Security Assessment for the Helios–Selene Curve Cycle

Alp Bassa and Ben Sepanski

1 Introduction

In August 2025 Veridise was engaged by Monero to conduct an assessment of security and suitability for FMCP++ [13] of the Helios/Selene curve cycle introduced by tevador [16].

Further details have been provided below. The result of the assessment can be summarized below:

1.1 Summary

As cycles of curves necessarily have prime order, some of the criteria applied to curves (see for instance [5]), in particular in the DH context, are not applicable and not necessary for the very particular intended use. In particular criteria about ladders, completeness and indistinguishability can only be partially satisfied, but this does not pose a security threat for the particular use. Care should however be taken during the implementation for possible shortcomings of the curves in this respect. In particular it should be ensured that curve membership checks for points are performed whenever necessary. Moreover, completeness issues (operations involving the identity point, doublings and additions to inverses) should be handled carefully during implementation.

As the only feasible way of constructing cycles of curves of cryptographic sizes is through the use of complex multiplication methods, the resulting curves will (by construction) have small discriminants. As in the case of pairing friendly curves, requiring a lower bound of 2^{100} as in the case of SafeCurves is not an option. Computationally it would be possible to reach CM field discriminants at the order of 2^{50}, but because of the sparsity of suitable discriminants at this order this might imply security and efficiency sacrifices at other places, hence this is not required.

There are however more serious issues regarding rigidity and twist security. For details see Sections 2.7.6 and 2.7.7.

1.2 Recommendation

As the search for a curve pair is a task to be performed once, it is crucial to conduct it with the aim of ensuring the highest achievable security. As twist security for both curves in the cycle can be easily achieved (see Section 2.7.7 and Table 1 in Section 4), including it among the search criteria is natural and essential. Subsequently, a detailed investigation of the distribution of operations to be performed during operation will reveal meaningful conditions to be included to ensure efficiency. The search should be conducted with all conditions well-defined and explicitly stated, in a transparent and reproducible way. This is crucial to mitigate any suspicions about ill intentions and ensure a wider adoption. Although the Helios–Selene curve cycle has many desirable properties, it falls short in all three of the above steps. Twist security for both curves is small (around 100 bits) as it had not been included as a search criterion. The choice of curves leaves room for improvement in terms of efficiency, as can be seen by the fact that (twist secure)

curves with better parameters for the incorporated efficiency consideration (e.g. bits in Crandall reduction) are readily found. It is conceivable that further improvements are within reach by a deeper analysis of the requirements and the inclusion of relevant conditions in the search criteria. Lastly, the curve cycle does not seem to be the first satisfying all search conditions, but rather is a subsequent one with lower twist security. This might cast doubt on the choice of the curve parameters and impact trust and adoption.

Better cycles (in terms of security and efficiency) are readily available. As an example, curve cycles constructed from discriminant -31617403 have twist security and require only multiplication by a 125 bit scalar during Crandall reduction, have $q-1$ not divisible by any small prime and satisfy all other search conditions. This discriminant is the smallest (in absolute value) satisfying all conditions (including twist security) and as such would be constructed in a well-defined and transparent way. This example should not be considered as a recommendation, but merely has been included to show that improvements for security and efficiency are easily within reach.

The search for the curve cycle is only to be performed once and each subsequent change will need to be well motivated and explained. As such, it deserves being done meticulously and carefully. As such, we recommend performing a detailed analysis of the requirements (particularly with regards to efficiency) in the intended application and repeating the search with the aim of finding an optimal cycle in terms of security and efficiency.

In particular, deciding on the particular architecture and algorithms to be used in the implementation based on a detailed analysis of the application before the search will be beneficial. Subsequently relevant conditions can be included as search criteria.

Comments on the current search criteria have been provided in Section 2.6.1. In particular some of the conditions could potentially be weakened to allow space for security and efficiency gains in other places.

2 Background

2.1 Cycles of Curves

Cycles of curves were defined in [2] and consists of a pair of elliptic curves $E_p/\mathbb{F}_p$ and $E_q/\mathbb{F}_q$, such that the cardinality of the base field of each of the curves is equal to the group order of the other one, i.e. $|E_p(\mathbb{F}_p)| = q$ and $|E_q(\mathbb{F}_q)| = p$. A curve cycle for a suitable prime pair p, q can provide substantial efficiency gains for zk-proofs by eliminating emulation of foreign field arithmetic in the constraints and replacing it by two classes of constraints over the fields $\mathbb{F}_p$ and $\mathbb{F}_q$. The curve Ed25519 ([3]) is defined over the field $\mathbb{F}_p$ with $p = 2^{255} - 19$ and plays a prominent role for Monero anonymity sets. Hence, having a curve cycle involving Ed25519 is desirable for the efficiency of the corresponding zk-proofs. This however is not possible, as the group of rational points on Ed25519 is not prime order, but of the form 4ℓ for a prime ℓ. Rather, one considers a cycle of curves E_p/E_q of prime order defined over the fields $\mathbb{F}_p$ and $\mathbb{F}_q$, with $p = 2^{255} - 19$ the base field of the Ed25519. The curve Ed25519 can then be embedded in the scalar field of E_q, which has cardinality p.

2.2 SafeCurves

Cycles of curves with nonprime order, i.e., curves $E/\mathbb{F}_p$ and $E'/\mathbb{F}_q$, with $\#E(\mathbb{F}_p) = h \cdot q$ and $\#E'(\mathbb{F}_q) = h' \cdot p$ do no exist for $p, q > 4$ (see [8]). As we are forced to look for cycles of prime order curves, some of the criteria in SafeCurves are not applicable. In particular, the SafeCurves conditions on ladders, completeness and indistinguishability can only be partially satisfied. We

will still use the SafeCurves criteria as a guideline, as is customary during the construction of cycles of elliptic curves and give details regarding all deviations.

2.3 Related Results in the Literature

The question of finding elliptic curves with $2^{255} - 19$ rational points and the related question of cycles involving the Ed25519 basefield have been considered in the past. Poelstra [14] raised it in the Curves mailing list. Guillevic and Masson [9] performed a search for cycles of curves such that Ed25519 can be embedded in one of them (one of them has cardinality $2^{255} - 19$). They searched over square-free discriminants $D \equiv 3 \pmod 4$ in the range $1 - 10^{10}$. They mention two notable discriminants: $D = -65012179$ and $D = -103953715$ (note that the second discriminant is mentioned in a talk by Guillevic [10]). The curves in the cycle with discriminant $D = -65012179$ have twist security, as one curve has a quadratic twist of prime order and the other one has a twist whose order has a prime factor of 215 bits. Both curves do not satisfy the search criteria that led to Helios-Selene. The discriminant -65012179 cycle has q which is too small such that the difference to 2^{255} does not fit into two 64 bit words (condition (iii) in Section 2.6), the $D = -103953715$ cycle has q larger than 2^{255} (condition (i) in Section 2.6). See also [1].

2.4 Other cycles of Curves

Other cycles of curves have previously been constructed. We are not concerned with pairings on the curves. Such cycles are referred to as plain cycles. The Tweedledum–Tweedledee cycle of curves has been constructed for Halo [6]. For the Halo2 project these have been replaced by the Pallas and Vesta curves.

2.5 Definition of the Curve/Ed25519 Curve and the Helios/Selene Curve Cycle

2.5.1 Curve25519/Ed25519

Curve25519 is a Montgomery curve defined over $\mathbb{F}_p$ with $p = 2^{255} - 19$, and given by the equation

$$y^2 = x^3 + 486662x^2 + x.$$

It is birationally isomorphic to the twisted Edwards curve Ed25519 given by

$$-x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$$

and has order 8ℓ with $\ell = 2^{252} + 27742317777372353535851937790883648493$.

As our principal concern is the cardinality of its basefield, by abuse of nomenclature we will use the name Ed25519 for the birational isomorphism class.

2.5.2 Helios

The Helios curve is defined over $\mathbb{F}_p$ with $p = 2^{255} - 19$ (the basefield of the Ed25519 curve), and is given by the equation $y^2 = x^3 - 3 \cdot x + b$ with

$$b = 15789920373731020205926570676277057129217619222039203958068448089789960834 12$$

The base point is chosen to be the point with smallest x coordinate and even y coordinate:

$$(3, 37760095087190773158272406437720879471285821656958791565335581949097084993268)$$

2.5.3 Selene

The Selene curve is defined over $\mathbb{F}_q$ with $q = 2^{255} - 85737960593035654572250192257530476641$, and is given by the equation $y^2 = x^3 - 3 \cdot x + b$ with

$$b = 50691664119640283727448954162351551669994268339720539671652090628799494505816$$

The base point is chosen to be one of the points with smallest x coordinate:

$$(1, 55227837453588766352929163364143300868577356225733378474337919561890377498066)$$

Note that Helios and Selene form a cycle with $E_p(\mathbb{F}_p) = q$ and $E_q(\mathbb{F}_q) = p$.

2.6 Search that Lead to the Helios–Selene Cycle

The search that resulted in the Helios–Selene cycles [16] imposed several additional conditions motivated by ease and efficiency of the implementation.

(i) $q < 2^{255}$

This is to ensure that compressed points of E_q can be encoded with 256 bits and corresponds to considering only curves with positive Frobenius trace.

(ii) $q \equiv 3 \pmod 4$

This is for efficient square root calculation in $\mathbb{F}_q$.

(iii) $q > 2^{255} - 2^{127}$.

This is to ensure fast operations during Crandall's modular reduction algorithm.

(iv) Both E_p and E_q should be of the form $y^2 = x^3 - 3 \cdot x + b$.

This is to be able to use more efficient group law formulas.

(v) The constant term b in the curve equation should be a non-square in the base field. This ensures that none of the points has x-coordinate 0, which prevents power analysis attacks.

2.6.1 Comments on the Search Criteria

On condition (ii)

The condition $q \equiv 3 \pmod 4$ is included to ensure efficient square root calculation in $\mathbb{F}_q$. However for $q \equiv 5 \pmod 8$ efficient square root computation methods are available as well (as used in the case of the curve Ed25519). Hence including $q \equiv 5 \pmod 8$ could be reconsidered, especially if the larger search space allows for better security or other efficiency gains.

On condition (iii)

Let $\gamma = 2^{255} - q$. Crandall's reduction algorithm requires multiplications by 2γ. Hence the condition $2\gamma < 2^{128}$ (i.e. $\gamma < 2^{127}$) is included among the search conditions so that it fits into two machine words on 64-bit CPUs, making modular reductions faster. However multiplication by 2γ can be realized by a multiplication by γ followed by a fast left shift. Replacing the condition by $\gamma < 2^{128}$ will lead to a larger search space, within which finding more secure curves (with regards to twist security for instance, one could then use the curve corresponding to the discriminant -65012179 mentioned in Section lit) or curves with other properties increasing efficiency will be possible. One can for instance aim to optimize the bit representation of γ to speed up multiplication. Values of q such that γ has few bits or $\gamma - 1$ is highly divisible by 2 might lead to speed-ups.

We have

$$q = 2^{255} - \gamma = |E_p(\mathbb{F}_p)| = p + 1 - t_p = 2^{255} - 18 - t_p,$$

where t_p is the trace of Frobenius of E_p. Hence by the Hasse-Weil bound, we already have $\gamma = t_p + 18 \leq 2\sqrt{p} + 18 \sim 2^{128.5}$. So $\gamma < 2^{128}$ is a rather mild condition.

Even if doing a shift operation is not desired, optimizing the bit representation of $2^{256} - 2q$ depending on the algorithms and the architecture to be used might be reasonable to include in among the search conditions

On conditions (iv) and (v)

The above criteria can be split into two: conditions (i),(ii),(iii) regarding the prime q (which is determined by the discriminant D), and once q (and hence the discriminant) is fixed, conditions (iv), (v) on the elliptic curve equation of that discriminant. As $|D|$ is rather large, for a given discriminant there is a large number of isomorphism classes of elliptic curves (given by the class number of the imaginary quadratic number field of the given fundamental discriminant, which by the analytic class number formula grows like $\sqrt{|D|}$, up to logarithmic factors). Each isomorphism class contains an elliptic curve $y^2 = x^3 + ax + b$ over $\mathbb{F}_q$, which is isomorphic over $\mathbb{F}_q$ to an elliptic curve of the form $y^2 = x^3 - 3x + b$ exactly when $-3/a$ is a 4-th power in $\mathbb{F}_q$. This should happen fairly often (with probability around $1/4$ in case of uniform distribution). Similarly, an elliptic curve with constant term $b = 0$ will have j-invariant 1728, which will correspond to at most one of the roots of the $\sim \sqrt{|D|}$ roots of the Hilbert class polynomial. Hence for a given (large enough) $|D|$, a curve satisfying conditions (iv) and (v) should be easy to find.

2.7 SafeCurve Criteria

2.7.1 Field of Definition

Both $\mathbb{F}_p$ and $\mathbb{F}_q$ are prime fields, as required by Safecurves (and also by Brainpool standard and NSA Suite B standard). We have $p \equiv 5 \pmod 8$ (there is no choice here as it is supposed to be the order of the Ed25519 basefield) and $q \equiv 3 \pmod 4$, allowing faster square-root computations. Note that $q \equiv 5 \pmod 8$ would also have been an option, see Section 2.6.1 (ii).

2.7.2 Curve Equations

Both curves are given by short Weierstrass equations with $a = -3$.

2.7.3 Group Order

Both p and q are 255 bit and hence substantially bigger than 2^{200}, the lower bound required by Safecurves.

2.7.4 Embedding Degree

Helios has an embedding degree of $\frac{q-1}{2}$, Selene has an embedding degree of $\frac{p-1}{2}$. Hence, both satisfy the Safecurve criteria (which agrees with the Brainpool criteria) of having embedding degree at least $\frac{\ell-1}{100}$ for a prime order group of order ℓ.

2.7.5 CM Field Discriminant

The cycle of curves is constructed using the theory of Complex Multiplication. Hence both resulting curves have CM by an order in an imaginary quadratic field of fairly low absolute discriminant ($D = -7857907$). This seems unavoidable given that the use of CM is the only known viable method.

Trying to pick random curves until one with the right cardinality is obtained is not feasible, as the prime p is too large and this would require an average of $2\sqrt{p}+O(1)$ trials. The optimizations along the lines as the ones given in [15, Section 3] can be implemented, but would most likely not be sufficient for a prime of this size.

Using the CM method, the lower bound 2^{100} required by SafeCurves is out of reach, however discriminants at the order of 2^{50} are easily within reach using [7] for finding a suitable discriminant and [15] to find the Hilbert class polynomial and the equations (see also the example with discriminant $\sim 10^{13}$ below). Further, even the smaller discriminants obtained in this search (e.g. the discriminant $D = -7858907$) are large enough so that a GLV type construction to speed-up multiplication and the corresponding rho method are not feasible. As the degree of the endomorphisms are lower bounded by $|D|/4$, there are no low degree endomorphisms.

Fortunately, requiring large discriminants is a preemptive measure, to protect against the possibility that the use of special low CM discriminant curves can be exploited in some way in the future. Forgoing this criteria is customary for all cycles of elliptic curves used. In particular, the secp256k1/secq256k1 [14], Tweedledum/Tweedledee [6], Pallas/Vesta, Pluto/Eris pairs all have complex multiplication by an order in a field of discriminant -3.

2.7.6 Rigidity

In order not to give rise to any suspicion about backdoors and ill intentions while choosing the curve parameters, the curves should be generated using a transparent and reproducible process. The use of such a process is called "rigidity". The generated trust is crucial to ensure wider adoption. A general practice is to choose the smallest element / first occurrence in various situations.

However, although discriminants -1571315 and -7194667 give rise to curves satisfying all search conditions, the absolutely larger discriminant -7857907 is used to obtain the Helios–Selene cycle of curves. Hence the statement

"The first discriminant value that produces a prime q that matches our conditions (1), (2) and (3) is D = -7857907"

from [16] is incorrect.

This is particularly worrisome, as both of the curves in the Helios–Selene curve cycle have very low twist security, namely 99 bits and 107 bits. In fact, among the $84 = 42 \cdot 2$ curves in Table 1 in Section 4 satisfying all search criteria, only 7 have twist security less than 99 and 13 have twist security less than 107. One of the discriminants ($D = -1571315$) seems to have been identified before by tevador on March 9, 2023[1]. This choice of a subsequent curve cycle with both curves of particularly low twist security seems arbitrary and might undermine trust in the choice of curve parameters.

One reason might be that this is the first discriminant such that $q - 1$ is not divisible by 3 (or other small primes). This however has not been included as a search condition.

Subsequently added remark: after this point has been raised, the script used by tevador for the search was inspected and the reason for the omission turned out to be a bug. These discriminants had been missed as not all roots of unity in the ring have been taken into consideration. As such, the choice was the most rigid one under false premises. This subsequently has been corrected.

[1] `https://github.com/monero-project/research-lab/issues/100\#issuecomment-1460898013`

2.7.7 Twists

A twist of an elliptic curve E is another elliptic curve that is isomorphic to E over the algebraic closure of the base field. For an elliptic curve E given by the short Weierstrass equation $y^2 = x^3 + a \cdot x + b$ over $\mathbb{F}_q$, and a non-square $d \in \mathbb{F}_q^\times$, the curve E' over $\mathbb{F}_q$ given by $d \cdot y^2 = x^3 + a \cdot x + b$ (which can be put in a short Weierstrass form) is isomorphic to E over the quadratic extension field $\mathbb{F}_q(\sqrt{d})$. Except in the special cases of j-invariants 0 and 1728, this quadratic twists is (up to isomorphism) the only one. Operations in the elliptic curve and its twist are intimately related. Hence an elliptic curve deemed secure (with respect to other criteria) might not be safe to use in case the twist is weak. As an example, in a given protocol, although the discrete logarithm problem is hard in E, if the necessary membership tests are not performed, a malicious adversary could provide a point on the twist, where the discrete logarithm problem is easy, which would lead to leakage or breaking of the scheme. Even if no such attack is known for the given scheme, there might be other ways of exploiting twist insecurity in the future, and it is prudent not to leave such an attack surface open.

It should be noted that twist security is not an ubiquitous criterion in conducted searches, especially in case the search space is limited because of challenging conditions like high 2-adicity. It was included in the search leading to the Tweedledum/Tweedledee curves [11] for HALO, but it has been dropped in the search leading to the Pallas/Vesta curves [12] for HALO2. Twist security does play a prominent role in [9].

As already mentioned in Section 2.7.6, both curves in the Helios–Selene cycle have particularly low twist security. The order of the twist of the Helios curve factors as

$$11 \cdot 29 \cdot 863 \cdot 23082923729 \cdot 1017099830901941062045156007 03 \cdot 895762787896668507359457364659 07,$$

whereas the twist of the Selene curve has order factoring as

$$424102339 \cdot 2246409397681 \cdot 123021165100832544760186241 \cdot 4939801339394265214070088683 53.$$

Hence the largest prime dividing the order of them are of size 107 bits and 99 bits, respectively. Hence both curves have rather small twist security. As indicated in Section 2.7.6 these values are small compared to curves in cycles with other discriminants. As can be seen in Section 4, twist security can be easily ensured without any other sacrifices. Hence there is no reason to not include it in the search criteria, without even having to consider if it poses a security risk in the intended application. The lack of twist security can be considered a rather serious security risk.

2.7.8 Ladders, Completeness and Indistinguishability

Ladders are uniform, constant-time scalar-multiplication algorithms. Elliptic curves forming cycles necessarily have prime order groups. Hence they are not birationally isomorphic to Edwards curves or Montgomery curves. As such they only support the Brior–Joye ladder.

Completeness requires the curve to have group law formulas which work without exceptions. Efficient complete formulas are not readily available for any curve of prime order. Care must be taken during implementation for timing variations and missed checks for cases violating completeness.

Indistinguishability requires that encodings of curve points are indistinguishable from random bits. For Weierstrass curves, this is not satisfied by the standard encodings as many x-coordinates do not correspond to points on a curve. For all curves of prime order, standard techniques like Elligator / Elligator 2 [4] are not applicable. However, Elligator squared [17] can be used if desired. Fortunately, indistinguishability is not a priority for the given application. Attackers

are expected to already know the ciphersuite in use, and verifiers are expected to perform curve checks on any received points.

It should be noted that the curves in the cycle necessarily have prime order. Given the current state of knowledge, the above comments would apply to all such curves and hence are not a shortcoming of the Helios–Selene curve cycle.

3 Some Notable Curve Pairs

Below we collect a list of notable discriminant giving particularly interesting curve pairs. This is not intended as a suggestion or constitutes the search of an extensive search, but shows that alternative curve pairs with better security/efficiency properties are available.

- **D=-31750123, q=0x7ffffffffffffffffffffffffffff735481d1969f317f9850b68df11df53**
  $q \equiv 3 \mod 4$, there are no odd primes < 15 dividing $q - 1$, twist security 222 bits / 211 bits

  This curve satisfies all the search conditions, has no small prime dividing $q - 1$, however compared to the Helios–Selene cycle has substantially larger twist security 222/211 bits (hence satisfying the SafeCurves criteria) and fewer bits for $2^{255} - q$ (124 instead of 127). This would be the first discriminant with twist security satisfying all search conditions.

  Other discriminants with twist security satisfying all search conditions, for which however $q - 1$ is divisible by some small primes are -3583705955 (see also below), -9410900827 and -9983910435.

- **D=-1724391267, q=0x7fffffffffffffffffffffffeaa752b72586c573a1f8ff8b64db49bff**
  $q \equiv 3 \mod 4$, odd primes < 15 dividing $q - 1$ are $3, 13$, twist security 127 bits / 226 bits

  For $\gamma = 2^{255} - q$, we have $2^{10} | \gamma - 1$. Letting $\theta = (\gamma - 1)/2^{10}$ (119 bits) in Crandall's reduction algorithm multiplication of x by $2^{256} - 2q$ can be performed by

  - multiplication by θ (119 bits)
  - 10 left shifts
  - addition of x
  - 1 left shift

  This example shows that interesting cycles might be available, without restrictions on the size of $\gamma = 2^{255} - q$.

- **D=-3583705955, q=0x7fffffffffffffffffffffffffe84b3e3ffc7c2e6fcc9de2cbff2d96cf**
  $q \equiv 3 \mod 4$, odd primes < 15 dividing $q - 1$ are 3, twist security 210 bits / 234 bits

  $\gamma = 2^{255} - q$ has only 125 bits, out of which just 49 are 1. So multiplication by γ will be faster. Moreover $2^{256} - 2q$ fits into 128 bits.

- **D=-837862243, q=0x7fffffffffffffffffffffffffa0af08dbd67166bccbb685d35e50f9**
  $q \equiv 1 \mod 8$, odd primes < 15 dividing $q - 1$ are $3, 11$, twist security 126 bit / 127 bits

  $\gamma = 2^{255} - q$ has 119 bits. Note however that $q \equiv 1 \mod 8$. This example is more to exemplify that noticeable improvements might be possible in Crandall's reduction algorithm.

- **D=-10013341781587, q=0x7fffffffffffffffffffffffffc427869469b5b2699e2e37ca1e0b4d0f**
  $q \equiv 3 \mod 4$, odd primes < 15 dividing $q - 1$ are 3, twist security 147 bits/124 bits

  Note that $|D| > 2^{43}$. So going up to large absolute discriminants is possible.

- **D=-92169307, q=0x7fffffffffffffffffffffffffbc0147e1eea14722afe38c177dbf8595** $q \equiv 5 \mod 8$, there are no odd primes < 15 dividing $q-1$, twist security 247 bits / 256 bits

  The twist E'_p has prime order (256 bits), whereas the twist E'_q has order divisible by a 247 bit prime. This would be the first discriminant if twist security would have been included in the search conditions, with $q \equiv 5 \mod 8$.

- **D=-65012179, q=0x7fffffffffffffffffffffffff34a2208109393ca351aa6d362f601a5f** $q \equiv 3 \mod 4$, there are no odd primes < 15 dividing $q-1$, twist security 255 bits / 215 bits

  The twist E'_q has prime order (255 bits), whereas the twist E'_q has order divisible by a 215 bit prime. This is the curve from [9] mentioned in Section 2.3. The condition on γ is satisfied in its weakened form (see Section 2.6.1)

4 Tables

Table 1 lists discriminants for the CM field up to around 10^{10}, for which cycles satisfying conditions (i) and (ii) from Section 2.6 are satisfied. As explained in Section 2.6, for a given discriminant (which is sufficiently large in absolute value), conditions (iv) and (v) should not cause any problem. Hence these have not been included in the search. Equations for curves of the given discriminant can be easily found by computing the Hilbert class polynomial. Moreover, condition (ii) seems too strict as $q \equiv 5 \mod 8$ would also be an interesting choice. Condition (ii) has been replaced by a coloring scheme in the second column.

Column 1 lists the discriminant of the CM field. The discriminant corresponding to the Helios–Selene pair is highlighted in red. The two discriminants that are (absolutely) smaller than the discriminant of the CM field of the Helios–Selene pair, which satisfy all search criteria but have been skipped are highlighted in orange. Discriminants mentioned in Section 3 are highlighted in cyan.

Column 2 lists the residue class of resulting q modulo 8. Hence discriminants giving q

- satisfying condition (ii) have been highlighted in green,
- not satisfying condition (ii) but $q \equiv 5 \mod 8$ have been highlighted in lime.

Column 3 and 4 list the size of the largest prime divisor (in bits) of the order of the groups of the twists E'_q and E'_p. Curves satisfying the relevant Safecurves criteria of being larger than 200 are highlighted in green.

Column 5 lists the size of $\gamma = 2^{255} - q$. Crandall reduction will require multiplication by 2γ.

Column 6 lists all odd primes less than 15, which divide $q-1$.

The embedding degree of both curves in all cases is at least $(\ell-1)/12$ where ℓ is the order of the group and hence has not been included in the table. Embedding degrees of twists have not been computed.

The same coloring scheme applies to Table 2, Table 3 and Table 4.

Table 2 presents examples of discriminants of absolute value around $10^{10}, 10^{11}, 10^{12}, 10^{13}$, to show that the computations can be pushed to rather large discriminants.

Table 3 lists discriminants (up to around $5 \cdot 10^9$) which would have been included to Table 1 if the condition (iii) would have been weakened to $\gamma < 2^{128}$ as explained in Section 2.6.1. The Columns Crandall has been dropped, as all would have value 128.

Table 4 lists discriminants (up to around $5 \cdot 10^9$)which would have been included if even the weakened form of condition (iii) would have been dropped.

Table 1

D	$q \mod 8$	Twist E'_q	Twist E'_p	Crandall	Primes
-91515	1	194	115	127	3
-1099643	1	128	146	127	3
-1571315	7	119	139	127	3
-7194667	7	148	241	127	3
-7857907	7	99	107	127	
-31617403	5	216	79	125	3
-31750123	3	222	211	124	
-62002483	1	132	155	127	5
-83232771	1	105	216	126	3, 7
-92169307	5	247	256	127	
-93036315	3	103	209	127	3
-100055947	7	143	145	125	
-154142011	1	189	231	126	13
-189935715	3	252	199	127	3, 5
-204370795	3	115	132	127	5, 7
-268755307	5	203	128	127	
-281357651	1	87	113	127	3, 5, 11
-289977907	3	177	86	126	
-367827107	5	150	114	127	3
-406549355	3	127	245	127	3, 11
-471122787	3	195	242	126	3
-490527507	7	114	190	127	3
-614880843	1	228	175	127	3
-651912619	5	134	233	127	5
-669378715	7	251	159	123	5, 7, 11
-692013211	5	180	135	124	
-699283483	5	132	165	126	7
-741440283	3	89	105	127	3, 13
-770929963	1	168	149	123	5
-837862243	1	126	127	119	3, 11
-881184867	1	135	97	126	3, 7
-895993939	3	113	160	126	
-912034243	7	95	211	127	
-1010643747	1	62	100	127	3, 11
-1089513107	7	126	252	124	3, 5
-1172942227	1	142	249	127	
-1215043867	1	129	117	126	3
-1354169499	5	127	87	126	3, 5
-1374050707	7	151	248	127	
-1394762683	7	200	104	127	7
-1517757915	5	227	208	127	3, 5
-1818125835	1	105	191	127	3
-1835848059	3	195	99	127	3, 5, 7

D	$q \mod 8$	Twist E'_q	Twist E'_p	Crandall	Primes
-1995059795	1	187	221	127	3, 11
-2131571515	7	247	100	127	5, 7
-2212708723	1	99	109	126	3
-2353453747	7	196	139	124	
-2525798283	7	90	200	127	3
-2642577915	3	108	133	126	3, 5
-2801790139	7	248	156	125	3, 5
-2805888835	7	151	193	126	5
-2930327779	1	178	183	127	5
-2962945987	1	176	130	127	5
-3166390427	7	92	115	127	3
-3211572435	7	146	218	127	3, 5
-3230015947	7	140	180	127	3
-3524381403	5	92	124	127	3
-3583705955	7	210	234	125	3
-4135140123	5	191	144	127	3
-4567338483	5	235	69	127	3, 11
-4598074835	1	216	102	127	3, 5
-4613882755	1	175	221	127	3, 5, 7
-4737079667	7	86	237	126	3
-4758921043	1	240	123	126	
-5229149115	1	248	113	125	3, 5
-5434594435	1	182	109	126	3
-5479193715	1	141	119	126	3
-5703206107	1	107	187	126	
-5722075923	5	201	254	127	3
-5965737259	7	75	244	126	3, 7
-6123900595	1	243	233	127	3, 5
-6488926635	5	153	132	127	3
-6688734963	1	149	205	127	3
-6725255035	7	126	213	126	3
-6771269523	7	154	169	127	3
-6896958483	1	171	127	125	3
-6918310507	7	109	195	127	
-7049755707	1	233	136	126	3, 11
-7838599939	3	132	214	127	5, 13
-8174174443	1	126	107	127	5
-8304544315	1	217	228	124	
-8504824947	7	251	118	127	3
-8662630387	3	139	126	127	7, 11
-9410900827	3	231	238	126	7, 11
-9775832827	5	171	109	126	
-9894819243	3	210	150	124	3
-9907447299	5	115	94	126	3
-9983910435	3	228	234	127	3
-10025391451	5	217	102	127	5, 7
-10664400755	1	177	88	127	3
-10858666827	3	138	191	127	3, 5

Table 2

D	$q \mod 8$	Twist E'_q	Twist E'_p	Crandall	Primes
-10025391451	5	217	102	127	5, 7
-101262509731	1	236	184	126	7, 11
-1006526355115	5	176	215	127	5
-10013341781587	7	147	124	126	3

Table 3

D	$q \mod 8$	Twist E'_q	Twist E'_p	Primes
-167995	3	190	226	11, 13
-4059339	3	120	236	3
-18923731	5	108	185	
-25799635	1	227	109	5, 13
-43584691	3	104	108	3
-54558307	3	162	138	
-65012179	7	255	215	
-67034235	1	85	243	3
-125442843	3	100	123	3
-137469067	5	178	175	
-151301243	1	137	114	3
-193762963	5	255	176	13
-251868963	3	135	137	3, 11
-261756723	3	224	205	3
-262700395	3	225	237	5, 7
-309744003	5	108	113	3, 5, 13
-312573115	1	248	191	3, 5
-328922115	1	180	163	3
-373562851	7	203	195	3
-481212883	1	238	140	3
-500038611	5	233	128	3
-596047827	1	93	162	3, 5
-666555963	5	152	130	3, 7
-705663627	7	86	144	3
-731837107	1	172	128	3, 11
-776013627	5	174	148	3
-865378155	7	116	140	3, 7
-947098723	3	246	129	
-972154635	5	97	157	3
-1049422603	1	144	205	7
-1059124483	1	108	165	
-1147887187	5	227	105	3
-1212288555	1	240	146	3, 5
-1386591643	1	118	152	
-1413912427	5	116	215	11
-1494382587	1	249	93	3, 5, 7, 11
-1595595835	7	247	208	5

D	$q \mod 8$	Twist E'_q	Twist E'_p	Primes
-1612531835	1	126	106	3, 5, 7
-1619681331	5	67	145	3, 5
-1694323483	1	159	172	3, 7
-1769791915	7	207	138	5, 11
-1907668355	3	153	99	3, 5
-2021256547	5	193	173	
-2457387411	3	149	118	3
-2694141867	3	128	74	3
-2802073987	1	144	133	3
-2810238595	7	153	231	7
-3034806811	7	80	185	5
-3234663955	5	102	134	
-3527828683	1	199	172	3, 5
-3774936147	3	101	86	3, 7
-3808174515	3	207	129	3
-4099049139	7	194	181	3, 5
-4679458219	1	143	239	
-4755806635	5	181	98	7
-4823544723	1	221	199	3, 13
-4824928123	3	214	241	
-4853777307	5	182	124	3
-4963245627	5	69	243	3
-5005709539	3	135	235	7
-5230400155	3	179	147	5
-5306951827	5	168	251	
-5371705315	5	103	237	3

Table 4

D	$q \mod 8$	Primes
-15203	7	3, 7, 11
-124123	1	3, 7
-2504947	3	3, 13
-3574995	3	3
-4871323	5	
-6384387	7	3
-9502827	3	3, 5, 7
-15578291	7	3, 5
-18522507	1	3, 7
-23684667	3	3
-25165435	3	5
-32542115	3	3
-38775003	7	3
-43623427	3	
-45099667	1	3
-48584915	3	3, 11
-70277483	7	3, 7, 11, 13

D	$q \mod 8$	Primes
-73323867	5	3
-87850195	7	13
-126773299	5	3
-143195043	5	3
-152236011	1	3, 5
-156015787	7	
-170373587	1	3, 7
-194703763	3	
-199610995	1	3, 5, 7
-205298923	7	11
-208217755	3	
-210297027	1	3
-264577107	5	3
-277795347	5	3, 11
-287159195	7	3, 5
-304761955	7	5, 13
-305284715	7	3, 5
-329175307	3	7
-333439635	7	3
-360493435	7	
-389101003	3	3
-389660035	5	5
-439272595	5	
-450121907	7	3, 5
-461503443	3	3
-461770955	1	3, 5
-473093835	1	3, 5
-600912419	7	3
-612118435	7	3, 7
-628092987	3	3, 11
-699819283	3	
-800524003	1	7
-819034627	7	3, 5, 11
-825306835	3	3, 5, 11
-826299427	7	
-865352947	7	5
-881066443	3	11
-901647795	5	3
-926478659	7	3, 5
-953198683	5	3, 7
-995997227	1	3
-1062355939	7	3, 5
-1086027659	7	3, 5, 7
-1125152467	5	7
-1275924667	3	3, 7
-1291726603	3	7
-1301047987	5	3, 5
-1391777627	1	3, 5

D	q mod 8	Primes
-1451089307	1	3
-1460297779	5	5
-1465283467	5	
-1491820467	5	3
-1543878795	7	3
-1615947195	1	3, 5
-1723977555	5	3, 5, 7, 11
-1724391267	7	3, 13
-1876414555	5	3
-2091732267	1	3
-2119944067	7	3
-2147644115	5	3, 11
-2211056115	3	3, 5
-2216403331	5	5
-2291103299	5	3, 5
-2344451467	7	
-2465568283	3	7
-2562327843	5	3
-2582962307	5	3, 5
-2771541547	7	5, 7
-2791066947	7	3, 11
-2973393307	1	7
-3146800715	1	3, 5
-3227069315	7	3, 5
-3234250963	1	7
-3379252683	3	3
-3501901987	7	
-3919555803	7	3
-4166657587	1	3
-4175939107	5	
-4285764555	5	3, 5
-4405097603	7	3
-4454061683	1	3, 11
-4465086595	1	
-4481168843	1	3
-4486120147	3	3, 7
-4607810635	7	5
-4683790459	1	7
-4689579835	7	3, 5, 7
-4770038731	7	7
-4823936867	7	3
-4832549619	7	3
-4845041979	5	3
-4897587755	5	3
-4955013067	5	3, 11
-5013088323	7	3
-5078418115	7	3, 5
-5114646643	5	

D	$q \mod 8$	Primes
-5126212515	3	3
-5143174843	1	

References

[1] Aranha, D.F., El Housni, Y., Guillevic, A, *A survey of elliptic curves for proof systems*, Des. Codes Cryptogr. 91, 3333–3378 (2023).

[2] Ben-Saison, E., Chiesa, A., Tromer, E., Virza, M., *Scalable Zero Knowledge via Cycles of Elliptic Curves*. In: Garay, J.A., Gennaro, R. (eds) Advances in Cryptology – CRYPTO 2014. CRYPTO 2014. Lecture Notes in Computer Science, vol 8617. Springer, Berlin, Heidelberg.

[3] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., Yang, B.-Y., *High-speed high-security signatures*, Journal of Cryptographic Engineering 2 (2012), 77–89.

[4] Bernstein, D. J., Hamburg, M., Krasnova, A., Lange, T., *Elligator: Elliptic-curve points indistinguishable from uniform random strings*, ACM Conference on Computer and Communications Security 2013.

[5] Bernstein, D. J., Lange, T., *SafeCurves: choosing safe curves for elliptic-curve cryptography*, `https://safecurves.cr.yp.to`, accessed 27 August 2012.

[6] Bowe, S., Grigg, J., Hopwood, D., *Recursive Proof Composition without a Trusted Setup*, Cryptology ePrint Archive, Paper 2019/1021, 2019.

[7] Bröker, R., Stevenhagen, P., *Efficient CM-Constructions of Elliptic Curves over Finite Fields*, Mathematics of Computation, vol. 76, no. 260, 2007.

[8] Chiesa, A., Chua, L., Weidner, M., *On Cycles of Pairing-Friendly Elliptic Curves*, SIAM Journal on Applied Algebra and Geometry, Vol 3. N. 2, 2019.

[9] Guillevic, A., Masson, S., *Embedded Curves and Embedded Families for SNARK-Friendly Curves*, Cryptology ePrint Archive, Paper 2024/1737, 2024.

[10] Guillevic, A., *Elliptic curves for SNARK and proof systems*, talk at the 25th Workshop on Elliptic Curve Cryptography Taipei, Taiwan. `https://people.rennes.inria.fr/Aurore.Guillevic/talks/2024-10-ECC/24-10-30-ECC-Aurore.pdf`

[11] Hopwood, Daira, `https://github.com/daira/tweedle`.

[12] Hopwood, Daira, `https://github.com/zcash/pasta`.

[13] Parker, Luke "Kayaba", *FCMP++*, manuscript, May 6, 2024.

[14] Poelstra, A., *Curve with group order 2$\hat{}$255-19*, Curves mailing list, March 21, 2018, `https://moderncrypto.org/mail-archive/curves/2018/000992.html`.

[15] Sutherland, A. V. *Computing Hilbert class polynomials with the Chinese Remainder Theorem*, Mathematics of Computation, vol. 80, no. 273, 2011, pp. 501–38.

[16] tevador, *Elliptic curve tower-cycle for Curve25519*, `https://gist.github.com/tevador/4524c2092178df08996487d4e272b096`

[17] Tibouchi, M., *Elligator Squared: Uniform Points on Elliptic Curves of Prime Order as Uniform Random Strings*, in: Christin, N., Safavi-Naini, R. (eds) Financial Cryptography and Data Security, Lecture Notes in Computer Science, vol 8437, Springer, Berlin, Heidelberg.

A Code to Find Discriminants

Below you can find SAGEMATH code (implementing Cornachia's algorithm / LLL) to find discriminants for cycles of elliptic curves. Note that `LOWER` is set to `2^255-2^127` to satisfy search criterion (ii) from Section 2.6, generating discriminants from Table 1. To use the weaker condition from Section 2.6.1, replace by `2^255-2^128`, generating discriminants from Table 3.

```
LOWER=2^255-2^127
N=2^255-19
N2=2*N
l=floor(2*sqrt(N))
N4=2*N2
k=GF(N)
D=-3
L=[]
while (True):
    d=k(D)
    if d.is_square():
        a=N2
        b=ZZ(d.sqrt())
        if b%2!=D%2: b=N-b
        while (b>l):
            r=a%b
            a=b
            b=r
        x=b
        q=(N+1-b)
        if  q>LOWER and D.divides(b^2-N4):
            quo=(b^2-N4).divide_knowing_divisible_by(D)
            if  (quo.is_square() and q.is_pseudoprime()):
                print(f"\nD={D}, q={q}")
                L.append([D,q])
    D=D-8
    while (not(is_squarefree(-D))): D=D-8
```