum **12**(3), 315–364 (2000)
23. 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)
24. Kollár, J.: Unirationality of cubic hypersurfaces. Journal of the Institute of Mathematics of Jussieu **1**(3), 467–476 (2002)
25. Kollár, J., Mella, M.: Quadratic families of elliptic curves and unirationality of degree 1 conic bundles. American Journal of Mathematics **139**(4), 915–936 (2017)

26. Koshelev, D.: Batch point compression in the context of advanced pairing-based protocols (2021), `https://eprint.iacr.org/2021/1446`
27. Koshelev, D.: New point compression method for elliptic $\mathbb{F}_{q^2}$-curves of j-invariant 0. Finite Fields and Their Applications **69**, Article 101774 (2021)
28. Koshelev, D.: Some remarks on how to hash faster onto elliptic curves (2021), `https://eprint.iacr.org/2021/1082`
29. Koshelev, D.: 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)
30. Koshelev, D.: The most efficient indifferentiable hashing to elliptic curves of j-invariant 1728. Journal of Mathematical Cryptology **16**(1), 298–309 (2022)
31. Koshelev, D.: Optimal encodings to elliptic curves of j-invariants 0, 1728. SIAM Journal on Applied Algebra and Geometry **6**(4), 600–617 (2022)
32. Koshelev, D.: Hashing to elliptic curves over highly 2-adic fields $\mathbb{F}_q$ with $O(\log q)$ operations in $\mathbb{F}_q$ (2023), `https://eprint.iacr.org/2023/121`
33. Koshelev, D.: Magma code (2023), `https://github.com/dishport/Batching-Cipolla-Lehmer-Muller-square-root-algorithm-with-hashing-to-elliptic-curves`
34. 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
35. 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)
36. Lidl, R., Niederreiter, H.: Finite fields, Encyclopedia of Mathematics and its Applications, vol. 20. Cambridge University Press, Cambridge (1997)
37. Müller, S.: On the computation of square roots in finite fields. Designs, Codes and Cryptography **31**(3), 301–312 (2004)
38. Pornin, T.: Optimized discrete logarithm computation for faster square roots in finite fields (2023), `https://eprint.iacr.org/2023/828`
39. Sedlacek, V., Suchanek, V., Dufka, A., Sys, M., Matyas, V.: DiSSECT: Distinguisher of standard and simulated elliptic curves via traits. In: Batina, L., Daemen, J. (eds.) Progress in Cryptology – AFRICACRYPT 2022. Lecture Notes in Computer Science, vol. 13503, pp. 493–517. Springer, Cham (2022)
40. Shallue, A., van de Woestijne, C.E.: Construction of rational points on elliptic curves over finite fields. In: Hess, F., Pauli, S., Pohst, M. (eds.) Algorithmic Number Theory. ANTS 2006. Lecture Notes in Computer Science, vol. 4076, pp. 510–524. Springer, Berlin, Heidelberg (2006)
41. 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)
42. Shoup, V.: A computational introduction to number theory and algebra. Cambridge University Press, Cambridge, 2 edn. (2008)
43. Skałba, M.: Points on elliptic curves over finite fields. Acta Arithmetica **117**(3), 293–301 (2005)
44. Sutherland, A.V.: Structure computation and discrete logarithms in finite abelian p-groups. Mathematics of Computation **80**(273), 477–500 (2011)
45. Swinnerton-Dyer, P.: Rational points on some pencils of conics with 6 singular fibres. Annales de la Faculté des Sciences de Toulouse: Mathématiques (Série 6) **8**(2), 331–341 (1999)

46. Sze, T.W.: On taking square roots without quadratic nonresidues over finite fields. Mathematics of Computation **80**(275), 1797–1811 (2011)
47. Tibouchi, M., Kim, T.: Improved elliptic curve hashing and point representation. Designs, Codes and Cryptography **82**(1–2), 161–177 (2017)
48. 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)
49. Wahby, R.S., Boneh, D.: Fast and simple constant-time hashing to the BLS12-381 elliptic curve. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(4), 154–179 (2019)