

Lollipops of pairing-friendly elliptic curves for composition of proof systems

Craig Costello¹ and Gaurish Korpai²

¹ Microsoft Research, Redmond, USA
craigco@microsoft.com

² University of Arizona, Tucson, USA
gkorpai@arizona.edu

Abstract. We construct *lollipops* of pairing-friendly elliptic curves, which combine pairing-friendly *chains* with pairing-friendly *cycles*. The cycles inside these lollipops allow for unbounded levels of recursive pairing-based proof system composition, while the chains leading into these cycles alleviate a significant drawback of using cycles on their own: the only known cycles of pairing-friendly elliptic curves force the initial part of the circuit to be arithmetised on suboptimal (much larger) finite fields. Lollipops allow this arithmetisation to instead be performed over finite fields of an optimal size, while preserving the unbounded recursion afforded by the cycle.

The notion of pairing-friendly lollipops itself is not novel. In 2019 the Coda + Dekrypt “SNARK challenge” offered a \$20k USD prize for the best lollipop construction, but to our knowledge no lollipops were submitted to the challenge or have since emerged in the literature. This paper therefore gives the first construction of such lollipops.

The main technical ingredient we use is a new way of instantiating pairing-friendly cycles over supersingular curves whose characteristics correspond to those in MNT cycles. The vast majority of MNT cycles that *exist* are unable to be instantiated in practice, because the corresponding CM discriminant is too large to construct the MNT curves explicitly. Our method can be viewed as a workaround that allows cycles to be instantiated regardless of the CM discriminant of the MNT curves.

Keywords: Proof systems · Composition · Pairing-friendly cycles · MNT curves

1 Introduction

Numerous constructions of *2-cycles* of elliptic curves are now deployed as the foundation of succinct non-interactive arguments of knowledge (SNARKs) – see [1] for an extensive survey. Such 2-cycles involve two elliptic curves, E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q , with $p \approx q$ such that $p = \#\hat{E}(\mathbb{F}_q)$ and $q = \#E(\mathbb{F}_p)$, and fall into one of three categories:

- (i) *Both E and \hat{E} are pairing-friendly.* These cycles were first proposed for use in scalable pairing-based SNARKs by Ben-Sasson, Chiesa, Tromer and Virza [5] and use instances of the only known cycle of ordinary pairing-friendly curves coming from the Miyaji-Nakabayashi-Takano (MNT) construction [33,29]. For example, the Mina protocol [20] is built on top of a cycle of MNT curves.
- (ii) *One of E and \hat{E} is pairing-friendly.* These *hybrid* cycles can be readily constructed by taking any pairing-friendly curve E/\mathbb{F}_p of prime order q , e.g. a Barreto-Naehrig (BN) curve [2], and partnering it with the non-pairing-friendly³ curve \hat{E}/\mathbb{F}_q of prime order p , which is not only guaranteed to exist, but necessarily has the same CM discriminant as E (c.f. [38]). Examples of hybrid cycles found in the wild are Hopwood’s Pluto/Eris cycle [28], Meckler’s BN382 cycle [32], and Williamson’s BN254/Grumpkin cycle [46].

³ The only known exception here is when E is a particular instance of an MNT curve, in which case \hat{E} can also be pairing-friendly and the cycle would then fall into category (i).

- (iii) *Neither E nor \hat{E} are pairing-friendly.* Proof systems that avoid the use of pairings altogether can still exploit the recursive composition afforded by a cycle; the Bulletproofs [11] system is one such popular example. Non-pairing-friendly cycles found in the wild include Poelstra’s secp/secq cycle [34], Bowe, Grigg and Hopwood’s Tweedledee/Tweedledum cycle [9], and Hopwood’s Pasta cycle [27].

The 2-cycles in cases (ii) and (iii) (where at least one of the curves is not pairing-friendly) are much easier to find and construct than the 2-cycles in case (i). The instantiations falling into cases (ii) and (iii) that are cited above involve curves whose underlying field sizes are either optimally small, or else very close to it. On the other hand, the extra restrictions imposed by insisting that both E and \hat{E} are pairing-friendly make constructing the cycles in case (i) notoriously difficult; beyond the MNT construction, there remain no known methods of constructing cycles of ordinary pairing-friendly elliptic curves [12,3]. The two curves in the MNT cycle have embedding degrees that are too small to balance the elliptic curve discrete logarithm problem (ECDLP) and the finite field discrete logarithm problem (DLP) at moderate levels of security. Thus, instantiating these cycles forces E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q to be defined over finite fields whose sizes are *much* larger than the sizes of fields that non-pairing-friendly curves could be defined over. For example, the Mina protocol uses the MNT753 curves E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q with $p \approx q \approx 2^{753}$ in order to target the 112-bit security level. This is because E and \hat{E} have embedding degrees 4 and 6, meaning that both of the ECDLP’s in $E(\mathbb{F}_p)$ and $\hat{E}(\mathbb{F}_q)$ and both of the DLP’s in $\mathbb{F}_{p^4}^\times$ and $\mathbb{F}_{q^6}^\times$ need to be secure. In this instance, the size of $\mathbb{F}_{p^4}^\times$ is around 2^{3072} , which is just enough to meet the 112-bit security level. If, however, neither E/\mathbb{F}_p nor \hat{E}/\mathbb{F}_q are required to be pairing-friendly, then we could work with $p \approx q \approx 2^{224}$ to achieve 112-bit ECDLP security. It follows that group operations like (multi-)scalar multiplications would be much more efficient on these smaller curves than on the MNT753 cycle. In addition, all of the circuit *arithmetisation* that occurs inside SNARKs takes place over \mathbb{F}_p and/or \mathbb{F}_q , so pairing-friendly cycles decrease the performance of the field arithmetic necessary to work with the circuit as well.

The reason many SNARK instantiations still opt for cycles where one or both of E and \hat{E} are pairing-friendly is that pairing-based proof systems offer a number of advantages over non-pairing-based proof systems. Groth’s SNARK [23], often dubbed **Groth16**, is perhaps the most ubiquitous pairing-based proof system in both the academic literature and in practical SNARK implementations. The reason is that the proof sizes in Groth’s construction are *constant*, i.e. are independent of the size of the witness/statement they are proving. On the other hand, pairing-free proof systems like Bulletproofs [11] are currently (at best) logarithmic in the size of the witness. Depending on the target application, the resulting proof sizes in these protocols might be acceptable, but in terms of the succinctness property that is fundamental to the real-world appeal of SNARKs, **Groth16** and its pairing-based variants remain unrivalled by their non-pairing-based counterparts.

The absence of optimal pairing-friendly *cycles* has led many SNARK designers to instead opt for *chains* of pairing-friendly curves [15,17,1]. Such chains allow for composition of pairing-based proofs, but the number of times proofs can be composed recursively is strictly less than the number of curves in the chain. For example, most chains found in the literature are 2-chains [16,17], which only allow for one round of proof composition. Again, 2-chains may suffice for specific target applications, but such applications are a far cry from the sorts of scalable pairing-based proof systems that unbounded recursive composition would support [5].

In this paper we give constructions of pairing-friendly (2, 2)-lollipops⁴, which are 2-chains that are connected to 2-cycles. In Figure 1 we depict the differences between these three possibilities: a ‘•’ represents a pairing-friendly curve, and (here and throughout the paper) an arrow \rightarrow pointing from A to B means B is pairing-friendly with respect to the (characteristic of the) field of definition of A .

⁴ In general, an (m, n) -lollipop would have m curves in the “stick” and n curves in the cycle (one curve is common to both the stick and the cycle), but we will focus on (2, 2)-lollipops in this paper.

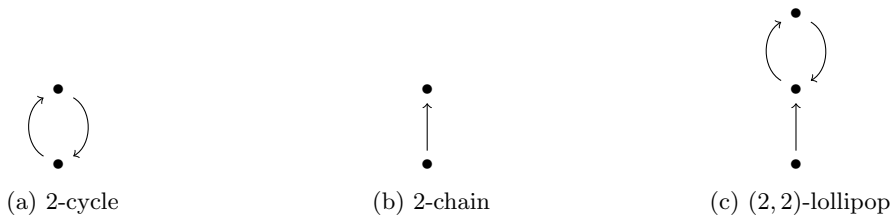


Fig. 1. A 2-cycle, a 2-chain, and their combination: a (2,2)-lollipop. Further explanation in text.

Pairing-friendly lollipops and the SNARK Challenge. The idea of constructing lollipops of pairing-friendly curves is not novel. In 2019, as part of the “Coda+Dekrypt SNARK challenge” [13], the founders of the Coda protocol [6] (now called the Mina protocol⁵) offered a cash prize of \$20,000 USD for the highest quality submission of pairing-friendly lollipops. In defining the challenge and submission requirements, Meckler [31] started with the fact that the Coda Protocol is built on top of the pairing-friendly cycle of oversized MNT753 curves discussed above with $p \approx q \approx 2^{753}$, before pointing out that their “*inefficiency affects not only the SNARK prover, but leaks into the rest of the application as well. The reason is that our SNARK needs to certify all cryptographic computations in Coda (signatures, hashes, etc.) and so those primitives need to be efficiently described using \mathbb{F}_r arithmetic (where r is the order of one of the curves in our cycle). But r is large (about 753 bits), which means outside of the SNARK, our cryptographic operations are a lot slower than they could be.*” In order to address this inefficiency, Meckler envisioned being able to construct a chain of pairing-friendly curves that eventually leads into the MNT753 cycle, but one for which the prime order subgroup size of the first pairing-friendly curve in the chain is much closer to optimal, e.g. $r \approx 2^{256}$. The Coda+Dekrypt SNARK challenge has since expired, but to the best of our knowledge the call for lollipops went unanswered. Subsequently, the Mina protocol still suffers from the same inefficiencies described above.

This work. We construct lollipops of pairing-friendly curves that allow unbounded recursive pairing-based proof composition and simultaneously allow the initial SNARK circuit arithmetic to be performed over a field whose size is either optimal with respect to a given security level, or much closer to it. We illustrate the idea below by comparing the MNT298 cycle from the original work proposing the use of pairing-friendly cycles [5] to Example 1 of our construction: lollipop-305-158.

The MNT298 instance targets the 80-bit security level by working with a cycle of ordinary MNT curves defined over the 298-bit primes p and q . The corresponding DLP instances lie in $\mathbb{F}_{p^4}^\times$ and $\mathbb{F}_{q^6}^\times$, which lie in the fields of size 1192 and 1788 bits, respectively. The main reason this MNT298 cycle is suboptimal at the 80-bit security level is that the primes p and q are almost twice as large as the sizes of primes that non-pairing-friendly curves could be defined over at this security level.

As an alternative, consider lollipop-305-158. It contains a 2-cycle of supersingular curves, $\hat{\mathcal{E}}_{305}^2/\mathbb{F}_p$ and $\mathcal{E}_{305}^3/\mathbb{F}_q$, where p and q are two 305-bit primes. Just like the MNT construction above, pairings map ECDLP instances to instances of the DLP in $\mathbb{F}_{p^4}^\times$ and $\mathbb{F}_{q^6}^\times$, which are fields of size 1220 and 1830 bits, respectively. In this case, however, there is a third curve, E_{305}/\mathbb{F}_p , which is ordinary; it is pairing-friendly with respect to a 158-bit prime r , and the order- r Weil pairing on E_{305} maps into the same multiplicative group (i.e. $\mathbb{F}_{p^4}^\times$) as the order- q Weil pairing on $\hat{\mathcal{E}}_{305}^2$. Moreover, the complexity of the best attack against the ECDLP in $E_{305}(\mathbb{F}_p)[r]$ closely matches the 80 bits of security offered by the corresponding DLP’s in $\mathbb{F}_{p^4}^\times$. On average, Pollard’s rho [35] algorithm solves the ECDLP in $E_{305}(\mathbb{F}_p)[r]$ in $\sqrt{\frac{\pi \cdot r}{4}} \approx 2^{79}$ elliptic curve group operations.

In Figure 2 we illustrate the difference between the original MNT298 cycle and lollipop-305-158. Here and throughout the paper, the notations \mathcal{E} and E correspond to supersingular and ordinary

⁵ Mina [20] uses recursive composition of pairing-based proofs so that “Users only need to check a singular, recursive ‘Proof of Everything’”.

pairing-friendly curves, respectively; the notation \mathbf{E} corresponds to a non-pairing-friendly curve. Subscripts denote the bitlength of the characteristic of the field of definition. Numeric superscripts denote the degree of the extension in the field of definition, if greater than 1. The superscripts ‘W’ and ‘Ed’ denote prime order Weierstrass and composite order (twisted) Edwards curves, respectively.

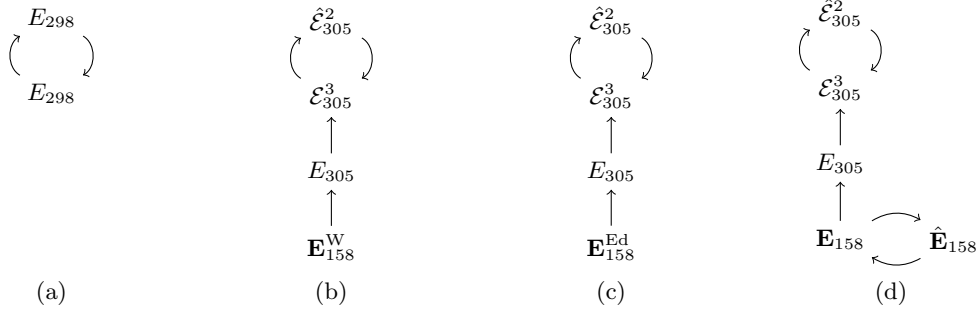


Fig. 2. MNT298 vs. (the three incarnations of) lolipop-305-158. Further explanation in text.

For every lolipop presented in this paper, we construct three options of non-pairing-friendly curves to define over \mathbb{F}_r and attach to the *stick*. This is the reason there are three incarnations of lolipop-305-158 in Figure 2; all have the same pairing-friendly-lollipop but differ only in the definition of the non-pairing-friendly curve \mathbf{E} at the bottom. One goal of this paper is to present lollipops that are ready to use out-of-the-box, so we mimicked some of the constructions that have appeared in the literature, in particular for the scalar field of Bowe’s BLS12-381 curve [8]. The curve \mathbf{E}_{158}^W is a short Weierstrass curve of the form $y^2 = x^3 - 3x + b$, where $b = 7032$ is minimal such that \mathbf{E}_{158}^W and its quadratic twist have prime order. The curve $\mathbf{E}_{158}^{\text{Ed}}$ is a twisted Edwards curve of the form $-x^2 + y^2 = 1 + dx^2y^2$, where $d = 7821$ is minimal such that the cofactors of the $\mathbf{E}_{158}^{\text{Ed}}$ and its quadratic twist are optimally small, i.e. 8 and 4, respectively; this construction is analogous to Bowe and Hopwood’s Jubjub curve⁶ defined over the BLS-381 scalar field. Finally, the curves \mathbf{E}_{158} and $\hat{\mathbf{E}}_{158}$ are a non-pairing-friendly cycle of curves. They both have complex multiplication (CM) discriminant $D = -3$, and can be written as $\mathbf{E}_{158}/\mathbb{F}_r: y^2 = x^3 + 2$ and $\hat{\mathbf{E}}_{158}/\mathbb{F}_{\hat{r}}: y^2 = x^3 + 2$, where \hat{r} is also a 158-bit prime such that $\#\mathbf{E}_{158}(\mathbb{F}_r) = \hat{r}$ and $\#\hat{\mathbf{E}}_{158}(\mathbb{F}_{\hat{r}}) = r$. Both of these curves come equipped with an efficient endomorphism that can accelerate (multi-)scalar multiplications, analogous to the secp/secq [34] and Pasta [27] cycles. Non-pairing-based SNARKs like Bulletproofs [11] could be instantiated using the \mathbf{E}_{158} and $\hat{\mathbf{E}}_{158}$ cycle, similar to their instantiations on the secp/secq and Pasta cycles, but the key difference is that the lollipop of pairing-friendly curves connected to \mathbf{E}_{158} and $\hat{\mathbf{E}}_{158}$ allows the succinct Groth16-style proofs to combine, compose and check the proofs on the non-pairing-friendly cycle.

The main technical ingredient we use to construct the lollipops in this paper is a new way of computing pairings over parameters corresponding to those MNT cycles for which the discriminant is too large to construct the MNT curves explicitly. The MNT cycle works with the primes $p = x^2 - x + 1$ and $q = x^2 + 1$ for some $x \in \mathbb{Z}$, but the MNT curves E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q can only be constructed if the CM discriminant D , the squarefree part of $3x^2 - 2x + 3$, is *small* enough (say, less than 10^{17}). Of all the x values that correspond to p and q being prime, those that also correspond to a small discriminant D are rather rare. As x grows large, the probability that $3x^2 - 2x + 3$ happens to be divisible by a square that is large enough to make $D < 10^{17}$ becomes exponentially small. This is why MNT curves are constructed using special values of x , those which are found as the solutions of Pell equations. Every candidate value for D gives rise to a new Pell equation, the solution of which can be used as a candidate x value; if such an x also corresponds to p and q being prime, then the CM method can construct the MNT curves E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q . The main

⁶ See <https://github.com/zkcrypto/jubjub>

problem, however, is that the solutions to Pell equations are extremely large in general. The chance of finding a solution x that also happens to be the right size at a target security level is unlikely.

In Section 3 we show how cycles can be instantiated for *all* $x \in \mathbb{Z}$ that give $p = x^2 - x + 1$ and $q = x^2 + 1$ as primes. Rather than constructing the MNT cycle of ordinary curves E and \hat{E} over \mathbb{F}_p and \mathbb{F}_q , we construct a cycle of supersingular curves \mathcal{E} and $\hat{\mathcal{E}}$ over extension fields that are smaller than \mathbb{F}_{p^4} and \mathbb{F}_{q^6} , e.g. the curves $\hat{\mathcal{E}}_{305}^2/\mathbb{F}_p$ and $\mathcal{E}_{305}^3/\mathbb{F}_q$ defined over \mathbb{F}_{p^2} and \mathbb{F}_{q^3} in Figure 2. Elements of the pairing groups \mathbb{G}_1 and \mathbb{G}_2 are now defined over extension fields, so group and pairing arithmetic will both be significantly slower than in the ordinary case (more on this in a moment). However, the target groups \mathbb{G}_T in both cases remain unchanged (i.e. are still $\mathbb{F}_{p^4}^\times$ and $\mathbb{F}_{q^6}^\times$), so the pairings themselves will not be too much slower in our supersingular instantiation. The reason it is crucial to untether cycles of curves from their CM discriminants is because the curves in the cycle and the curves we attach to them (in the stick of the lollipop) necessarily have different CM discriminants. As we show in Section 4, if we want to use the CM method to construct the ordinary MNT curves in a cycle, we will not be able to construct the curves in the stick, and vice versa. If we instead instantiate the cycle using supersingular curves as above, then this allows us to focus on finding parametrisations where the ordinary pairing-friendly curve(s) in the stick of the lollipop correspond to a CM discriminant that is small enough to be constructed explicitly.

The reward for slower operations in the supersingular cycle is the introduction of the smaller finite field \mathbb{F}_r , which means the initial circuit arithmetisation and/or the non-pairing-friendly proof system will now be much more efficient. This trade-off will be favourable in scenarios like the one Meckler described, where we can “*perform the bulk of the computation in our proofs in \mathbb{F}_r [...], and then just use the big curves in the cycle for composition and combining proofs. That is, just use the cycle for the relatively small computation of checking other verifiers.*”

In Section 5 we present the 18 lollipops we found using the method we describe in Section 4. These lollipops offer between 80 and 128 bits of security and, just like lollipop-305-158 above, all come with three options for the non-pairing-friendly curves defined over \mathbb{F}_r ; a twist-secure prime order Weierstrass curve, a twist-secure composite order (twisted) Edwards curve, and a low discriminant cycle of ordinary elliptic curves.

Limitations and drawbacks. There are three main downsides of the construction proposed in this paper: finding instances beyond the 128-bit security level, finding instances where r has large *2-adicity*, and the drawbacks that arise within the cycle.

High-security levels. All of the lollipops we present in this paper are labelled as lollipop-X-Y, where X is the bitlength of the primes p and q in the cycle and Y is the bitlength of r , the characteristic of the smaller finite field where the arithmetisation takes place. The method we used to construct these lollipops in Section 4 involves factoring many numbers of around X bits until one of them contains a Y-bit prime factor with $Y \approx 2\lambda$, where λ is the target security level. For example, lollipop-305-158 described above was found by factoring many numbers of around 300 bits until one of them contained a factor close to 160 bits, in order to target the 80-bit security level.

The size of X becomes larger at higher security levels, meaning that larger factorisations are required. For example, recall that the MNT753 cycle used in Mina has $p \approx q \approx 2^{753}$ in order to target 112-bit security. Finding an optimal lollipop at this level would mean factoring X-bit numbers, where $X \approx 753$, until one happens to contain a Y-bit prime factor, where $Y \approx 224$. Given that the current integer factorisation record is for an 829-bit RSA modulus, the computation time for which was roughly 2700 core years⁷, factoring many numbers whose sizes are close to this record becomes the main obstacle in the way of finding better lollipop parameters. On the one hand, the numbers we are trying to factor are not RSA moduli, i.e. are not necessarily the product of two

⁷ This record is due to Boudot, Gaudry, Guillevic, Heninger, Thome and Zimmerman – see <https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;dc42ccd1.2002>.

similarly sized primes, meaning we can sometimes get lucky and find full factorisations in a matter of minutes or hours. On the other hand, many of the candidate X -bit numbers with $X > 700$ had to be abandoned due to the factorisations not terminating over several days. Between the 80- and 112-bit security levels, i.e. with $296 < X < 753$, we had enough candidate X -bit numbers to get lucky with faster-than-expected factorisations that resulted in Y -bit prime factors whose sizes were close to optimal. At the 128-bit security level, however, the only meaningful example we found had $X = 956$ and $Y = 451$; unfortunately, this is still much larger than the optimal size of $Y \approx 256$. At the 192- and 256-bit security levels, the numbers we would be required to factor are thousands of bits long (see Section 4), so finding lollipops using our approach becomes prohibitively expensive.

Large 2-adicity. In the context of SNARKs, arithmetic in the field \mathbb{F}_r is more efficient if $r - 1$ is divisible by a large power of 2 [4] (see also [5, Appendix C.2]); this power of 2 is often referred to as the *2-adicity* of the curve E/\mathbb{F}_r . While some curves can be constructed to have very large 2-adicity, cycles of MNT curves are found via the solution of Pell equations, and such solutions cannot be tailored to have large 2-adicity. As was pointed out in the original work proposing MNT cycles for SNARKs, large 2-adicity becomes a rather restrictive property “because it requires seeing enough curves until, by sheer statistics, one finds [an instance] with a high-enough [2-adicity]” [5, §3.2]. In the case of the lollipops in this paper, we are juggling even more restrictions, and these greatly reduce the number of instances we can hope to sift through in the search for one with high 2-adicity. It turns out that none of the 18 lollipops we found were lucky enough to come equipped with an appreciably large 2-adicity.

Nevertheless, we point out that the much smaller sizes of \mathbb{F}_r afforded by lollipops is likely to give rise to faster circuit arithmetisation than large 2-adicities would afford in the overblown fields \mathbb{F}_p and \mathbb{F}_q . Moreover, as we discuss briefly in Remark 1, the workaround that we describe in Section 3 can be used independently of lollipop constructions. This allows one to take any large primes $p = x^2 - x + 1$ and $q = x^2 + 1$ and define a useful supersingular cycle over fields of those characteristics. In particular, one can take values of x where $2^\ell \mid x$ and (so long as p and q are prime) immediately get a highly 2-adic pairing-friendly cycle for which $2^\ell \mid p - 1$ and $2^{2\ell} \mid q - 1$.

The cycle vs. stick trade-off. As mentioned above, our construction of lollipops trades much faster circuit arithmetisation in the field \mathbb{F}_r with inefficiencies in the cycle. The pairing groups corresponding to the two curves in the cycle are now defined over extension fields of \mathbb{F}_p and \mathbb{F}_q , which will not only slow down the group operations, but will also affect the sizes of any proofs computed in the cycle. For example, proofs in the **Groth16** system are two elements of \mathbb{G}_1 and one element of \mathbb{G}_2 for asymmetric pairings, or three elements of \mathbb{G} for symmetric pairings [23, Table 2]. Now, Proposition 1 replaces E/\mathbb{F}_p with $\mathcal{E}/\mathbb{F}_{p^2}$ and \hat{E}/\mathbb{F}_q with $\hat{\mathcal{E}}/\mathbb{F}_{q^3}$; in both cases the size of the larger \mathbb{G}_2 elements will remain unchanged, but the size of the smaller \mathbb{G}_1 elements increases by a factor of 2 and 3, respectively. Thus, proofs that were computed on E/\mathbb{F}_p will be a factor 1.5x larger on $\mathcal{E}/\mathbb{F}_{p^2}$, and proofs that were computed on \hat{E}/\mathbb{F}_q will be a factor 1.8x larger on $\hat{\mathcal{E}}/\mathbb{F}_{q^3}$.

In terms of efficiency, we reiterate that the pairings themselves will not be too much (i.e. orders of magnitude) worse in the supersingular scenario; the size of the target group dominates the complexity and the target groups are identical in the asymmetric and symmetric cases – see Figure 3. Since this work is solely constructive, we do not attempt to quantify these trade-offs any further, but instead leave the computational aspects of the supersingular cycle for future work.

Organisation. In Section 2 we give a brief background on pairing-friendly curves and the CM method. In Section 3 we show how to define a cycle of supersingular curves over fields of characteristic $p = x^2 - x + 1$ and $q = x^2 + 1$, which lays the foundation for the lollipop construction we present in Section 4. In Section 5 we present the 18 lollipop examples we found using this construction. We conclude the paper in Section 6.

2 Preliminaries

For $n \in \mathbb{Z}_{>0}$, the n -th *cyclotomic polynomial* $\Phi_n(x)$ is given by

$$\Phi_n(x) = \frac{x^n - 1}{\prod_{\substack{d|n \\ d < n}} \Phi_d(x)}.$$

Throughout this paper we will commonly make use of the following lemma.

Lemma 1 ([44, Lemma 2.9]). *Let p be a prime and $n, a \in \mathbb{Z}_{>0}$ such that $p \nmid na$. Then $\text{ord}_p(a) = n$ iff $p \mid \Phi_n(a)$.*

Pairing-friendly elliptic curves. For an extensive survey on pairing-friendly curves, we refer to [21]. Let E/\mathbb{F}_q be an elliptic curve and r be a large prime such that $r \mid \#E(\mathbb{F}_q)$. Then E is said to have *embedding degree* k (with respect to r) if $\text{ord}_r(q) = k$. Since $r \gg k$, we can use Lemma 1 to deduce that E has embedding degree k iff $r \mid \Phi_k(q)$. If k is *small* enough, e.g. $k \leq 50$, then E is said to be *pairing-friendly*. All pairing-friendly curves in this paper have $k \leq 6$.

The CM method. Hasse's theorem [26] states that the number of points on an elliptic curve E/\mathbb{F}_q is $\#E(\mathbb{F}_q) = q + 1 - t$, where the *trace of Frobenius* t is bounded by $|t| \leq 2\sqrt{q}$. On input of a given t within the Hasse interval, the way we construct a curve (i.e. compute its coefficients) with $q + 1 - t$ rational points is via the *complex multiplication* (CM) method [19, §4]. Write

$$DV^2 = 4q - t^2, \tag{1}$$

where $D, V \in \mathbb{Z}$, and where D is *squarefree*. The CM method finds E by computing the *Hilbert class polynomial* $H_D(X) \in \mathbb{F}_q[X]$, the roots of which correspond to j -invariants of elliptic curves whose CM *discriminant* is D . If $j \in \mathbb{F}_q$ is such that $H_D(j) = 0$, then we can write E as (a quadratic twist of) $E: y^2 = x^3 + ax - a$, where $a = -27j/(4(j - 1728))$; the only exceptions are $j = 0$, in which case we take E as (a quartic twist of) $E: y^2 = x^3 + 1$, and $j = 1728$, in which case we take E as (a sextic twist of) $E: y^2 = x^3 + x$. For details concerning twists, see [37, Proposition X.5.4].

When q is a prime, elliptic curves *exist* for all t with $|t| \leq 2\sqrt{q}$. When $q = p^n$ for a prime p , there are fewer than n values of t that do not correspond to an elliptic curve [36, Theorem 4.2]. Either way, when q is of cryptographic size, only a tiny fraction of these t values correspond to curves that can actually be constructed via the CM method. This is because the computation of the Hilbert class polynomial becomes infeasible if D is large. The time complexity of the best known algorithm for computing $H_D(X)$ is in $\tilde{O}(|D|)$ [41], and current record CM computations⁸ have $|D| < 10^{17}$. For q of cryptographic size, most traces t in the Hasse interval will correspond to a discriminant D in (1) that far exceeds those in these record computations. In the next subsection we will show how Pell equations can be used to find the special values of t that *do* correspond to sizes of D 's that make the CM method feasible.

Finding MNT cycles with small CM discriminants. Recall from Section 1 that 2-cycles of MNT curves are defined by taking large primes

$$p = \phi_6(x) = x^2 - x + 1 \quad \text{and} \quad q = \phi_4(x) = x^2 + 1 \tag{2}$$

for some $x \in \mathbb{Z}$. The MNT curve E/\mathbb{F}_p has trace $t_E = -x + 1$, which gives $\#E(\mathbb{F}_p) = p + 1 - t_E = x^2 + 1 = q$, while the MNT curve \hat{E}/\mathbb{F}_q has trace $t_{\hat{E}} = x + 1$, which gives $\#\hat{E}(\mathbb{F}_q) = q + 1 - t_{\hat{E}} = x^2 - x + 1 = p$. It follows from (2) that $\text{ord}_q(p) = 4$ and $\text{ord}_p(q) = 6$, so E has embedding degree 4 and \hat{E} has embedding degree 6. In both cases, substitution into (1) yields

$$DV^2 = 3x^2 - 2x + 3. \tag{3}$$

⁸ See <https://math.mit.edu/~drew/CMRecords.html>.

Over all instances of $x \in \mathbb{Z}$ that correspond to p and q in (2) being prime, we expect that the vast majority give rise to $D \approx p \approx q$. Put another way, if x is chosen at random from a large interval, we cannot expect the value of $3x^2 - 2x + 3$ from (3) to contain a large square factor V , and thus $D = O(x^2)$ in most cases.

The way MNT curves are constructed is to instead compute the few values of x that correspond to small values of D . Putting $U = 3x - 1$ into (3) yields the *generalised Pell equation*

$$U^2 - 3DV^2 = -8. \quad (4)$$

Solving (4) for a small, fixed value of D yields the pair $(U, V) \in \mathbb{Z}^2$. If $U \in 2 + 3\mathbb{Z}$, then we can take $x = (U + 1)/3$ and proceed by checking if $p = x^2 - x + 1$ and $q = x^2 + 1$ are prime. If they are, then we can use the CM method to construct the MNT curves E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q .

The reason we must solve many (e.g. millions of) Pell equations to find suitable MNT parameters is that the solutions (U, V) to (4) are unlikely to be the size we want at a given security level. Even in the cases where the solutions are of the right size, it is unlikely that both $U \in 2 + 3\mathbb{Z}$ and the corresponding values of p and q are both prime.

In the next section we show that cycles can still be constructed over fields of characteristic p and q regardless of the CM discriminant D .

3 Instantiating supersingular cycles when the CM method for ordinary MNT curves is infeasible

Recall from (2) that the MNT cycle has

$$p = x^2 - x + 1 \quad \text{and} \quad q = x^2 + 1,$$

and is such that

$$\text{ord}_q(p) = 6 \quad \text{and} \quad \text{ord}_p(q) = 4. \quad (5)$$

The ordinary curves E/\mathbb{F}_p and \hat{E}/\mathbb{F}_q are such that $\#E(\mathbb{F}_p) = q$ and $\#\hat{E}(\mathbb{F}_q) = p$; they can only be constructed by the CM method for those values of x which give rise to a small enough D in (3). In this section we show an alternative cycle construction over $\bar{\mathbb{F}}_p$ and $\bar{\mathbb{F}}_q$ that uses supersingular curves. These supersingular curves have j -invariants in \mathbb{F}_p and \mathbb{F}_q , but the disadvantage of invoking the supersingular construction is that the respective q - and p -torsion points (that define the cycle) only become rational over extension fields. The crucial advantage, however, is that this construction works for *all* values of x that give rise to p and q as primes. As we show in the next section, this means rather than solving Pell equations to ensure the MNT curves in the cycle have low discriminant, we can instead solve the Pell equations that ensure the curve in the stick of the lollipop has a low enough discriminant for the CM method.

Cycles of supersingular elliptic curves were recently given in [14], but the cycles we introduce here have an important difference. Those in [14] set $q \equiv 1 \pmod p$ so that any curve defined over \mathbb{F}_q is forced to have embedding degree 1 with respect to p . Furthermore, the construction in [14] also forced the bitlength of q to always be (at least) twice the bitlength of p . In what follows we instead exploit the relationship between the special p and q in the MNT construction to avoid these impositions; our supersingular cycles have $p \approx q$ and allow for both curves to have embedding degrees greater than 1, as we see in Proposition 1.

We start by specialising the definition of a pairing-friendly n -cycle (see Section 4) to the case of $n = 2$.

Definition 1 (Pairing-friendly 2-cycle). *We say that two elliptic curves $\mathcal{E}/\mathbb{F}_{p^u}$ and $\hat{\mathcal{E}}/\mathbb{F}_{q^v}$ are a pairing-friendly 2-cycle, denoted*

$$\mathcal{E} \rightleftharpoons \hat{\mathcal{E}},$$

if

- (i) $p \mid \#\hat{\mathcal{E}}(\mathbb{F}_{q^v})$;
- (ii) $q \mid \#\mathcal{E}(\mathbb{F}_{p^u})$;
- (iii) \mathcal{E} is pairing-friendly with respect to q ; and
- (iv) $\hat{\mathcal{E}}$ is pairing-friendly with respect to p .

In Figure 3 giving we give a depiction of the constructions that follow in Proposition 1 and Proposition 2. The only difference in the two constructions is the field of definition of the curve \mathcal{E} , which is \mathbb{F}_{p^2} in Proposition 1 and \mathbb{F}_{p^4} in Proposition 2. This difference arises based on the values of $p \bmod 3$ and/or $p \bmod 4$, because Waterhouse's theorem [45, Theorem 4.1] gives precise conditions on the (non-)existence of various supersingular curves that depends on these values. With $p = x^2 - x + 1$ and $q = x^2 + 1$, it turns out that (for all $x \neq 2$) there are only 4 classes of $x \in \mathbb{Z}/12\mathbb{Z}$ that can give both p and q prime; odd x gives even q and $x \in \{2 + 12\mathbb{Z}, 8 + 12\mathbb{Z}\}$ gives p a multiple of 3.⁹ This is why both of the propositions below start with the statement of the classes of $x \in \mathbb{Z}/12\mathbb{Z}$.

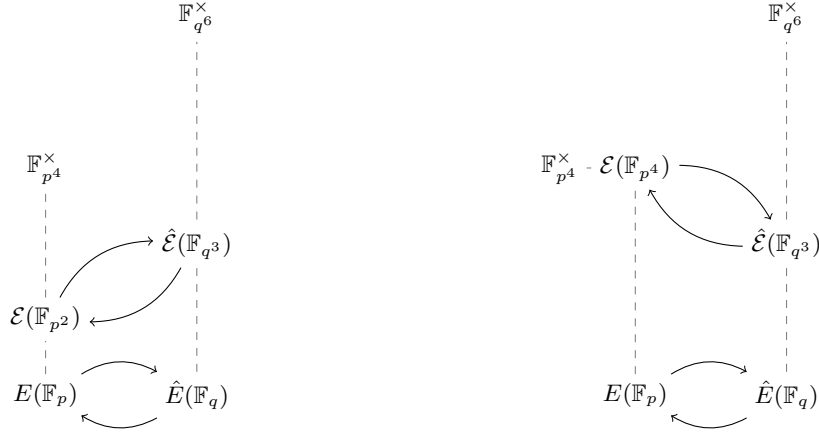


Fig. 3. Ordinary MNT cycle vs. supersingular cycles: Proposition 1 (left) and Proposition 2 (right). E and \hat{E} are the ordinary MNT curves, while \mathcal{E} and $\hat{\mathcal{E}}$ are the supersingular curves. The target groups $\mathbb{F}_{p^4}^\times$ and $\mathbb{F}_{q^6}^\times$ are the same for both constructions. E and \hat{E} have embedding degrees 4 and 6. $\hat{\mathcal{E}}$ has embedding degree 2 with respect to p . \mathcal{E} either has embedding degree 2 (left) or 1 (right) with respect to q .

Waterhouse's theorem [45, Theorem 4.1] says there is precisely one isogeny class of supersingular curves over a given odd degree extension of \mathbb{F}_q , i.e. the class with trace $t = 0$; it follows that this class necessarily contains the lifts of those curves in the unique isogeny class with $t = 0$ defined over \mathbb{F}_q . In the statements of Propositions 1 and 2, the curve $\hat{\mathcal{E}}$ is written as $\hat{\mathcal{E}}/\mathbb{F}_{q^3}$, even though we can actually write it as a curve that is defined over \mathbb{F}_q . The reason we write it this way in the propositions is to align with Definition 1 and to make it clear that we only obtain a cycle when considering the group of rational points in \mathbb{F}_{q^3} , which is where we find p -torsion.

Proposition 1. *Let $x \in \{6 + 12\mathbb{Z}, 10 + 12\mathbb{Z}\}$ be such that $p = x^2 - x + 1$ and $q = x^2 + 1$ are prime. Then there exist two supersingular curves*

$$\mathcal{E}/\mathbb{F}_{p^2} \quad \text{and} \quad \hat{\mathcal{E}}/\mathbb{F}_{q^3}$$

such that

$$\mathcal{E} \rightleftharpoons \hat{\mathcal{E}}.$$

⁹ Note that when $x = 2$ we still get a cycle with $p = 3$ and $q = 5$. There is a supersingular curve $\mathcal{E}/\mathbb{F}_{3^2}$ with $j(\mathcal{E}) = 1728$ and $\#\mathcal{E}(\mathbb{F}_{3^2}) = 2q$, and a curve $\hat{\mathcal{E}}/\mathbb{F}_q$ with $j(\hat{\mathcal{E}}) = 0$ and $\#\hat{\mathcal{E}}(\mathbb{F}_5) = 2p$. Both curves have embedding degree 2.

Moreover, \mathcal{E} has embedding degree 2 with respect to q , and $\hat{\mathcal{E}}$ has embedding degree 2 with respect to p .

Proof. We prove each of the requirements in Definition 1. For (i), there is precisely one isogeny class of supersingular curves over \mathbb{F}_{q^3} , all of which have $q^3 + 1$ rational points [45, Theorem 4.1(5)(i)]. It follows that $\#\hat{\mathcal{E}}(\mathbb{F}_{q^3}) = q^3 + 1 = (q + 1)\Phi_6(q)$; together with (5) and Lemma 1, this gives $p \mid \#\hat{\mathcal{E}}(\mathbb{F}_{q^3})$. For (ii), taking $x \in \{6 + 12\mathbb{Z}, 10 + 12\mathbb{Z}\}$ gives $p \equiv 3 \pmod{4}$ and $p \equiv 1 \pmod{3}$, which combined with [45, Theorem 4.1(5)(ii)] says there exists a supersingular curve $\mathcal{E}/\mathbb{F}_{p^2}$ with $\#\mathcal{E}(\mathbb{F}_{p^2}) = p^2 + 1$. Since $p^2 + 1 = \Phi_4(p)$, we again use (5) and Lemma 1 to give $q \mid \#\mathcal{E}(\mathbb{F}_{p^2})$. The embedding degrees follow directly from [22, Theorem IX.20], which also proves (iii) and (iv). \square

Proposition 2. *Let $x \in \{12\mathbb{Z}, 4 + 12\mathbb{Z}\}$ be such that $p = x^2 - x + 1$ and $q = x^2 + 1$ are prime. Then there exist two supersingular curves*

$$\mathcal{E}/\mathbb{F}_{p^4} \quad \text{and} \quad \hat{\mathcal{E}}/\mathbb{F}_{q^3}$$

such that

$$\mathcal{E} \cong \hat{\mathcal{E}}.$$

Moreover, \mathcal{E} has embedding degree 1 with respect to q , and $\hat{\mathcal{E}}$ has embedding degree 2 with respect to p .

Proof. We again prove each of the requirements in Definition 1. The proof of (i) is identical to that of Proposition 1. For (ii), taking $x \in \{4 + 12\mathbb{Z}, 12\mathbb{Z}\}$ gives $p \equiv 1 \pmod{4}$ and $p \equiv 1 \pmod{3}$, which combined with [45, Theorem 4.1(2)] says there exists a supersingular curve $\mathcal{E}/\mathbb{F}_{p^4}$ with trace $t = -2p^2$, i.e. with $\#\mathcal{E}(\mathbb{F}_{p^4}) = p^4 + 2p^2 + 1$. Since $p^4 + 2p^2 + 1 = (\Phi_4(p))^2$, we use (5) and Lemma 1 to give $q \mid \#\mathcal{E}(\mathbb{F}_{p^4})$. The embedding degrees again follow directly from [22, Theorem IX.20], which also proves (iii) and (iv). \square

Security. We point out that there is no known difference in the security picture of our supersingular cycles and the underlying MNT cycles. Both cycles can use pairings to map ECDLP instances to the same order- p subgroup of $\mathbb{F}_{q^6}^\times$ and/or the same order- q subgroup of $\mathbb{F}_{p^4}^\times$, so the finite field DLP's are identical in both cases. Due to the large fields in the supersingular scenario, there may be a small concrete difference in solving the ECDLP's directly via Pollard's rho algorithm, but a real-world attack would never target the ECDLP's; the small embedding degrees always ensure that the finite field DLPs are the weak point of the cycle itself. The whole point of this paper is to find another pairing-friendly curve, E , whose defining field and embedding degree (and thus DLP complexity) is the same as one of the curves in the cycle, but for which the complexity of solving the ECDLP in $E[r]$ is much closer to the complexity of solving the DLP in $\mathbb{F}_{p^4}^\times$.

Bröker's algorithm. Over a field \mathbb{F}_{p^u} , the algorithm that constructs supersingular curves of a given trace (i.e. group order) is due to Bröker [10]. It starts by constructing a supersingular curve over the ground field, \mathcal{E}/\mathbb{F}_p , and then outputs $\mathcal{E}'/\mathbb{F}_{p^u}$ as an \mathbb{F}_{p^u} -twist of \mathcal{E} . Since twists have the same j -invariant, it follows that the supersingular curves output by Bröker's algorithm¹⁰ always have $j(\mathcal{E}') \in \mathbb{F}_p$, regardless of the field of definition of \mathcal{E}' – see Table 1. Bröker's algorithm also constructs \mathcal{E}/\mathbb{F}_p by finding a root j_0 of the Hilbert class polynomial $H_{\mathcal{D}}(X) \in \mathbb{F}_p[X]$, but in the supersingular case the value of \mathcal{D} that is used is the first prime $\mathcal{D} \equiv 3 \pmod{4}$ where $-\mathcal{D}$ is not a quadratic residue in \mathbb{F}_p .

Remark 1 (High-security, highly 2-adic cycles). One of the drawbacks of the MNT cycle is that it is difficult to construct instances with large 2-adicity, particularly at high-security levels [5,24]. If we momentarily forget about lollipops, then we point out that the constructions in this section can

¹⁰ In general, most supersingular curves have $j \notin \mathbb{F}_p$, but they always come from the same isogeny class as a curve with $j \in \mathbb{F}_p$, which is what Bröker's algorithm outputs.

be used to instantiate a standalone cycle with very large 2-adicity at any security level. We can simply search for values of x where $2^\ell \mid x$ such that $p = x^2 - x + 1$ and $q = x^2 + 1$ are prime, and we immediately get a cycle where $2^\ell \mid p - 1$ and $2^{2\ell} \mid q - 1$. We ran a quick search for the largest ℓ 's corresponding to 8 values of x with $\lceil \log_2 x \rceil \in \{128, 192, 256, 384, 512, 768, 1024, 2024\}$, and found the following values that give p and q primes: $x = 2^{113} \cdot 32123$, $x = 2^{176} \cdot 40335$, $x = 2^{239} \cdot 108445$, $x = 2^{370} \cdot 10431$, $x = 2^{493} \cdot 354617$, $x = 2^{748} \cdot 885549$, $x = 2^{1006} \cdot 226419$, and $x = 2^{2027} \cdot 1526763$. Of course, whatever is gained by the high 2-adicity of the cycle is to be traded off with the “cycle vs. stick” drawbacks of the supersingular construction that we discussed in Section 1.

4 Constructing lollipops

In this section we describe the construction that we used to find the 18 example lollipops in the next section. We start by defining chains and cycles of pairing-friendly curves, pulling together definitions from El Housni and Guillevic [17, §2.3] and Chiesa, Chua and Weidner [12, Definition 7.1].

Definition 2 (Pairing-friendly chain). *An m -chain of pairing-friendly elliptic curves is a list of distinct curves $E_1/\mathbb{F}_{p_1^{d_1}}, \dots, E_m/\mathbb{F}_{p_m^{d_m}}$ with each p_i a large prime, such that $p_i \mid \#E_{i+1}$ and E_{i+1} is pairing-friendly with respect to p_i for $i = 1 \dots m - 1$.*

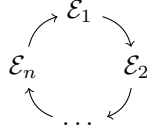
Definition 3 (Pairing-friendly cycle). *An n -cycle of pairing-friendly elliptic curves is a list of distinct curves $E_1/\mathbb{F}_{q_1^{e_1}}, \dots, E_n/\mathbb{F}_{q_n^{e_n}}$ with each q_i a large prime, such that $q_i \mid \#E_{i+1}$ and E_{i+1} is pairing-friendly with respect to q_i for $i = 1 \dots n - 1$, and such that $q_n \mid \#E_1$ and E_1 is pairing-friendly with respect to q_n .*

We can now define lollipops by combining Definition 2 and Definition 3.

Definition 4 (Pairing-friendly lollipop). *An (m,n) -lollipop of pairing-friendly curves is an m -chain*

$$E_1 \rightarrow \dots \rightarrow E_m$$

of pairing-friendly curves, together with an n -cycle



of pairing-friendly curves, such that $\{E_1, \dots, E_m\} \cap \{E_1, \dots, E_n\} = E_m$.

Recall from Section 1 that in this paper we will restrict to the case of $m = n = 2$ and present $(2, 2)$ -lollipops.

Following Definition 4, and with the cycles we have defined in Section 3, our task is to find another pairing-friendly curve defined over \mathbb{F}_p with $p = x^2 - x + 1$ or \mathbb{F}_q with $q = x^2 + 1$. The key to our construction is to observe that

$$\Phi_4(p) = \underbrace{(x^2 + 1)}_q \cdot \underbrace{(x^2 - 2x + 2)}_{N_q} \tag{6}$$

and

$$\Phi_6(q) = \underbrace{(x^2 - x + 1)}_p \cdot \underbrace{(x^2 + x + 1)}_{N_p}. \tag{7}$$

Ideally, we want to find an ordinary curve E/\mathbb{F}_p with $\#E(\mathbb{F}_p) = N_q$, so E immediately has embedding degree 4 with respect to any large prime factor r of N_q . Alternatively, we can also look for a curve E/\mathbb{F}_q with $\#E(\mathbb{F}_q) = N_p$, so that E has embedding degree 6 with respect to any large prime factor r of N_p . As we discussed in Section 2, the feasibility of constructing these curves is related to the corresponding CM equations. We therefore examine the two cases separately.

Case 1: r divides $\#E(\mathbb{F}_p) = N_q$. If $\#E(\mathbb{F}_p) = N_q$, then we must have that $t_E = x$, and hence the CM equation becomes

$$\begin{aligned} DV^2 &= 4p - t^2, \\ &= 4(x^2 - x + 1) - x^2, \\ &= 3x^2 - 4x + 4. \end{aligned} \tag{8}$$

Case 2: r divides $\#E(\mathbb{F}_q) = N_p$. If $\#E(\mathbb{F}_q) = N_p$, then we must have that $t_E = -x + 1$, and hence the CM equation becomes

$$\begin{aligned} DV^2 &= 4q - t^2, \\ &= 4(x^2 + 1) - (-x + 1)^2, \\ &= 3x^2 + 2x + 3. \end{aligned} \tag{9}$$

The next step is to make substitutions that transform these CM equations into generalised Pell equations. Taking $U = 3x - 2$ in (8) or taking $U = 3x + 1$ in (9), we get the generalised Pell equation

$$U^2 - 3DV^2 = -8,$$

which is the same as (4).

We first observe that any solution $U, V \in \mathbb{Z}$ to this generalised Pell equation must have $U \notin 3\mathbb{Z}$. The substitutions $U = 3x - 2$ and $U = 3x + 1$ that gave rise to (8) and (9) reveal that the solutions we are interested in are those with $U \in 1 + 3\mathbb{Z}$. For any such solution, the integer $x = (U - 1)/3$ will satisfy (8) and the integer $x = (U + 2)/3$ will satisfy (9). Given that we need x to be even for $q = x^2 + 1$ to be prime, however, we see that only one of these two (consecutive) integers can be used as a candidate x -value for our lollipop construction. Nevertheless, it is convenient that we only need to write and launch one generalised Pell equation solver to search for solutions to both (8) and (9).

Before giving details of the full lollipop search we implemented, we first modify some results from [29] in order to streamline the discriminants D that we search over.

Lemma 2. *Let D be as in (8) for $x \in 2\mathbb{Z}$. Then $3D \equiv 9 \pmod{24}$.*

Lemma 3. *Let D be as in (9) for $x \in 2\mathbb{Z}$. Then $3D \equiv 3, 6, 18, 27 \pmod{48}$.*

In what follows we set $D' = 3D$. The above lemmas show we only need to search over $1/8$ of the D' values.

The high-level algorithm. We summarise the above discussion by presenting the full lollipop algorithm as follows. On input of a small D' (i.e. $D' < 10^{17}$) as in Lemma 2 or Lemma 3, do the following:

1. Solve the generalised Pell equation $U^2 - D'V^2 = -8$.
 - (a) If $D' \equiv 9 \pmod{24}$ and if $U \equiv 1 \pmod{3}$, then set $x = (U - 1)/3$ and set $N = N_q = x^2 - 2x + 2$ from (6). Otherwise, pick a new D' and start again.
 - (b) If $D' \equiv 3, 6, 18, 27 \pmod{48}$ and if $U \equiv 1 \pmod{3}$, then set $x = (U + 2)/3$ and set $N = N_p = x^2 + x + 1$ from (7). Otherwise, pick a new D' and start again.
2. If $p = x^2 - x + 1$ and $q = x^2 + 1$ are both prime, proceed to Step 3, otherwise pick a new D' and start again.

3. Factor N , and let r be a prime divisor of N that is large enough to meet the requisite security level (e.g. $r \geq 2^{2\lambda}$ for λ -bit security). If no such r exists, pick a new D' and start again.
4. If $N = N_q$, then compute the Hilbert class polynomial $H_D(X) \in \mathbb{F}_p[X]$, otherwise if $N = N_p$ then compute the Hilbert class polynomial $H_D(X) \in \mathbb{F}_q[X]$.
5. Compute a root j_0 of $H_D(X)$ in $\mathbb{F}_p[X]$ (resp. $\mathbb{F}_q[X]$) and then construct the elliptic curve E such that $j(E) = j_0$ (see Section 2).
6. Output the lollipop as

$$E \rightarrow \mathcal{E} \rightleftharpoons \hat{\mathcal{E}},$$

where $\mathcal{E}/\mathbb{F}_{p^2}$ and $\hat{\mathcal{E}}/\mathbb{F}_{q^3}$ are the curves from Proposition 1 if $x \in \{6 + 12\mathbb{Z}, 10 + 12\mathbb{Z}\}$, or else are the curves $\mathcal{E}/\mathbb{F}_{p^4}$ and $\hat{\mathcal{E}}/\mathbb{F}_{q^3}$ from Proposition 2.

We immediately discuss some practical adjustments that can be made in the first four steps.

Step 1. Methods for solving the generalised Pell equation typically use variants of the continued fractions method, which allows us to bound the size of the solutions we want and abort if the solutions grow larger than this bound. We set a very large bound in an effort to try and find high-security instances, but any search for lollipops of a specific size would be accelerated significantly if the bound is lowered to reflect this.

Step 2. For the sake of efficiency, primality testing should be probabilistic within the full search algorithm and probable primes should be confirmed after the lollipops are output.

Step 3. For the sizes of lollipops we are searching for, this step is by far the most cumbersome. Here we are looking for one prime factor r of a certain size, so full factorisations are not necessary, and in many cases it is wise to abort factorisations that do not terminate after a nominal amount of time (see below).

Step 4. For larger sizes of D' , there are other invariants analogous to the j -invariant for which the class polynomials are more efficient to compute and more compact to store, e.g. the Weber f -invariant. These alternative invariants also impose restrictions on D' , but are much preferred if they are compatible with those restrictions in Lemma 2 and Lemma 3. More details can be found in Sutherland's `classpoly` package.¹¹

Details of our search. We now give the more fine-grained details of the search we ran using the algorithm above that produced the lollipops in the next section. We break down the discussion according to the six steps of the search algorithm.

- **Step 1.** We used the `parallel GP` interface of `PARI/GP` [42] running on AMD Ryzen Threadripper PRO 3995WX with 128 GB memory to solve the generalised Pell equation $U^2 - D'V^2 = -8$ using [29, Algorithm 1] and bounding U at 2500 bits, for the following instances:
 - (i) Equation (8) with $D' \in \{48\ell + 3, 48\ell + 6, 48\ell + 18, 48\ell + 27\}$ as per Lemma 3. The equation was solved for $\ell = 0$ to $\ell = 1, 295, 124, 911$, and solutions less than 2500 bits were obtained for 76, 656, 763 of them.
 - (ii) Equation (9) with $D' \in 24\ell + 9$ as per Lemma 2. The equation was solved for $\ell = 0$ to $\ell = 5, 082, 799, 955$, and solutions less than 2500 bits were obtained for 63, 146, 643 of them.
- **Step 2.** We used the `parallel GP` interface of `PARI/GP` running on AMD Ryzen Threadripper PRO 3995WX with 128 GB memory to perform the following searches:

¹¹ See <https://math.mit.edu/~drew/classpoly.html>.

- (i) For the solutions of (8) with $U \equiv 1 \pmod{3}$, we set $x = (U-1)/3$ and tested the primality of $p = x^2 - x + 1$ and $q = x^2 + 1$, finding 3,912 pairs of primes (p, q) .
 - (ii) For the solutions of (9) with $U \equiv 1 \pmod{3}$, we set $x = (U+2)/3$ and tested the primality of $p = x^2 - x + 1$ and $q = x^2 + 1$, finding 4,281 pairs of primes (p, q)
- **Step 3.** We again used the `parallel GP` interface of PARI/GP running on AMD Ryzen Threadripper PRO 3995WX with 128 GB memory to perform the following factorisations (using PARI/GP’s internal factoring algorithms):
 - (i) Of the 3,912 pairs of primes (p, q) corresponding to (8), 337 candidates had both p and q at least 298 bits long.¹² Of these, we were able to successfully factor 203 of them.
 - (ii) Of the 4,281 pairs of primes (p, q) corresponding to (9), 350 candidates had both p and q at least 298 bits long. Of these, we were able to successfully factor 186 of them.
 - **Step 4.** The following computations were run on an Intel Core i7-12800HX with 16 GB memory. Let r be the biggest factor of $N \in \{N_p, N_q\}$ and recall that $N_p \approx N_q \approx p \approx q$. For all prime pairs where (p, q) had $r > 2^{150}$ and where $N \gg r$ (we used a rough rule that $\log(N) \geq 2\log(r)$, but made some exceptions), we computed the Hilbert class polynomial¹³ in $\mathbb{F}_p[x]$ or $\mathbb{F}_q[x]$ using sutherland’s `classpoly` package [40,18].
 - **Step 5.** To find the roots of the polynomial $h_D(X)$ output from Step 4, and then to construct the curve E accordingly, we used `Magma` [7] running on Intel Core i7-12800HX with 16 GB memory.
 - **Step 6.** We found a total of 18 lollipops $E \rightarrow \mathcal{E} \rightleftharpoons \hat{\mathcal{E}}$, 14 of which had the curve E defined over \mathbb{F}_p with embedding degree 4, and the other 4 of which had E defined over \mathbb{F}_q with embedding degree 6. These are all described in detail in the next section.

5 Examples

We now present the 18 example of lollipops that were found in the search described in the previous section. We begin by recalling some notation from Section 1 and set some additional notation that is used in Figure 4 and Table 1:

- The examples are labelled as `lollipop-X-Y`, where X denotes the bitlength of p and q , i.e. the characteristics of the fields of definition of the pairing-friendly curves in the lollipop, and Y denotes the bitlength of r , the characteristic of the non-pairing-friendly curve(s) attached to it. We also use X and Y as subscripts of curves to indicate the size of the underlying field characteristic.
- An arrow $A \rightarrow B$ indicates that the curve B is pairing-friendly with respect to the characteristic of the field of definition of the curve A .
- \mathcal{E}_X and $\hat{\mathcal{E}}_X$ are the supersingular (and thus pairing-friendly) curves in the cycle; \mathcal{E}_X is defined over \mathbb{F}_{p^2} or \mathbb{F}_{p^4} , while $\hat{\mathcal{E}}_X$ is defined¹⁴ over \mathbb{F}_{q^3} .
- The curve E_X , which is such that $\#E_X = h \cdot r$, with r a Y -bit prime. The curve E_X is pairing-friendly with respect to r ; it has embedding degree 4 if it is defined over \mathbb{F}_p and embedding degree 6 if it is defined over \mathbb{F}_q .

¹² We wanted a minimum of 80 bits of security, so chose the original MNT298 size from [5] as the lower bound on the sizes of p and q .

¹³ This is computed using `classpoly d 0 q` where $d = -D$ if $-D \equiv 1 \pmod{4}$ or $d = -4D$ if $-D \equiv 2, 3 \pmod{4}$.

¹⁴ Recall from Section 3 that the curve $\hat{\mathcal{E}}_X$ can be defined over \mathbb{F}_q ; however, we do not find rational p -torsion (and thus a cycle) until we consider the points in $\hat{\mathcal{E}}_X(\mathbb{F}_{q^3})$.

- A bold \mathbf{E} denotes a non-pairing-friendly curve that is attached to the lollipop. Each example lollipop comes with three non-pairing-friendly incarnations: a cycle of two curves $\mathbf{E}_Y/\mathbb{F}_r$ and $\hat{\mathbf{E}}_Y/\mathbb{F}_{\hat{r}}$ with $r = \#\hat{\mathbf{E}}_Y$ and $\hat{r} = \#\mathbf{E}_Y$, a twist-secure short Weierstrass curve $\mathbf{E}_Y^W/\mathbb{F}_r$, and a twist-secure (twisted) Edwards curve $\mathbf{E}_Y^{\text{Ed}}/\mathbb{F}_r$.

Figure 4 depicts the four types of lollipops that can arise under our construction. These types depend on the field of definition of the curve E_X and the field of definition of the curve \mathcal{E}_X . Types (a) and (b) have E_X defined over \mathbb{F}_p , while types (c) and (d) have E_X defined over \mathbb{F}_q . Types (a) and (c) have \mathcal{E}_X defined over \mathbb{F}_{p^2} as in Proposition 1, while types (b) and (d) have \mathcal{E}_X defined over \mathbb{F}_{p^4} as in Proposition 2.

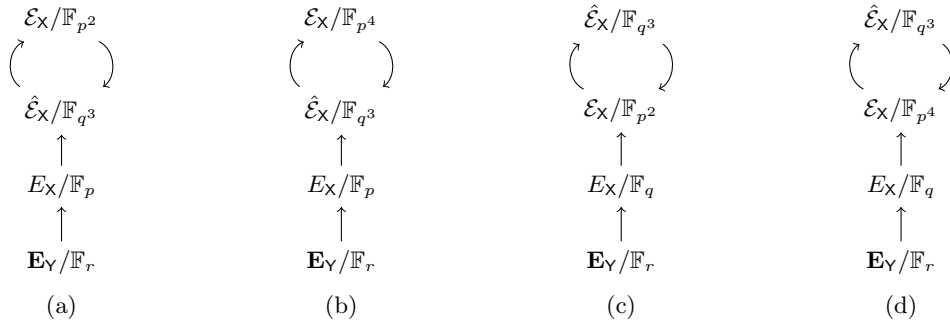


Fig. 4. The four types of lollipop-X-Y instances. Further explanation in text.

Table 1 summarises the 18 lollipops that follow it. We do not recommend that readers trudge through the details of all 18 of the example lollipops, but rather to cherry-pick one or two examples of interest after viewing Table 1. We wanted to include all of the lollipops for which there was an appreciable difference between X and Y, and for which the ECDLP and DLP securities were in the same ballpark. The only exception here is `lollipop-956-451`, whose ECDLP security is far greater than its DLP security; we included this because its the only example of practical interest that we found at the 128-bit security level.

Readers wishing to verify or experiment with any of the lollipops can refer to the `Magma` or `Pari/GP` code that is found at

<https://github.com/microsoft/lollipops>.

All of the lollipops have slightly different properties, the most practically relevant of which are given in Table 1. The second column indicates the lollipop type, in reference to the four possibilities in Figure 4. The next column gives \hat{D} , the discriminant of the non-pairing-friendly cycle $\mathbf{E}_Y/\mathbb{F}_r$ and $\hat{\mathbf{E}}_Y/\mathbb{F}_{\hat{r}}$; these parameters were computed by tweaking the routine¹⁵ used to find Masson, Sanso and Zhang’s Bandersnatch curve [30]. Observe that Examples 1 and 6 have $\hat{D} = 3$, which means that both \mathbf{E} and $\hat{\mathbf{E}}$ can exploit efficient endomorphisms of the form $\phi: (x, y) \rightarrow (\xi x, y)$, similar to the `secp/secq` [34] and `Pasta` [27] cycles. The only other examples which may have an endomorphism that is efficient enough to use in practice are Example 3, which has $\hat{D} = 67$, and Example 5, which has $\hat{D} = 43$; these correspond to endomorphisms of degree 17 and 11, respectively.¹⁶ The next three columns are associated with the pairing-friendly curve, E_Y , in the stick of the lollipop. The first gives the discriminant D of the generalised Pell equation in (4) that was solved to find the lollipop, and the second gives the embedding degree, k , of E_Y (with respect to r). If $k = 4$, then E_Y is defined over \mathbb{F}_p and the CM equation is (8); if $k = 6$, then E_Y is defined over \mathbb{F}_q and the CM equation is (9). The next column gives the bit-security of $E_Y[r]$ against Pollards rho

¹⁵ See <https://github.com/asanso/Bandersnatch/blob/main/python-ref-impl/small-disc-curves.py>.

¹⁶ If $\{1, \beta\}$ is an integral basis for the ring of integers in $\mathbb{Q}(\sqrt{\hat{D}})$, then the degree of the endomorphism ϕ is $N(\beta) - 1$; for all of the \hat{D} in Table 1, we get $\deg(\phi) = (\hat{D} + 1)/4$. Stark’s algorithm [39] can be used to derive the explicit formulas to compute ϕ .

algorithm [35], calculated as $\lceil \log_2(\sqrt{\pi r/4}) \rceil$. The remaining columns summarise the curves in the lollipop, starting with the field of definition of $\mathcal{E}_X/\mathbb{F}_{p^u}$, where $u \in \{2, 4\}$ depending on whether the lollipop corresponds to Proposition 1 or Proposition 2. We then give the j -invariants of $\mathcal{E}_X \in \mathbb{F}_p$ and $\hat{\mathcal{E}}_X \in \mathbb{F}_q$, which are computed during Bröker’s algorithm (see Section 3). Finally, in the last two columns we estimate the DLP security of the lollipop against the special tower number field sieve (S-TNFS), which were obtained using the **SageMath** [43] program for TNFS simulation by Guillemic and Singh [25]. All the simulations were run for 10^5 samples with $\deg h = 2$ because the prime p is given by a polynomial of degree 2 for our curves [17, Appendix B].

Example # (lollipop-) X-Y	Fig. 4 type	the (ordinary) stick				the (supersingular) lollipop				
		\mathbf{E}_V cycle	friendly E_V		sec.	$\mathcal{E}_X/\mathbb{F}_{p^u}$		$\hat{\mathcal{E}}_X$	sec.	
			\hat{D}	D		k	$E_V[r]$		u	$j(\mathcal{E}_X)$
1. 305-158	(a)	3	54105234	4	78	2	1728	0	77	88
2. 312-164	(b)	192547	6110889	4	81	4	-884736	0	78	89
3. 314-154	(c)	67	8162838387	6	76	2	1728	8000	79	89
4. 347-192	(b)	159307	118564569	4	95	4	-3375	0	82	95
5. 348-168	(b)	43	8310359121	4	83	4	-3375	0	83	95
6. 351-196	(a)	3	180658	4	97	2	1728	0	83	95
7. 354-182	(b)	18403	11984649	4	90	4	21 ... 33	0	83	95
8. 360-262	(a)	101971	6515276374	4	130	2	1728	8000	84	95
9. 442-201	(a)	427	1121454146	4	100	2	1728	0	94	106
10. 447-234	(c)	6339	21781087203	6	116	2	1728	8000	95	106
11. 454-179	(c)	355	7643719763	6	89	2	1728	8000	96	106
12. 470-217	(b)	2003	6965939657	4	108	4	-32768	0	97	109
13. 489-201	(b)	547	372894729	4	100	4	-3375	0	98	111
14. 493-189	(b)	57891	9926408913	4	93	4	-3375	0	99	111
15. 538-235	(a)	22339	137671666	4	117	2	1728	8000	104	115
16. 574-261	(a)	3019	4381481154	4	129	2	1728	0	108	119
17. 585-216	(d)	3315	8780293827	6	107	4	-3375	11 ... 85	109	121
18. 956-451	(a)	TBD	120605958	4	225	2	1728	0	142	153

Table 1. A summary of the 18 lollipops in this section. Further explanation in text.

Example 1. (lollipop-305-158). Solving (8) with $D = 54105234$ finds $x \equiv 10 \pmod{12}$ such that p and q are prime 305-bit primes and gives a type-(a) lollipop in Fig. 4. The curves in the cycle are $\hat{\mathcal{E}}_{305}/\mathbb{F}_q: y^2 = x^3 + 1$ and $\mathcal{E}_{305}: y^2 = x^3 + (\mu + 1)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{305}/\mathbb{F}_p has embedding degree 4 with respect to a 158-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{158}^W: y^2 = x^3 - 3x + b$ with $b = 7032$ and $\mathbf{E}_{158}^d: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 7821$. Both curves in the cycle \mathbf{E}_{158} and $\hat{\mathbf{E}}_{158}$ have $\hat{D} = 3$ and are equipped with endomorphisms of the form $\phi: (x, y) \mapsto (\xi x, y)$.

Example 2. (lollipop-312-164). Solving (8) with $D = 6110889$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 312-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker’s algorithm terminates with $D = 19$ and $H_{\mathcal{D}}(X) = X + 884736$ outputs a curve $\mathcal{E}_{312}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{312}) = -884736$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 13 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{312}/\mathbb{F}_q: y^2 = x^3 + 1$. The curve in the stick is E_{312}/\mathbb{F}_p has embedding degree 4 with respect to a 164-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{164}^W: y^2 = x^3 - 3x + b$ with $b = 6457$ and $\mathbf{E}_{164}^d: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 2709$. Both curves in the cycle \mathbf{E}_{164} and $\hat{\mathbf{E}}_{164}$ have $\hat{D} = 192547$.

Example 3. (lollipop-314-154). Solving (9) with $D = 8162838387$ finds $x \equiv 6 \pmod{12}$ such that p and q are prime 314-bit primes and gives a type-(c) lollipop in Fig. 4. Over \mathbb{F}_q , Bröker’s algorithm terminates with $\hat{D} = -8$ and $H_{\hat{D}}(X) = X - 8000$ and outputs $\hat{\mathcal{E}}_{314}$ with $j(\hat{\mathcal{E}}_{314}) = 8000$. The other curve in the cycle is $\mathcal{E}_{314}: y^2 = x^3 + (1 + \mu)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{314}/\mathbb{F}_q has embedding degree 6 with respect to a 154-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{154}^W: y^2 = x^3 - 3x + b$ with $b = 17221$ and $\mathbf{E}_{154}^{\text{Ed}}: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 4468$. Both curves in the cycle \mathbf{E}_{154} and $\hat{\mathbf{E}}_{154}$ have $\hat{D} = 67$.

Example 4. (lollipop-347-192). Solving (8) with $D = 118564569$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 347-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker’s algorithm terminates with $\mathcal{D} = 7$ and $H_{\mathcal{D}}(X) = X + 3375$ outputs a curve $\mathcal{E}_{347}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{347}) = -3375$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 2 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{347}/\mathbb{F}_q: y^2 = x^3 + 1$. The curve in the stick is E_{347}/\mathbb{F}_p has embedding degree 4 with respect to a 192-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{192}^W: y^2 = x^3 - 3x + b$ with $b = 11566$ and $\mathbf{E}_{192}^{\text{Ed}}: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 69127$. Both curves in the cycle \mathbf{E}_{192} and $\hat{\mathbf{E}}_{192}$ have $\hat{D} = 159307$.

Example 5. (lollipop-348-168). Solving (8) with $D = 8310359121$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 347-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker’s algorithm terminates with $\mathcal{D} = 7$ and $H_{\mathcal{D}}(X) = X + 3375$ outputs a curve $\mathcal{E}_{348}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{348}) = -3375$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 7 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{347}/\mathbb{F}_q: y^2 = x^3 + 1$. The curve in the stick is E_{348}/\mathbb{F}_p has embedding degree 4 with respect to a 168-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{168}^W: y^2 = x^3 - 3x + b$ with $b = 26688$ and $\mathbf{E}_{168}^{\text{Ed}}: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 78971$. Both curves in the cycle \mathbf{E}_{168} and $\hat{\mathbf{E}}_{168}$ have $\hat{D} = 43$.

Example 6. (lollipop-351-196). Solving (8) with $D = 1625922$ finds $x \equiv 10 \pmod{12}$ such that p and q are prime 351-bit primes and gives a type-(a) lollipop in Fig. 4. The curves in the cycle are $\hat{\mathcal{E}}_{351}/\mathbb{F}_q: y^2 = x^3 + 1$ and $\mathcal{E}_{351}: y^2 = x^3 + (\mu + 1)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{351}/\mathbb{F}_p has embedding degree 4 with respect to a 196-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{196}^W: y^2 = x^3 - 3x + b$ with $b = 7193$ and $\mathbf{E}_{196}^{\text{Ed}}: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 128421$.

An unlikely coincidence arose in this example. There were actually two prime order curves over \mathbb{F}_r with CM discriminant -3 . The curve $\mathbf{E}_{196}/\mathbb{F}_r: y^2 = x^3 - 5$ has prime group order \hat{r} , forming a cycle with the curve $\hat{\mathbf{E}}_{196}/\mathbb{F}_{\hat{r}}: y^2 = x^3 - 5$, while the curve $\mathbf{E}'_{196}/\mathbb{F}_r: y^2 = x^3 + 28$ has prime group order \hat{r}' , forming a cycle with the curve $\hat{\mathbf{E}}'_{196}/\mathbb{F}_{\hat{r}'}: y^2 = x^3 + 28$. We depict this in Figure 5, where we dropped the subscripts. All of these curves come equipped with endomorphisms of the form $\phi: (x, y) \mapsto (\xi x, y)$.

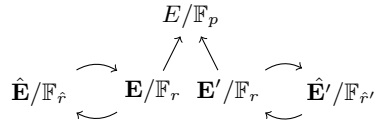


Fig. 5. A rare coincidence of two $D = -3$ non-pairing-friendly cycles possible over \mathbb{F}_r .

Example 7. (lollipop-354-182). Solving (8) with $D = 11984649$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 354-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker’s algorithm terminates with $\mathcal{D} = 31$ and $H_{\mathcal{D}}(X)$ of degree 3, and outputs a curve $\mathcal{E}_{354}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{354}) = 21 \dots 33$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 2 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{354}/\mathbb{F}_q: y^2 = x^3 + 1$. The curve in the stick is E_{354}/\mathbb{F}_p has embedding degree 4 with respect to a 182-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{182}^W: y^2 = x^3 - 3x + b$ with $b = 7906$

and $\mathbf{E}_{182}^{\text{Ed}}$: $-x^2 + y^2 = 1 + dx^2y^2$ with $d = 2758$. Both curves in the cycle \mathbf{E}_{182} and $\hat{\mathbf{E}}_{182}$ have $\hat{D} = 18403$.

Example 8. (lollipop-360-262). Solving (8) with $D = 6515276374$ finds $x \equiv 6 \pmod{12}$ such that p and q are prime 360-bit primes and gives a type-(a) lollipop in Fig. 4. Over \mathbb{F}_q , Bröker's algorithm terminates with $\hat{D} = -8$ and $H_{\hat{D}}(X) = X - 8000$ and outputs $\hat{\mathcal{E}}_{360}$ with $j(\hat{\mathcal{E}}_{360}) = 8000$. The other curve in the cycle is \mathcal{E}_{360} : $y^2 = x^3 + (2 + \mu)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{360}/\mathbb{F}_p has embedding degree 4 with respect to a 262-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{262}^{\text{W}}$: $y^2 = x^3 - 3x + b$ with $b = 43954$ and $\mathbf{E}_{262}^{\text{Ed}}$: $-x^2 + y^2 = 1 + dx^2y^2$ with $d = 119965$. The curves in the cycle \mathbf{E}_{262} and $\hat{\mathbf{E}}_{262}$ have $\hat{D} = 101971$.

Example 9. (lollipop-442-201). Solving (8) with $D = 1121454146$ finds $x \equiv 10 \pmod{12}$ such that p and q are prime 442-bit primes and gives a type-(a) lollipop in Fig. 4. The curves in the cycle are $\hat{\mathcal{E}}_{442}/\mathbb{F}_q$: $y^2 = x^3 + 1$ and \mathcal{E}_{442} : $y^2 = x^3 + (\mu + 1)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{442}/\mathbb{F}_p has embedding degree 4 with respect to a 201-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{201}^{\text{W}}$: $y^2 = x^3 - 3x + b$ with $b = 1858$ and $\mathbf{E}_{201}^{\text{Ed}}$: $-x^2 + y^2 = 1 + dx^2y^2$ with $d = 80379$. Both curves in the cycle \mathbf{E}_{201} and $\hat{\mathbf{E}}_{201}$ have $\hat{D} = 18403$.

Example 10. (lollipop-447-234). Solving (9) with $D = 8162838387$ finds $x \equiv 6 \pmod{12}$ such that p and q are prime 447-bit primes and gives a type-(c) lollipop in Fig. 4. Over \mathbb{F}_q , Bröker's algorithm terminates with $\hat{D} = -8$ and $H_{\hat{D}}(X) = X - 8000$ and outputs $\hat{\mathcal{E}}_{447}$ with $j(\hat{\mathcal{E}}_{447}) = 8000$. The other curve in the cycle is \mathcal{E}_{447} : $y^2 = x^3 + (5 + \mu)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{447}/\mathbb{F}_q has embedding degree 6 with respect to a 234-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{234}^{\text{W}}$: $y^2 = x^3 - 3x + b$ with $b = 10885$ and $\mathbf{E}_{234}^{\text{Ed}}$: $x^2 + y^2 = 1 + dx^2y^2$ with $d = -54523$. Both curves in the cycle \mathbf{E}_{234} and $\hat{\mathbf{E}}_{234}$ have $\hat{D} = 6339$.

Example 11. (lollipop-454-179). Solving (9) with $D = 7643719763$ finds $x \equiv 6 \pmod{12}$ such that p and q are prime 454-bit primes and gives a type-(c) lollipop in Fig. 4. Over \mathbb{F}_q , Bröker's algorithm terminates with $\hat{D} = -8$ and $H_{\hat{D}}(X) = X - 8000$ and outputs $\hat{\mathcal{E}}_{454}$ with $j(\hat{\mathcal{E}}_{454}) = 8000$. The other curve in the cycle is \mathcal{E}_{454} : $y^2 = x^3 + (5 + \mu)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{454}/\mathbb{F}_q has embedding degree 6 with respect to a 179-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{179}^{\text{W}}$: $y^2 = x^3 - 3x + b$ with $b = 10827$ and $\mathbf{E}_{179}^{\text{Ed}}$: $x^2 + y^2 = 1 + dx^2y^2$ with $d = -68661$. Both curves in the cycle \mathbf{E}_{179} and $\hat{\mathbf{E}}_{179}$ have $\hat{D} = 355$.

Example 12. (lollipop-470-217). Solving (8) with $D = 6965939657$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 470-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker's algorithm terminates with $D = 11$ and $H_D(X) = X + 32768$ and outputs a curve $\mathcal{E}_{470}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{470}) = -32768$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 11 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{470}/\mathbb{F}_q$: $y^2 = x^3 + 1$. The curve in the stick is E_{470}/\mathbb{F}_p has embedding degree 4 with respect to a 217-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{217}^{\text{W}}$: $y^2 = x^3 - 3x + b$ with $b = 72802$ and $\mathbf{E}_{217}^{\text{Ed}}$: $-x^2 + y^2 = 1 + dx^2y^2$ with $d = 70192$. Both curves in the cycle \mathbf{E}_{217} and $\hat{\mathbf{E}}_{217}$ have $\hat{D} = 2003$.

Example 13. (lollipop-489-201). Solving (8) with $D = 372894729$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 489-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker's algorithm terminates with $D = 7$ and $H_D(X) = X + 3375$ and outputs a curve $\mathcal{E}_{489}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{489}) = -3375$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 2 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{489}/\mathbb{F}_q$: $y^2 = x^3 + 1$. The curve in the stick is E_{489}/\mathbb{F}_p has embedding degree 4 with respect to a 201-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{201}^{\text{W}}$: $y^2 = x^3 - 3x + b$ with $b = 4438$ and $\mathbf{E}_{201}^{\text{Ed}}$: $-x^2 + y^2 = 1 + dx^2y^2$ with $d = 96027$. Both curves in the cycle \mathbf{E}_{201} and $\hat{\mathbf{E}}_{201}$ have $\hat{D} = 547$.

Example 14. (lollipop-493-189). Solving (8) with $D = 9926408913$ finds $x \equiv 4 \pmod{12}$ such that p and q are prime 493-bit primes and gives a type-(b) lollipop in Fig. 4. Bröker's algorithm terminates with $D = 7$ and $H_D(X) = X + 3375$ and outputs a curve $\mathcal{E}_{493}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{493}) = -3375$, where

$\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 2 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. The other curve in the cycle is $\hat{\mathcal{E}}_{493}/\mathbb{F}_q: y^2 = x^3 + 1$. The curve in the stick is E_{493}/\mathbb{F}_p has embedding degree 4 with respect to a 189-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{189}^W: y^2 = x^3 - 3x + b$ with $b = 40288$ and $\mathbf{E}_{189}^{Ed}: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 54091$. Both curves in the cycle \mathbf{E}_{189} and $\hat{\mathbf{E}}_{189}$ have $\hat{D} = 57891$.

Example 15. (lollipop-538-235). Solving (8) with $D = 137671666$ finds $x \equiv 6 \pmod{12}$ such that p and q are prime 538-bit primes and gives a type-(a) lollipop in Fig. 4. Over \mathbb{F}_q , Bröker’s algorithm terminates with $\hat{D} = -8$ and $H_{\hat{D}}(X) = X - 8000$ and outputs $\hat{\mathcal{E}}_{538}$ with $j(\hat{\mathcal{E}}_{538}) = 8000$. The other curve in the cycle is $\mathcal{E}_{538}: y^2 = x^3 + (1 + \mu)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{538}/\mathbb{F}_p has embedding degree 4 with respect to a 235-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{235}^W: y^2 = x^3 - 3x + b$ with $b = 11095$ and $\mathbf{E}_{235}^{Ed}: x^2 + y^2 = 1 + dx^2y^2$ with $d = 101828$. The curves in the cycle \mathbf{E}_{235} and $\hat{\mathbf{E}}_{235}$ have $\hat{D} = 22339$.

Example 16. (lollipop-574-261). Solving (8) with $D = 4381481154$ finds $x \equiv 10 \pmod{12}$ such that p and q are prime 574-bit primes and gives a type-(a) lollipop in Fig. 4. The curves in the cycle are $\hat{\mathcal{E}}_{574}/\mathbb{F}_q: y^2 = x^3 + 1$ and $\mathcal{E}_{574}: y^2 = x^3 + (\mu + 1)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{574}/\mathbb{F}_p has embedding degree 4 with respect to a 261-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{261}^W: y^2 = x^3 - 3x + b$ with $b = 7182$ and $\mathbf{E}_{261}^{Ed}: -x^2 + y^2 = 1 + dx^2y^2$ with $d = 75745$. Both curves in the cycle \mathbf{E}_{261} and $\hat{\mathbf{E}}_{261}$ have $\hat{D} = 3019$.

Example 17. (lollipop-585-216). Solving (9) with $D = 8780293827$ finds $x \equiv 0 \pmod{12}$ such that p and q are prime 585-bit primes and gives a type-(d) lollipop in Fig. 4. Bröker’s algorithm terminates with $\mathcal{D} = 7$ and $H_{\mathcal{D}}(X) = X + 3375$ and outputs a curve $\mathcal{E}_{585}/\mathbb{F}_{p^4}$ with $j(\mathcal{E}_{585}) = -3375$, where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 2 = 0$ and $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}(\nu)$ with $\nu^2 = \mu$. Over \mathbb{F}_q , Bröker’s algorithm terminates with $\hat{D} = 47$ and $H_{\hat{D}}(X)$ of degree 5, and outputs $\hat{\mathcal{E}}_{585}$ with $j(\hat{\mathcal{E}}_{585}) = 11 \dots 85$. The curve in the stick is E_{585}/\mathbb{F}_q has embedding degree 6 with respect to a 216-bit prime r . The non-pairing-friendly curves over \mathbb{F}_r are $\mathbf{E}_{216}^W: y^2 = x^3 - 3x + b$ with $b = 8146$ and $\mathbf{E}_{216}^{Ed}: x^2 + y^2 = 1 + dx^2y^2$ with $d = -36607$. Both curves in the cycle \mathbf{E}_{216} and $\hat{\mathbf{E}}_{216}$ have $\hat{D} = 3315$.

Example 18. (lollipop-956-451). Solving (8) with $D = 120605958$ finds $x \equiv 10 \pmod{12}$ such that p and q are prime 956-bit primes and gives a type-(a) lollipop in Fig. 4. The curves in the cycle are $\hat{\mathcal{E}}_{956}/\mathbb{F}_q: y^2 = x^3 + 1$ and $\mathcal{E}_{956}: y^2 = x^3 + (\mu + 1)x$ where $\mathbb{F}_{p^2} = \mathbb{F}_p(\mu)$ with $\mu^2 + 1 = 0$. The curve in the stick is E_{956}/\mathbb{F}_p has embedding degree 4 with respect to a 451-bit prime r . The searches for the non-pairing-friendly curves are still running. Given the discrepancy in the DLP and ECDLP security for this lollipop (see Table 1), it is possible that a composite order curve \mathbf{E}/\mathbb{F}_r with a prime order subgroup of closer to 256-bits is preferable. In this case it should be possible to find such a curve that has a discriminant low enough to exploit endomorphisms, analogous to the Bandersnatch curve [30] (but with a larger cofactor).

6 Conclusion

We gave the first construction of lollipops of pairing-friendly curves with a view towards efficient recursive proof system composition. Along the way, we gave a new way of instantiating pairing-friendly cycles using supersingular curves, which avoids some restrictions imposed by the MNT construction and ultimately paved the way for the lollipop construction in Section 4.

We applied a moderate amount of computation to search for lollipops and found 18 examples of practical interest between the 80- and 128-bit security levels. Our search did not venture higher than $D = 10^{12}$, while Sutherland’s CM records are for discriminants that are several orders of magnitude larger than this. This means we only searched a fraction of the discriminants for which the CM method is feasible. In reference to the practical adjustments that can be made to the main search algorithm in Section 4, we therefore believe that better lollipop instances are merely a matter of more computational investment.

Acknowledgments. Part of this work was done while Gaurish was an intern at Microsoft Research. Thanks to Patrick Longa, Andrew Sutherland, and Allan Steel for their technical assistance. Thanks to Michael Naehrig and Greg Zaverucha for several discussions during the preparation of this work.

References

1. Aranha, D.F., El Housni, Y., Guillevic, A.: A survey of elliptic curves for proof systems. *Designs, Codes and Cryptography* pp. 1–46 (2022)
2. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer (2005)
3. Bellés-Muñoz, M., Urroz, J.J., Silva, J.: Revisiting cycles of pairing-friendly elliptic curves. In: CRYPTO. LNCS, vol. 14082, pp. 3–37. Springer (2023)
4. Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E., Virza, M.: Snarks for C: verifying program executions succinctly and in zero knowledge. In: CRYPTO. *Lecture Notes in Computer Science*, vol. 8043, pp. 90–108. Springer (2013)
5. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Scalable zero knowledge via cycles of elliptic curves. In: CRYPTO. LNCS, vol. 8617, pp. 276–294. Springer (2014)
6. Bonneau, J., Meckler, I., Rao, V., E, S.: Coda: Decentralized cryptocurrency at scale. *IACR Cryptol. ePrint Arch.* p. 352 (2020), <https://eprint.iacr.org/2020/352>
7. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. *J. Symbolic Comput.* **24**(3-4), 235–265 (1997), computational algebra and number theory
8. Bowe, S.: BLS12-381: New zk-SNARK Elliptic Curve Construction. <https://electriccoin.co/blog/new-snark-curve/> (2017)
9. Bowe, S., Grigg, J., Hopwood, D.: Halo: Recursive proof composition without a trusted setup. *IACR Cryptol. ePrint Arch.* p. 1021 (2019), <https://eprint.iacr.org/2019/1021>
10. Bröker, R.: Constructing supersingular elliptic curves. *J. Comb. Number Theory* **1**(3), 269–273 (2009)
11. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: *IEEE S&P*. pp. 315–334. IEEE Computer Society (2018)
12. Chiesa, A., Chua, L., Weidner, M.: On cycles of pairing-friendly elliptic curves. *SIAM Journal on Applied Algebra and Geometry* **3**(2), 175–192 (2019)
13. Coda + Dekrypt SNARK Challenge, CoinList: Constructing optimal pairing-friendly curves (2019), available from <https://web.archive.org/web/20240620204445/https://coinlist.co/build/coda/pages/theory>
14. Corte-Real Santos, M., Costello, C., Naehrig, M.: On cycles of pairing-friendly abelian varieties. In: CRYPTO. *Lecture Notes in Computer Science*, vol. 14928, pp. 221–253. Springer (2024)
15. Costello, C., Fournet, C., Howell, J., Kohlweiss, M., Kreuter, B., Naehrig, M., Parno, B., Zahur, S.: Geppetto: Versatile verifiable computation. In: *IEEE Symposium on Security and Privacy*. pp. 253–270. IEEE Computer Society (2015)
16. El Housni, Y., Guillevic, A.: Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition. In: CANS, LNCS, vol. 12579, pp. 259–279. Springer (2020)
17. El Housni, Y., Guillevic, A.: Families of SNARK-friendly 2-chains of elliptic curves. In: EUROCRYPT. LNCS, vol. 13276, pp. 367–396. Springer (2022)
18. Enge, A., Sutherland, A.V.: Class invariants by the CRT method. In: *Algorithmic number theory, Lecture Notes in Comput. Sci.*, vol. 6197, pp. 142–156. Springer, Berlin (2010)
19. F. Morain, F.: Building cyclic elliptic curves modulo large primes. In: EUROCRYPT, *Lecture Notes in Comput. Sci.*, vol. 547, pp. 328–336. Springer, Berlin (1991)
20. Foundation, M.: The Mina Protocol. <https://minaprotocol.com/>
21. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. *Journal of cryptology* **23**, 224–280 (2010)
22. Galbraith, S.: Pairings. In: *Advances in elliptic curve cryptography, London Math. Soc. Lecture Note Ser.*, vol. 317, pp. 183–213. Cambridge Univ. Press, Cambridge (2005)
23. Groth, J.: On the size of pairing-based non-interactive arguments. In: EUROCRYPT. LNCS, vol. 9666, pp. 305–326. Springer (2016)
24. Guillevic, A.: Pairing-friendly curves. <https://members.loria.fr/AGuillevic/pairing-friendly-curves/> (2021)
25. Guillevic, A., Singh, S.: On the alpha value of polynomials in the tower number field sieve algorithm. *Mathematical Cryptology* **1**(1), 1–39 (2021)

26. Hasse, H.: Zur theorie der abstrakten elliptischen funktionenkörper iii. die struktur des meromorphismenrings. die riemannsche vermutung. *Journal für die reine und angewandte Mathematik* **175**, 193–208 (1936)
27. Hopwood, D.: The Pasta Curves for Halo 2 and Beyond. <https://electriccoin.co/blog/the-pasta-curves-for-halo-2-and-beyond/> and <https://github.com/zcash/pasta> (2020)
28. Hopwood, D.: Pluto/Eris supporting evidence. <https://github.com/daira/pluto-eris> (2021)
29. Karabina, K., Teske, E.: On prime-order elliptic curves with embedding degrees $k = 3, 4$, and 6 . In: ANTS-VIII. LNCS, vol. 5011, pp. 102–117. Springer (2008)
30. Masson, S., Sanso, A., Zhang, Z.: Bandersnatch: a fast elliptic curve built over the BLS12-381 scalar field. *IACR Cryptol. ePrint Arch.* p. 1152 (2021), <https://eprint.iacr.org/2021/1152>
31. Meckler, I.: Constructing optimal pairing-friendly curves. <https://github.com/MinaProtocol/snark-challenge/blob/master/docs/theory.markdown> (2019)
32. Meckler, I.: *o(1) labs fork of zexe: implementation of bn382-plain*. https://github.com/o1-labs/zexe/tree/master/algebra/src/bn_382 (2020)
33. Miyaji, A., Nakabayashi, M., Takano, S.: Characterization of elliptic curve traces under FR-reduction. In: ICISC 2000. LNCS, vol. 2015, pp. 90–108. Springer (2000)
34. Poelstra, A.: Curve with group order $2^{255} - 19$. <https://moderncrypto.org/mail-archive/curves/2018/000992.html> (2018)
35. Pollard, J.M.: Monte Carlo methods for index computation (mod p). *Mathematics of computation* **32**(143), 918–924 (1978)
36. Schoof, R.: Nonsingular plane cubic curves over finite fields. *J. Combin. Theory Ser. A* **46**(2), 183–211 (1987). [https://doi.org/10.1016/0097-3165\(87\)90003-3](https://doi.org/10.1016/0097-3165(87)90003-3)
37. Silverman, J.H.: *The arithmetic of elliptic curves*, vol. 106. Springer (2009)
38. Silverman, J.H., Stange, K.E.: Amicable pairs and aliquot cycles for elliptic curves. *Exp. Math.* **20**(3), 329–357 (2011)
39. Stark, H.M.: Class-numbers of complex quadratic fields. In: *Modular Functions of One Variable I*, pp. 153–174. Springer (1973)
40. Sutherland, A.V.: Computing Hilbert class polynomials with the Chinese remainder theorem. *Math. Comp.* **80**(273), 501–538 (2011)
41. Sutherland, A.V.: Accelerating the CM method. *LMS J. Comput. Math.* **15**, 172–204 (2012)
42. The PARI Group, Univ. Bordeaux: PARI/GP version 2.15.5 (2024), available from <http://pari.math.u-bordeaux.fr/>
43. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.3) (2024), <https://www.sagemath.org>
44. Washington, L.C.: *Introduction to cyclotomic fields*, Graduate Texts in Mathematics, vol. 83. Springer-Verlag, New York, second edn. (1997)
45. Waterhouse, W.C.: Abelian varieties over finite fields. *Annales scientifiques de l'École normale supérieure* **2**(4), 521–560 (1969)
46. Williamson, Z.J.: Goblin Plonk: lazy recursive proof composition. <https://hackmd.io/@aztec-network/B19AA8812> (2023)