Subgroup membership testing on elliptic curves via the Tate pairing

Dmitrii Koshelev [1]

Computer sciences and networks department, Télécom Paris

Abstract. This note explains how to guarantee the membership of a point in the prime order subgroup of an elliptic curve (over a finite field) satisfying some moderate conditions. For this purpose, we apply the Tate pairing on the curve, however it is not required to be pairing-friendly. Whenever the cofactor is small, the given approach is more efficient than other known ones, because it needs to compute at most two n-th power residue symbols (with small n) in the basic field. In particular, we deal with two Legendre symbols for the curve Bandersnatch proposed by the Ethereum Foundation team. Due to recent improvements of Euclidean type constant-time algorithms for the Legendre symbol computation, the new subgroup check is almost free for that curve.

Key words: non-prime order elliptic curves, power residue symbol, subgroup membership testing, Tate pairing.

1 Introduction

As is well known, elliptic curves of prime order are actively used in discrete logarithm cryptography. However twisted Edwards curves [1] (with a cofactor multiple of 4) equally became very popular, which is characterized by the inclusion of two of them in the draft NIST SP 800-186 [2]. Recall that twisted Edwards curves are birationally isomorphic to Montgomery ones (see, e.g., [3, Section 9.12.1]) and vice versa. By the way, the two curves in question are called Curve25519 [4] and Ed448-Goldilocks [5]. It is also worth noting twisted Hessian curves [6] whose the cofactor is a multiple of 3. Nevertheless, as far as we know, they have not been used in real-world cryptography.

To avoid timing attacks complete addition formulas are often utilized, because they are correctly defined for any pair of rational points on an elliptic curve (defined over a finite field). Such formulas exist even for curves of prime order in the Weierstrass form, but they are quite inefficient according to [7], [8]. In turn, the twisted Edwards form enjoys the fastest complete addition formulas among all known forms of elliptic curves.

To be protected against fault attacks from [9] [10] in any elliptic cryptography protocol, when receiving a point from a communication channel, it is necessary to make sure that it belongs to the appropriate elliptic curve. Fortunately, this can be easily done by substituting the coordinates of the point in the curve equation. However in the presence of a non-trivial cofactor even more work needs to be done. The seminal article [11] discusses subgroup security

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

ResearchGate: `https://www.researchgate.net/profile/dimitri-koshelev`

LinkedIn: `https://www.linkedin.com/in/dimitri-koshelev`

GitHub: `https://github.com/dishport`

for the multiplicative group of a finite field, although its arguments are also valid in the elliptic curve setting.

To thwart the subgroup attack it is often sufficient to just multiply an obtained point by the cofactor. Nevertheless, in a series of complicated protocols it is required to check whether a point really belongs to the subgroup of prime order. For example, as it turned out [12], multiplication by the cofactor in the signature scheme used in the Monero cryptocurrency and a number of others could lead to double-spending if any of the malicious users noticed this bug. In addition, multiplication by the cofactor forces protocol developers to complicate their security proofs even if they take place.

An obvious way to test membership in the prime order subgroup is to multiply a point by the order of that subgroup. Even if the curve enjoys an effectively computable endomorphism, which makes it possible to apply the GLV technique or its variations (see, e.g., [3, Section 11.3.3]), the mentioned test is still laborious. Another way is to determine whether the endomorphism of multiplication by the cofactor has a non-empty inverse image in the rational point group (cf. [3, Exercise 11.6.1]). Lemma 1 provides an elegant answer to this question by means of the Tate pairing [3, Section 26.3]. Of course, this lemma cannot be regarded as a new result.

Incidentally, another answer generally requires sequential finding roots of several polynomials over the basic field (see details in [13, Section 3], [14, Section 4]). It is well known that even a square root is at best expressed via exponentiation in the field. And for highly 2-adic fields one has to be content with the slower Tonelli–Shanks algorithm (represented, e.g., in [3, Section 2.9]). The latter situation arises for zk-SNARK friendly curves Jubjub [15] and Bandersnatch [16].

Among solutions to the subgroup attack problem, those called Decaf [17] and Ristretto [18] (for the cofactors 4 and 8 respectively) deserve special mention. We should be aware that their essence extends to other cofactors. The ones 4 and 8 are simply the most important in practice. In particular, there is the actual draft [19] about Decaf and Ristretto applied to the curves Ed448-Goldilocks and Curve25519 respectively.

Let us stick to the notation of Section 2. In a nutshell, the mentioned solutions find coset representatives (canonical in some sense) of the quotient group $E(\mathbb{F}_q)/E(\mathbb{F}_q)[e] \simeq G$. These representatives are not required to lie in the subgroup G, which is a more natural system of representatives. The Decaf-style approach undoubtedly provides protection, but it has a number of disadvantages. First, formally speaking, it does not answer the question of belonging to G. Second, at least for now, there is no general theory generalizing Decaf. Therefore determining canonical representatives may not be immediate for an arbitrary cofactor. Third, in our opinion, even the authentic Decaf (not to mention Ristretto) manipulates quite cumbersome formulas in comparison with the more laconic Tate pairing.

Finally, the sources [20], [21], [22] contain more information on subgroup membership checking in the case of pairing-friendly curves (see, e.g., [3, Section 26.6]). The author carefully analysed that for such curves the test of the given work does not surpass the state-of-the-art tests in performance. Looking ahead, the reason lies in huge cofactors, which occur for today's pairing groups (including $\mathbb{G}_1$). With the permission of the reader, details are omitted. Thus despite the fact that the Tate pairing underlies the new test, it is relevant only for non-pairing-friendly curves, although this property is not utilized anywhere by us.

2 New subgroup check

Consider an elliptic curve $E\colon y^2 = x^3 + a_2x^2 + a_4x + a_6$ (with the point $\mathcal{O} := (0:1:0)$ at infinity) over a finite field $\mathbb{F}_q$ of characteristic $p > 2$. We know [3, Theorem 9.8.1] that the rational point group $E(\mathbb{F}_q) \simeq \mathbb{Z}/n_0 \times \mathbb{Z}/n_1$, where $n_1 \mid n_0$. As is customary in discrete logarithm cryptography, there is a subgroup $G \subset E(\mathbb{F}_q)$ of large prime order r such that $r \mid\mid n_0$, but $r \nmid n_1$ (and hence $n_1 \mid e$, but $r \nmid e$, where $e := n_0/r$). In other words, $E(\mathbb{F}_q) = G \times E(\mathbb{F}_q)[e]$. So the order $N := \#E(\mathbb{F}_q) = n_0 n_1$ and the cofactor of G equals $N/r = en_1$. Throughout the text, we assume that $e \mid q-1$, but $p \nmid N$. For the sake of uniformity, put $e_0 := e$ and $e_1 := n_1$. Besides, let $E(\mathbb{F}_q)[e] = \langle P_0 \rangle \times \langle P_1 \rangle$, where $\mathrm{ord}(P_i) = e_i$.

Recall that for any $k \mid q-1$ the reduced Tate pairing [3, Section 26.3.2], [23, Equality (12)] can be represented in the form

$$t_k\colon E(\mathbb{F}_q)[k] \times E(\mathbb{F}_q)/kE(\mathbb{F}_q) \to \mu_k \qquad t_k(P,Q) := f_{k,P}(Q)^{(q-1)/k}, \tag{1}$$

where $\mu_k \subset \mathbb{F}_q^*$ is the group of all k-th roots of unity, $P \neq Q \neq \mathcal{O}$, and $f_{k,P} \in \mathbb{F}_q(E)$ is a Miller function satisfying the conditions

$$\mathrm{div}(f_{k,P}) = k(P) - k(\mathcal{O}), \qquad \Big(\Big(\frac{x}{y}\Big)^k \cdot f_{k,P}\Big)(\mathcal{O}) = 1.$$

The functions $f_{k,P}$ are recursively constructed, e.g., in [3, Section 26.3.1] by means of Miller's algorithm.

Note that the final exponentiation of the pairing t_k is nothing but the k-th power residue symbol $\left(\frac{\alpha}{q}\right)_k := \alpha^{(q-1)/k}$ when substituting $\alpha := f_{k,P}(Q)$. It is worth saying that we always can batch the inversion and symbol computation, since

$$\Big(\frac{\alpha_0/\alpha_1}{q}\Big)_k = \Big(\frac{\alpha_0\alpha_1^{k-1}}{q}\Big)_k$$

given $\alpha_i \in \mathbb{F}_q^*$. As said in [24], at least for $k \in \{2,3,4,5,7,8,11\}$ the symbol can be determined by efficient Euclidean type algorithms. In particular, for $k = 2$ we deal with the ordinary Legendre symbol and there are such algorithms executed in constant time by virtue of the sources [25], [26]. The important cases $k \in \{3,4,8\}$ are the subject of study in [27], [28], [29], and [30]. Finally, if k is not small, then the exponentiation is seemingly the best way to compute the power residue symbol.

For compactness of notation, we also define the homomorphisms

$$h_i\colon E(\mathbb{F}_q) \to \mu_{e_i} \qquad h_i(Q) := t_e(P_i, Q) = t_{e_i}(P_i, Q).$$

The last identity is from [3, Exercise 26.3.8]. For our purpose it is unnecessary to know the values $h_i(P_i)$, hence we can benefit from the form (1).

Lemma 1. *There are the equalities $G = eE(\mathbb{F}_q) = \ker(h_0) \cap \ker(h_1)$. In particular, $G = \ker(h_0)$ when $e_1 = 1$.*

Proof. Given a point $Q \in G$ we see that $Q = eR$ for $R := (e^{-1} \bmod r)Q$. The opposite inclusion $G \supset eE(\mathbb{F}_q)$ is even more trivial. Further, according to [3, Theorem 26.3.3] the Tate pairing is non-degenerate. Consequently, a point $Q \in E(\mathbb{F}_q)$ in fact belongs to $eE(\mathbb{F}_q)$ if and only if $t_e(P,Q) = 1$ for all $P \in E(\mathbb{F}_q)[e]$ or, equivalently, $h_0(Q) = h_1(Q) = 1$. Finally, it is readily seen that $f_{e,\mathcal{O}} = f_{1,\mathcal{O}} = 1$, so for $e_1 = 1$ the homomorphism h_1 is trivial. □

Lemma 1 gives rise to simple Algorithm 1.

Algorithm 1: New subgroup membership test

Data: a point $Q \in E(\mathbb{F}_q)$
Result: the answer to the question $Q \in G$?
begin
  compute the values $\beta_0 := h_0(Q)$ and $\beta_1 := h_1(Q)$ as described above;
  if $\beta_0 = \beta_1 = 1$ **then**
    return yes
  else
    return no
  end
end

Let's take a look at basic examples. For the first two of them we need simple facts from [31, Exercise 1.3.2].

The case $e_0 = 2$, $e_1 = 1$. Without lost of generality, $E\colon y^2 = x(x^2 + a_2x + a_4)$, where $a_2^2 - 4a_4,\ a_4 \notin (\mathbb{F}_q^*)^2$. The curves E are so-called double-odd curves thoroughly studied in [32], [33]. Clearly, $P_0 = (0,0)$ and $f_{2,P_0} = x$. Lemma 1 states that a point $(x,y) \in E(\mathbb{F}_q)$ lies in G if and only if $x \in (\mathbb{F}_q^*)^2$. We obtain the subgroup membership test invented in [32, Section 1.2].

The case $e_0 = e_1 = 2$. In this one (relevant for Bandersnatch), $E\colon y^2 = x(x-\alpha_1)(x-\alpha_2)$, where $\alpha_1, \alpha_2 \in \mathbb{F}_q^*$, but $\alpha_1\alpha_2 \notin (\mathbb{F}_q^*)^2$. Putting $\alpha_0 := 0$ in addition, we can pick, e.g., $P_i = (\alpha_i, 0)$. Therefore $f_{2,P_i} = x - \alpha_i$. Under the chosen parameters, Lemma 1 is exactly [31, Theorem 1.4.1], because $x - \alpha_2 \in (\mathbb{F}_q^*)^2$ automatically whenever $x - \alpha_i \in (\mathbb{F}_q^*)^2$ for $i \in \{0,1\}$. By the way, this result is used in [34, Section 3.2] to speed up compression of SIDH public keys.

The case $e_0 = 2^m$, $e_1 = 1$ for any $m \in \mathbb{N}$. The value $m = 3$ is relevant for Jubjub. As above, let $P_0 \in E(\mathbb{F}_q)$ be a point of order 2^m. For $0 \leqslant j \leqslant m$ we also introduce points $P_j = (x_j, y_j) := [2^j]P_0$ whose $\mathrm{ord}(P_j) = 2^{m-j}$ obviously. Be careful, there is a discrepancy with our previous notation P_1. Classical formulas of the double operation $P_{j+1} = [2]P_j$ on E (see, e.g., [3, Section 9.1]) are given as follows:

$$\lambda_j := \frac{3x_j^2 + 2a_2x_j + a_4}{2y_j}, \qquad x_{j+1} = \lambda_j^2 - 2x_j - a_2, \qquad y_{j+1} = \lambda_j(x_j - x_{j+1}) - y_j.$$

Moreover, we have [23, Equalities (1), (2), (14)]:

$$\ell_j := (y - y_j) - \lambda_j(x - x_j), \qquad \nu_j := x - x_j, \qquad \mu_j := \frac{\ell_j}{\nu_{j+1}},$$

$$f_{2^{j+1},P_0} = f_{2^j,P_0}^2 \cdot \mu_j, \qquad \text{and hence} \qquad f_{2^m,P_0} = \prod_{j=0}^{m-1} \mu_j^{2^{m-j-1}}.$$

<table>
<tr><th>Curve</th><th>$\lceil \log_2(q) \rceil$</th><th>e_0</th><th>e_1</th><th>$\nu_2(q-1)$</th></tr>
<tr><td>Curve25519 [4]</td><td>255</td><td>8</td><td rowspan="3">1</td><td>2</td></tr>
<tr><td>Ed448-Goldilocks [5]</td><td>448</td><td>4</td><td>1</td></tr>
<tr><td>Jubjub [15]</td><td rowspan="2">255</td><td>8</td><td rowspan="2">32</td></tr>
<tr><td>Bandersnatch [16]</td><td>2</td><td>2</td></tr>
</table>

Table 1: Some popular elliptic curves of non-prime order

Table 1 contains a series of elliptic curves utilized in real-world cryptography as well as some numerical values of interest to us. Among other things, $\nu_2(k)$ (for any $k \in \mathbb{N}$) is the natural number such that $2^{\nu_2(k)} \mid\mid k$. As we see, the first two curves are unfortunately not appropriate for the new subgroup check. They are added to the table just to be honest with the reader about non-universality of the current work. Incidentally, the last two curves are over the same field $\mathbb{F}_q$. Besides, in contrast to the other three curves, Bandersnatch is an incomplete twisted Edwards curve (see details in [35, Section 2]). Nevertheless, at points of its subgroup G the addition formulas are always defined. So for the mentioned curve the subgroup membership problem is even more relevant. In turn, the webpage [36] is dedicated to the given problem for Jubjub.

Acknowledgements. The author is grateful to Gottfried Gerold for a fruitful discussion on the importance of subgroup membership testing for the curve Bandersnatch.

References

[1] Bernstein D. J., Birkner P., Joye M., Lange T., Peters C., "Twisted Edwards curves", Progress in Cryptology – AFRICACRYPT 2008, LNCS, **5023**, ed. Vaudenay S., Springer, Berlin, Heidelberg, 2008, 389–405.

[2] Chen L., Moody D., Regenscheid A., Randall K., *Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters (Draft NIST special publication 800-186)*, https://csrc.nist.gov/publications/detail/sp/800-186/draft, 2019.

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

[4] Bernstein D. J., "Curve25519: New Diffie–Hellman speed records", Public Key Cryptography – PKC 2006, LNCS, **3958**, eds. Yung M., Dodis Y., Kiayias A., Malkin T., Springer, Berlin, Heidelberg, 2006, 207–228.

[5] Hamburg M., *Ed448-Goldilocks, a new elliptic curve*, https://eprint.iacr.org/2015/625, 2015.

[6] Bernstein D. J., Chuengsatiansup C., Kohel D., Lange T., "Twisted Hessian curves", Progress in Cryptology – LATINCRYPT 2015, LNCS, **9230**, eds. Lauter K., Rodríguez-Henríquez F., Springer, Cham, 2015, 269–294.

[7] Renes J., Costello C., Batina L., "Complete addition formulas for prime order elliptic curves", Advances in Cryptology – EUROCRYPT 2016, LNCS, **9665**, eds. Fischlin M., Coron J.-S., Springer, Berlin, Heidelberg, 2016, 403–428.

[8] Schwabe P., Sprenkels D. The complete cost of cofactor $h = 1$, Progress in Cryptology – INDOCRYPT 2019, LNCS, **11898**, eds. Hao F., Ruj S., Sen Gupta S., Springer, Cham, 2019, 375–397.

[9] Biehl I., Meyer B., Müller V., "Differential fault attacks on elliptic curve cryptosystems", Advances in Cryptology – CRYPTO 2000, LNCS, **1880**, eds. Bellare M., Springer, Berlin, Heidelberg, 2000, 131–146.

[10] Antipa A., Brown D., Menezes A., Struik R., Vanstone S., "Validation of elliptic curve public keys", Public Key Cryptography – PKC 2003, LNCS, **2567**, eds. Desmedt Y. G., Springer, Berlin, Heidelberg, 2003, 211–223.

[11] Lim C. H., Lee P. J., "A key recovery attack on discrete log-based schemes using a prime order subgroup", Advances in Cryptology – CRYPTO 1997, LNCS, **1294**, eds. Kaliski B. S., Springer, Berlin, Heidelberg, 1997, 249–263.

[12] luigi1111, Spagni R. "fluffypony", https://www.getmonero.org/2017/05/17/disclosure-of-a-major-bug-in-cryptonote-based-currencies.html, 2017.

[13] Miret J., Moreno R., Rio A., Valls M., "Determining the 2-Sylow subgroup of an elliptic curve over a finite field", *Mathematics of Computation*, **74**:249 (2005), 411–427.

[14] Miret J., Moreno R., Rio A., Valls M., "Computing the ℓ-power torsion of an elliptic curve over a finite field", *Mathematics of Computation*, **78**:267 (2009), 1767–1786.

[15] Electric Coin Company, *What is Jubjub?*, https://z.cash/technology/jubjub.

[16] Masson S., Sanso A., Zhang Z., *Bandersnatch: a fast elliptic curve built over the BLS12-381 scalar field*, https://eprint.iacr.org/2021/1152, 2021.

[17] Hamburg M., "Decaf: Eliminating cofactors through point compression", Advances in Cryptology – CRYPTO 2015, LNCS, **9215**, eds. Gennaro R., Robshaw M., Springer, Berlin, Heidelberg, 2015, 705–723.

[18] Hamburg M., de Valence H., Lovecruft I., Arcieri T., *Ristretto*, https://ristretto.group/ristretto.html.

[19] de Valence H., Grigg J., Tankersley G., Valsorda F., Lovecruft I., Hamburg M., *The ristretto255 and decaf448 groups*, https://datatracker.ietf.org/doc/draft-irtf-cfrg-ristretto255-decaf448, 2021.

[20] Barreto P. S. L. M., Costello C., Misoczki R., Naehrig M., Pereira G. C. C. F., Zanon G., "Subgroup security in pairing-based cryptography", Progress in Cryptology – LATINCRYPT 2015, LNCS, **9230**, eds. Lauter K., Rodríguez-Henríquez F., Springer, Cham, 2015, 245–265.

[21] Bowe S., *Faster subgroup checks for BLS12-381*, https://eprint.iacr.org/2019/814, 2019.

[22] Scott M., *A note on group membership tests for $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ on BLS pairing-friendly curves*, https://eprint.iacr.org/2021/1130, 2021.

[23] Enge A., "Bilinear pairings on elliptic curves", *Enseignement Mathématique*, **61**:1 (2016), 211–243.

[24] Joye M., Lapiha O., Nguyen K., Naccache D., "The eleventh power residue symbol", *Journal of Mathematical Cryptology*, **15**:1 (2021), 111–122.

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

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

[27] Weilert A., "Fast computation of the biquadratic residue symbol", *Journal of Number Theory*, **96**:1 (2002), 133–151.

[28] Damgård I. B., Frandsen G. S., "Efficient algorithms for the gcd and cubic residuosity in the ring of Eisenstein integers", *Journal of Symbolic Computation*, **39**:6 (2005), 643–652.

[29] Bach E., Sandlund B., *On Euclidean methods for cubic and quartic Jacobi symbols*, https://arxiv.org/abs/1807.07719, 2018.

[30] Wikström D., *On the l-ary gcd-algorithm and computing residue symbols*, https://www.csc.kth.se/~dog/research/papers/Wik04TR.pdf, 2004.

[31] Husemöller D., *Elliptic curves*, Graduate Texts in Mathematics, **111**, Springer, New York, 2004.

[32] Pornin T., *Double-odd elliptic curves*, https://eprint.iacr.org/2020/1558, 2020.

[33] Pornin T., Bottinelli P., Doussot G., Schorn E., *Double-odd elliptic curves*, https://doubleodd.group.

[34] Costello C., Jao D., Longa P., Naehrig M., Renes J., Urbanik D., "Efficient compression of SIDH public keys", Advances in Cryptology – EUROCRYPT 2017, LNCS, **10210**, eds. Coron J.-S., Nielsen J., Springer, Cham, 2017, 679–706.

[35] *Bandersnatch implementation notes*, https://hackmd.io/wliPP_RMT4emsucVuCqfHA?view, 2021.

[36] Hopwood D., *Calculate circuit costs of prime group operations (Ristretto or ctEdwards-subgroup)*, https://github.com/zcash/zcash/issues/4024, 2019.