How to hash onto $\mathbb{G}_2$ and not to hash onto $\mathbb{G}_1$ for pairing-friendly curves

Dmitrii Koshelev [1]

Computer sciences and networks department, Télécom Paris

Abstract. 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.

This short note 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 significantly exploit the presence of clearing 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 $a\colon \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r \subset \mathbb{F}_{q^k}^*$ 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. Throughout the text we mean hashing indifferentiable from a random oracle.

At the moment, the curve BLS12-381 (with $e = 2$) is the most popular in practice. Earlier for this curve (and a number of others) the author constructed encodings $\mathbb{F}_q^2 \to E_1(\mathbb{F}_q)$ and $\mathbb{F}_q \to E_2(\mathbb{F}_{q^2})$ computable in constant time of one exponentiation in $\mathbb{F}_q$. Combining the new ideas with these encodings, we obtain hash functions $\{0,1\}^* \to E_1(\mathbb{F}_q)$ and $\{0,1\}^* \to \mathbb{G}_2$, which seem to be difficult to speed up even more. We also discuss how much performance gain they provide over hash functions that are actively applied in the industry.

Key words: BLS12 family of pairing-friendly curves, clearing cofactor, indifferentiable hashing to elliptic curves, optimal ate pairings.

Introduction

This is an addendum to our recent articles [1], [2], [3]. So, with your permission, we do not provide a detailed introduction in order to avoid repetition. Good surveys on how to hash into (or onto) elliptic curves over finite fields are also represented in [4, §8], [5]. By the same reason, let us keep the notation of the abstract, clarifying some things.

Note that the condition $e \in \mathbb{N}$ is not automatically met, i.e., this is our assumption. It is claimed (e.g., in [4, 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 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 $E_1 \xrightarrow{\sim} E_2$. 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 $j(E_i) \neq 0, 1728$ (respectively, $d = 4$ iff $j(E_i) = 1728$ and $d = 6$ iff $j(E_i) = 0$).

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

[1] web page: https://www.researchgate.net/profile/Dimitri_Koshelev
email: dimitri.koshelev@gmail.com

$h_2 \colon S_2 \to E_2(\mathbb{F}_{q^e})$ 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 requires 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 rarely used 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 $\mathbb{G}_2$ can not be replaced by hashing to $\mathbb{G}_1$ in all situations.

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

As far as we know, (non-degenerate) *optimal ate pairings* $a \colon \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r \subset \mathbb{F}_{q^k}^*$ [4, Theorem 3.3.4] 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, §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 $a(Q, c_1'P) = a(Q,P)^{c_1'} = 1$. We stress that popular protocols (such as the Boneh–Franklin identity-based encryption [4, §1.6.4] or the aggregated BLS signature [9]) work correctly whether the order of P equals r or not. 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 [1] we construct an encoding $h_1 \colon \mathbb{F}_q^2 \to E_1(\mathbb{F}_q)$ for elliptic curves $E_1 \colon y^2 = x^3 + b$ (of j-invariant 0) provided that $\sqrt{b} \in \mathbb{F}_q$. Moreover, h_1 can be implemented in constant time of one raising to some power $n_1 \in \mathbb{N}$ in the field $\mathbb{F}_q$ (in addition to a few additions and multiplications). In particular, our encoding is applicable to the curve BLS12-381 for which $b = 4$ and $n_1 = (q-10)/27$. Due to [10, Table 1] this curve is a de facto standard in pairing-based cryptography.

More generally, the *Barreto–Lynn–Scott family* with $k = 12$ (see, e.g., [11, §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* [4, Example 4.2].

Recall that the famous *(indirect) Wahby–Boneh map* [12, §4] (based on the *simplified SWU* one) is 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, §5] twice applies the Wahby–Boneh encoding in order to act as a random oracle. By the way, the other indifferentiable hash function H_3 is even slower 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, §9.2.1] we obtain the inequalities

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

We can not claim that these upper bounds are mathematically correct, because we omitted $o(1)$ in 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, §9.1.3] (with $k = 5$), one can explicitly derive an addition chain for n_1 (resp. n_2) whose the length equals 449 (resp. 458). We invite the reader to independently check our conclusion, since the mentioned method is simple and has many public implementations. Curiously, a similar chain for n_2 of the same length 458, obtained by means of more advanced methods, appears in the optimized library [14]. Thus the Wahby–Boneh map applied twice is much slower than ours 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.

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

To our 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* [4, 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, unfortunately, for them the Miller loop is of a larger length than for (usual) optimal ate pairings. It is generally 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 next theorem we need the notions of (B-)*well-distributed* [15, Definitions 5, 7] and (ϵ-)*regular map* [15, Definition 3] (with respect to the uniform distribution on its domain).

Theorem 1. *Assume that exists a B-well-distributed encoding $h_2\colon \mathbb{F}_q \to E_2(\mathbb{F}_{q^e})$ (for $B \in \mathbb{R}_{>0}$) and a point of $E_2(\mathbb{F}_{q^e})$ of order $m \mid c_2$ (or, equivalently, $m \mid c_2'$). Then the map $[c_2'] \circ h_2\colon \mathbb{F}_q \to \mathbb{G}_2$ is ϵ-regular, where $\epsilon := B\sqrt{N_2/(mq)}$.*

Proof. Pick any point $P \in E_2(\mathbb{F}_{q^e})$ of order m. According to [15, Corollary 1] the encoding

$$F\colon \mathbb{F}_q \times [0, m) \to E_2(\mathbb{F}_{q^e}) \qquad (u, v) \mapsto h_2(u) + vP$$

is ϵ-regular for ϵ as in the statement of the theorem. It is readily checked that the composition $[c_2'] \circ F$ is still ϵ-regular. Since $[c_2'] \circ h_2 = [c_2'] \circ F$, the theorem is proved. □

For $e = 2$ an example of the desired encoding h_2 is given in [2] (modulo notation) as the composition $h_2 := \psi \circ \varphi \circ h$. Here $h\colon \mathbb{F}_q \to H(\mathbb{F}_q)$ is an encoding to some $\mathbb{F}_q$-curve H of geometric genus two, $\varphi\colon H \to E'$ is a (quadratic) $\mathbb{F}_{q^2}$-cover to an auxiliary elliptic $\mathbb{F}_{q^2}$-curve E' of j-invariant $\notin \mathbb{F}_q$, and finally $\psi\colon E' \to E_2$ is an $\mathbb{F}_{q^2}$-isogeny of small degree. By virtue of [3, Corollary 1], [2, Theorem 1] the encodings h and $\varphi \circ h$ are 2-well-distributed. The same is true for h_2 whenever $\psi\colon E'(\mathbb{F}_{q^2}) \xrightarrow{\sim} E_2(\mathbb{F}_{q^2})$, which follows if $\big(\deg(\psi), N_2\big) = 1$.

For the BLS12 family we have the parametrizations

$$c_2(z) = (z^8 - 4z^7 + 5z^6 - 4z^4 + 6z^3 - 4z^2 - 4z + 13)/9, \qquad c_2'(z) = 3(z^2-1)c_2(z)$$

according to [11, §4.1]. Recall that BLS12-381 has the form $E_2\colon y^2 = x^3 + 4(1+i)$ (where $i := \sqrt{-1} \notin \mathbb{F}_q$) and, as mentioned in [2, Introduction], there is the desired isogeny ψ of degree $7 \nmid N_2$. Besides, the group $E_2(\mathbb{F}_{q^2})$ possesses a point of order $m = c_2/(13 \cdot 23)$, because this number is square free. As a result, $\epsilon = 2\sqrt{13 \cdot 23 r/q} \leqslant 2^{-115/2}$ is a negligible value. Incidentally, this can not be said about BN curves, since for them $r/q = 1 + O(q^{-1/2})$.

It is worth noting that the encoding h can be computed in constant time of extracting one square root in $\mathbb{F}_q$. This is also true for h_2, since φ, ψ are algebraic maps of small degrees. Among other things, the denominators of their defining functions do not need to be inverted, because *Jacobian projective coordinates* (see, e.g., [12, §2]) are preferred for use in practice.

By analogy with Theorem 1, the map $[c_2'] \circ \mathrm{Map}_2$ (for Map_2 from [12, §5]) also turns out to be regular, that is the hash function H_4 from there actually acts as a random oracle. However this circumstance was not noticed in that article. In comparison to the Wahby–Boneh encoding, ours h_2 nevertheless allows to avoid extracting one square root in $\mathbb{F}_q$. The fact is that a square root in $\mathbb{F}_{q^2}$ (which appears in the simplified SWU map), as is well known, can be expressed via two square roots in $\mathbb{F}_q$. By the way, the other hash functions H_5, H_6 are even slower than H_4 by virtue of [12, Figure 1].

References

[1] Koshelev D., *Indifferentiable hashing to ordinary elliptic $\mathbb{F}_q$-curves of $j = 0$ with the cost of one exponentiation in $\mathbb{F}_q$*, https://eprint.iacr.org/2021/301, 2021.

[2] Koshelev D., *Faster indifferentiable hashing to elliptic $\mathbb{F}_{q^2}$-curves*, https://eprint.iacr.org/2021/678, 2021.

[3] Koshelev D., *Optimal encodings to elliptic curves of j-invariants 0, 1728*, https://eprint.iacr.org/2021/1034, 2021.

[4] El Mrabet N., Joye M., *Guide to Pairing-Based Cryptography*, Cryptography and Network Security Series, Chapman and Hall/CRC, New York, 2017.

[5] Faz-Hernandez A. et al., *Hashing to elliptic curves*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-hash-to-curve, 2021.

[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. et al., *BLS signatures*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-bls-signature, 2020.

[10] Sakemi Y. et al., *Pairing-friendly curves*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-pairing-friendly-curves, 2021.

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

[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, 154–179.

[13] Cohen H. et al., *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Discrete Mathematics and Its Applications, **34**, Chapman and Hall/CRC, New York, 2005.

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

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