More Embedded Curves for SNARK-Pairing-Friendly Curves

Aurore Guillevic

Univ Rennes, Inria, CNRS, IRISA, Rennes, France

Abstract. Embedded curves are elliptic curves defined over a prime field whose order (characteristic) is the prime subgroup order (the scalar field) of a pairing-friendly curve. Embedded curves have a large prime-order subgroup of cryptographic size but are not pairing-friendly themselves. Sanso and El Housni published families of embedded curves for BLS pairing-friendly curves. Their families are parameterized by polynomials, like families of pairing-friendly curves are. However their work did not found embedded families for KSS pairing-friendly curves. In this note we show how the problem of finding families of embedded curves is related to the problem of finding optimal formulas for $\mathbb{G}_1$ subgroup membership testing on the pairing-friendly curve side. Then we apply Smith's technique and Dai, Lin, Zhao, and Zhou (DLZZ) criteria to obtain the formulas of embedded curves with KSS, and outline a generic algorithm for solving this problem in all cases. We provide two families of embedded curves for KSS18 and give examples of cryptographic size. We also suggest alternative embedded curves for BLS that have a seed of much lower Hamming weight than Sanso et al. and much higher 2-valuation for fast FFT. In particular we highlight BLS12 curves which have a prime-order embedded curve that form a plain cycle (no pairing), and a second (plain) embedded curve in Montgomery form. A Brezing-Weng outer curve to have a pairing-friendly 2-chain is also possible like in the BLS12-377-BW6-761 construction. All curves have j-invariant 0 and an endomorphism for a faster arithmetic on the curve side.

Keywords: pairing-friendly curves · SNARK

1 Introduction

Elliptic curves for proof systems. With the development of proof-of-knowledge systems, in particular SNARK (Succinct Non-interactive ARgument of Knowledge), Pairing-friendly curves know a recent regain of interest. These curves are elliptic curves usually defined over a prime field $\mathbb{F}_p$ and equipped with an efficient bilinear map $e(\cdot, \cdot)$ that pairs points on the curve and outputs a value in a finite field. To instantiate the proof systems, a set of elliptic curves is required, and how they are related to each other varies. One case study can be first a zero-knowledge proof using a group of an elliptic curve, then a SNARK to prove the verification step of the previous proof, then a second SNARK to prove the verification circuit of the former. However the elliptic curves involved are not designed in the order they are used. The starting point is usually a pairing-friendly curve that is used for a SNARK, Groth [Gro16] was the first to achieve a cost as small as three pairings and additional multiplications/exponentiations. One starts by choosing that curve because a pairing-friendly curve should be designed on purpose. Usually, elliptic curves are not pairing-friendly. For the initial step (a first proof), Kosba et al. [KZM+15] were the first to introduce what is called now an *embedded curve*, that is a plain elliptic curve

E-mail: aurore.guillevic@inria.fr (Aurore Guillevic)

(non-pairing-friendly) whose field of definition has order given by the prime-order subgroup of the pairing-friendly curve. For the second SNARK, in the Geppetto work [CFH+15], Costello et al. constructed a 2-chain of pairing-friendly curves where a prime-order BN curve is the base curve. There are also *cycle* variants. One can mention ZEXE's cycle of pairing-friendly MNT curves [BCG+20], hybrid cycles (half-pairing cycles) made of a pairing-friendly BN curve and a plain curve, such as the Aztec Protocol half-pairing cycle made of the Ethereum BN-254 curve whith the plain curve Grumpkin [Azt] of 254 bits, Mina testnet half-cycle of 382 bits [Mec20], Pluto-Eris half-cycle [Hop21] of 446 bits; plain cycles (`secp256k1` and `secq256k1` [Poe18], Tweedle [Hop17b, BGH19], Pasta [Hop20b]). A survey paper can be found at [AEHG22].

Embedded curves. The initial proof statement is better formulated in a field that avoids *arithmetic mismatch*. For that, *embedded curves* are designed so that their field of definition is the scalar field of the pairing-friendly curve (the SNARK curve). Figure 1 from [AEHG22] illustrates the CØCØ embedded curve construction. The embedded curve does not form a cycle. For CØCØ, the embedded curve has order a small factor times a prime, hence cannot form a cycle (for that a prime-order curve would be required). Jubjub [Hop17a] and Bandersnatch [MSZ21] are embedded curves in twisted Edwards form for the BLS12-381 curve.

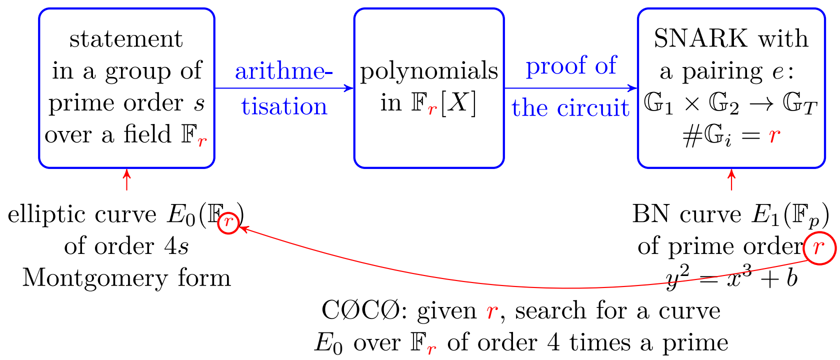

Figure 1: Kosba et al. construction [KZM+15], figure from [AEHG22]

Our contributions. We extend the construction of Sanso and El Housni [SEH24] to KSS curves and give, based on Dai, Lin, Zhao, and Zhou theorem [DLZZ23] and Smith technique [Smi15], an algorithm to derive the parameterized families of embedded curves which have the same discriminant as the pairing-friendly curve. To obtain prime-order embedded curves, the polynomial parameterizing the curve order should *generate primes*, a problem well-known in pairing-friendly constructions (see the Taxonomy paper [FST10]). For KSS18 curves of discriminant $-D = -3$, we obtain two embedded curve families that can generate curves of prime order. For KSS16 curves of discriminant $-D = -4$, the embedded curve with $-D = -4$ cannot be of prime order however its order can be four times a prime. We wrote a SageMath/Python script based on the `tnfs-alpha` code to generate seeds of BLS and KSS pairing-friendly curves that have a suitable embedded curve. Ou technique can be extended to Scott–Guillevic (Aurifeuillean) and Gasnier–Guillevic curves [SG18, GG23].

Organization of the paper. In Section 2 we propose better seeds (with much lower Hamming weight) for endomorphism-equipped embedded curves with BLS12. We target a 2-valuation of $2^{32} \mid p-1,\ r-1$. In Section 3 we solve the problem highlighted by Sanso and El Housni for KSS and provide *endomorphism-equipped prime-order embedded curve families* for KSS18. We also provide such families of order four times a prime for KSS16 curves. We conclude in Section 5.

2 Endomorphism-equipped embedded curves with BLS12 and BLS24

2.1 Sanso and El Housni construction of embedded curve families with BLS12 and BLS24

In [SEH24] Sanso and El Housni introduce a technique to obtain families of endomorphism-equipped embedded curves with BLS. They observe that the scalar field of BLS12 curves $r(u) = u^4 - u^2 + 1$ can be written in the form $r(u) = (t_e^2 + 3y_e^2)/4 = ((2u^2-1)^2 + 3(1)^2)/4$ to generate an embedded curve family with $-D = -3$, and $r(u) = (t_e^2 + 4y_e^2)/4 = ((2u)^2 + 4(1-u^2)^2)/4$ to generate an embedded curve family with $-D = -4$.

We rephrase Sanso and El Housni procedure as Algorithm 2.1. The output for BLS12 is Table 1 for embedded curves with j-invariant 0 ($-D = -3$). Two families can produce prime-order embedded curves. For $-D = -4$, Sanso and El Housni procedure will output Table 2. curves with j-invariant 1728 cannot have prime order as they always have at least one point of order two and an even order. One can note that the order is $u^4 - 3u^2 + 4 = u^2(u^2-1) - 2u^2 + 4$ which is always even whenever the parity of u.

Table 1: Parameters (t_e, y_e) such that $r = (t_e^2 + 3y_e^2)/4$ with $-D = -3$. A first pair is $(t_1, y_1) = (2u^2 - 1, 1)$ and the other pairs are for the quadratic, cubic and sextic twists. The fourth one's order $r + 1 - t_e = u^4 - 2u^2 + 4$ is not prime but can give three times a prime ($u = 1 \bmod 3$).

(t_e, y_e) s.t. $r = (t_e^2 + 3y_e^2)/4$		$r + 1 - t_e$	family
t_1, y_1	$2u^2 - 1, 1$	$u^4 - 3u^2 + 3$	yes
$-t_1, y_1$	$-2u^2 + 1, 1$	$(u^2 - u + 1)(u^2 + u + 1)$	no
$(t_1 + 3y_1)/2, (t_1 - y_1)/2$	$u^2 + 1, u^2 - 1$	$(u-1)^2(u+1)^2$	no
$(t_1 - 3y_1)/2, (t_1 + y_1)/2$	$u^2 - 2, u^2$	$u^4 - 2u^2 + 4$	(yes)
$-(t_1 - 3y_1)/2, (t_1 + y_1)/2$	$-u^2 + 2, u^2$	u^4	no
$-(t_1 + 3y_1)/2, (t_1 - y_1)/2$	$-u^2 - 1, u^2 - 1$	$u^4 + 3$	yes

Table 2: Parameters (t_e, y_e) such that $r = (t_e^2 + 4y_e^2)/4$ with $-D = -4$. A first pair is $(t_1, y_1) = (2u^2 - 2, u)$ and the other pairs are for the quadratic and quartic twists. The first one's order $r + 1 - t_e$ is not prime but can give two times a prime.

(t_e, y_e) s.t. $r = (t_e^2 + 4y_e^2)/4$		$r + 1 - t_e$	family
t_1, y_1	$2u^2 - 2, u$	$u^4 - 3u^2 + 4$	(yes)
$-t_1, y_1$	$-2u^2 + 2, u$	$u^4 + u^2 = u^2(u^2 + 1)$	no
$2y_1, t_1/2$	$2u, u^2 - 1$	$u^4 - u^2 - 2u + 2 = (u-1)^2(u^2 + 2u + 2)$	no
$-2y_1, t_1/2$	$-2u, u^2 - 1$	$u^4 - u^2 + 2u + 2 = (u+1)^2(u^2 - 2u + 2)$	no

2.2 Better seeds of embedded curves with BLS12

In [SEH24], Sanso and El Housni propose the seed `0xb504f33499580000` that generates a BLS12-380 curve and a prime-order embedded curve. Alternatively we generated the

Algorithm 2.1: Generating prime-order endormorphism-equipped embbeded curves with BLS or KSS [SEH24]

Input: parameterized pairing-friendly curve order $r(u)$ that generates primes, discriminant $-D$ for the embedded curve

Output: Embedded curve families of discriminant $-D$ or $\perp$

```
if −D is a square in Q[x]/(r(x)) then
    W_e(u) ← √(−D) mod r(u);
    (t_1(u), y_1(u)) ← half-gcd(W_e(u), r(u));
    if t_1(u)^2 + D y_1(u)^2 = 4r(u) then
        for (t_e(u), y_e(u)) in the set of twist parameters of (t_1(u), y_1(u)) do
            q_e(u) ← r + 1 − t_e;
            if q_e(u) is irreducible then
                Append (t_e, y_e, q_e) to the list of families;
        return the list of families
return ⊥
```

seeds in Table 3 of Hamming weight up to 6 in signed binary representation.

Moreover with a larger search space (Hamming weight 7), we were able to obtain seeds in Table 4 such that the BLS12 curve E admits at the same time a prime-order embedded curve E_1 (with its cycle plain curve E_0) and a second embedded curve E_2 of order 4 times a prime (like in the CØCØ construction). We think it can be of interest for interoperability purposes.

Table 3: Seeds u of Hamming weight ≤ 6 such that the BLS12 curve $E/\mathbb{F}_p$ has a high 2-valuation $2^L \mid p-1$, $2^L \mid r-1$ and admits a prime-order embedded curve $E_1/\mathbb{F}_r$ of j-invariant 0 that has a plain cycle curve $E_0/\mathbb{F}_q$, and $2^L \mid q-1$. For $2^L \mid u-1$, u is odd and the order is necessarily $q = u^4 - 3u^2 + 3$ because $q' = u^4 + 3$ is even for odd seeds u. All curves have $-D = -3$.

seed	L	equation $E_{\mathrm{BLS}}/\mathbb{F}_p$	p (bits)	r (bits)	embedded curve equation $E_1/\mathbb{F}_r$	plain cycle curve equation $E_0/\mathbb{F}_q$
`0x9ffc012000000001` $2^{63}+2^{61}-2^{50}+2^{40}+2^{37}+1$	37	$y^2 = x^3+1$	379	254	$y^2 = x^3+7$	$y^2 = x^3+15$
`-0xff97ffdffffffffff` $-2^{64}+2^{55}-2^{53}+2^{51}+2^{37}+1$	37	$y^2 = x^3+1$	383	256	$y^2 = x^3+11$	$y^2 = x^3+7$
`0x87fbc01000000001` $2^{63}+2^{59}-2^{50}-2^{46}+2^{36}+1$	36	$y^2 = x^3+1$	377	253	$y^2 = x^3+13$	$y^2 = x^3+11$
`0x80067fff00000001` $2^{63}+2^{51}-2^{49}+2^{47}-2^{32}+1$	32	$y^2 = x^3+1$	377	253	$y^2 = x^3+15$	$y^2 = x^3+5$

3 Embedded curves with KSS18

Building on Algorithm 2.1, Sanso and El Housni looked at KSS18 curves. The difficulty comes from finding a generic formula to express the parameterized KSS18 order $r = (u^6 + 37u^3 + 343)/343$ as a sum of two squares $r(u) = (t_e^2(u) + Dy_e^2(u))/4$. First note the identity $a_0^2 - a_0a_1 + a_1^2 = ((2a_0 - a_1)^2 + 3a_1^2)/4$. We rewrite r as

$$r(u) = (t^2 + 3y^2)/4 = ((t+y)/2)^2 - y(t+y)/2 + y^2 = a_0^2 - a_0a_1 + a_1^2 \tag{1}$$

and deduce that $(a_0, a_1) = ((t+y)/2, y)$ in other words, $(t, y) = (2a_0 - a_1, a_1)$. Then we recognize that (1) is exactly the formula of Dai, Lin, Zhao, and Zhou [DLZZ23, Remark 4] for $\mathbb{G}_1$ subgroup membership testing:

Table 4: Seeds u of Hamming weight 7 such that the BLS12 curve $E/\mathbb{F}_p$ has a high 2-valuation, a prime-order embedded curve $E_1/\mathbb{F}_r$ with a plain cycle curve $E_0/\mathbb{F}_q$ and a second embedded curve $E_2/\mathbb{F}_r$ of order $u^4+3=4s$ where s is prime. All curves have $-D=-3$.

seed	L	equation $E_{\mathrm{BLS}}/\mathbb{F}_p$	p (bits)	r (bits)	embedded curve equation $E_{1,2}/\mathbb{F}_r$	plain cycle curve equation $E_0/\mathbb{F}_q$
`0xffff007fda000001` $2^{64}-2^{48}+2^{39}-2^{29}-2^{27}+2^{25}+1$	25	$y^2=x^3+1$	383	256	$E_1: y^2=x^3+19$ $E_2: y^2=x^3+17$	$y^2=x^3+7$
`0xfc3ec00400000001` $2^{64}-2^{58}+2^{54}-2^{48}-2^{46}+2^{34}+1$	34	$y^2=x^3+1$	383	256	$E_1: y^2=x^3+23$ $E_2: y^2=x^3+29$	$y^2=x^3+29$
`-0xef000ffefdffffff` $-2^{64}+2^{60}+2^{56}-2^{44}+2^{32}+2^{25}+1$	25	$y^2=x^3+1$	382	256	$E_1: y^2=x^3+11$ $E_2: y^2=x^3+17$	$y^2=x^3+17$
`0xdf07fffdfc000001` $2^{64}-2^{61}-2^{56}+2^{51}-2^{33}-2^{26}+1$	26	$y^2=x^3+1$	382	256	$E_1: y^2=x^3+11$ $E_2: y^2=x^3+23$	$y^2=x^3+7$

Remark 1 ([DLZZ23, Remark 4]). The selected short vectors (a_0,a_1) listed in [DLZZ23, Table 4] satisfy that

$$\begin{cases} a_0^2-a_0a_1+a_1^2=r \text{ if } j(E)=0; \\ a_0^2+a_1^2=r \text{ if } j(E)=1728. \end{cases}$$

By [DLZZ23, Theorem 3], the recommended short vectors are actually independent with the selection of seeds.

Then we deduce that the formula Sanso and El Housni were looking for is

$$(a_0,a_1)=((u/7)^3,-18(u/7)^3-1) \iff (t,y)=(20(u/7)^3+1,-18(u/7)^3-1)\ . \quad (2)$$

We deduce Algorithm 3.1 and run it to obtain the prime-order endomorphism-equipped embedded curves with KSS18.

Algorithm 3.1: Generating prime-order endormorphism-equipped embbeded curve families with KSS18 and $-D=-3$

$r(u) \leftarrow (u^6-37u^3+343)/343$, a KSS18 curve order;
$(t_1(u),y_1(u)) \leftarrow (20(u/7)^3+1,-18(u/7)^3-1)$;
for $(t_e(u),y_e(u))$ *in the set of 6 twist parameters of* $(t_1(u),y_1(u))$ **do**
  $q_e(u) \leftarrow r+1-t_e$;
  if $q_e(u)$ *is irreducible* **then**
    Append (t_e,y_e,q_e) to the list of families;
return the list of families

Table 5: Embedded curves for KSS18, parameters (t_e,y_e) such that $r=(t_e^2+3y_e^2)/4$ with $-D=-3$. A first pair is $(t_1,y_1)=(20(u/7)^3+1,-18(u/7)^3-1)$ and the other pairs are for the quadratic, cubic and sextic twists. The first and fifth one's order $q=r+1-t_e$ are irreducible but multiple of 3.

(t_e,y_e) s.t. $r=(t_e^2+3y_e^2)/4$		$q=r+1-t_e$	family
t_1,y_1	$20(u/7)^3+1,-18(u/7)^3-1$	$(u^6+17u^3+343)/343$	(yes, 3)
$-t_1,y_1$	$-20(u/7)^3-1,-18(u/7)^3-1$	$(u^6+57u^3+1029)/343$	yes
$(t_1+3y_1)/2,(t_1-y_1)/2$	$-17(u/7)^3-1,19(u/7)^3+1$	$(u^6+54u^3+1029)/343$	yes
$(t_1-3y_1)/2,(t_1+y_1)/2$	$37(u/7)^3+2,(u/7)^3$	$u^6/7^3$	no
$-(t_1-3y_1)/2,(t_1+y_1)/2$	$-37(u/7)^3-2,(u/7)^3$	$(u^6+74u^3+1372)/343$	(yes, 3)
$-(t_1+3y_1)/2,(t_1-y_1)/2$	$17(u/7)^3+1,19(u/7)^3+1$	$(u^2-4u+7)(u^2-u+7)(u^2+5u+7)/343$	no

To conclude we mention the *halographs* project of Daira Hopwood at [Hop20a], who already in 2020 obtained the formulas of prime-order j-invariant 0 embedded curves forming

Table 6: Seeds u of Hamming weight ≤ 6 such that the KSS18 curve $E/\mathbb{F}_p$ has a high 2-valuation $2^L \mid r-1$ and admits a prime-order embedded curve $E_1/\mathbb{F}_r$ of j-invariant 0 that has a plain cycle curve $E_0/\mathbb{F}_q$. All curves have $-D=-3$.

seed	equation $E_{\mathrm{KSS}}/\mathbb{F}_p$	p (bits)	r (bits)	embedded curve equation $E_1/\mathbb{F}_r$	plain cycle curve equation $E_0/\mathbb{F}_q$
$q=(u^6+57u^3+1029)/343$					
`-0x10001efe7f00` $-2^{44}-2^{29}+2^{24}+2^{17}-2^{15}+2^{8}$	$y^2=x^3+2$	348	256	$y^2=x^3+5$	$y^2=x^3-4$
`-0xfdde07f8000` $-2^{44}+2^{37}+2^{33}+2^{29}-2^{23}+2^{15}$	$y^2=x^3+13$	348	256	$y^2=x^3+13$	$y^2=x^3-4$
`-0x1087ff6ff000` $-2^{44}-2^{39}-2^{35}+2^{23}+2^{20}+2^{12}$	$y^2=x^3+2$	348	256	$y^2=x^3+5$	$y^2=x^3-4$
$q=(u^6+54u^3+1029)/343$					
`0xfffe7f11000` $+2^{44}-2^{29}+2^{27}-2^{20}+2^{16}+2^{12}$	$y^2=x^3+2$	348	256	$y^2=x^3+7$	$y^2=x^3+2$
`-0xfdffe110200` $-2^{44}+2^{37}+2^{25}-2^{20}-2^{16}-2^{9}$	$y^2=x^3+13$	348	256	$y^2=x^3+11$	$y^2=x^3+2$
`-0xfd7ffdee000` $-2^{44}+2^{37}+2^{35}+2^{21}+2^{16}+2^{13}$	$y^2=x^3+2$	348	256	$y^2=x^3+7$	$y^2=x^3+2$

a plain cycle for BLS12 and KSS18. A careful look at the SageMath source code shows that it uses the same formulas as [SEH24] for BLS12. For KSS18, the change of variables $u \mapsto 7u$ allowed to obtain the formulas, avoiding the denominator issue that Sanso and El Housni faced.

4 A generic method

4.1 Two blocking conditions in Algorithm 2.1

4.1.1 Finding a square root of $-D$ modulo r

Looking at Algorithm 2.1, there are two steps that can fail. The first is testing if $-D$ is a square in $\mathbb{Q}(x)/(r(x))$. We note that it is a much stronger condition than asking for $-D$ being a square modulo a prime integer $r=r(u_0)$ for some seed u_0. For example, $-D=-2$ is not a square modulo $r(x)=\Phi_{12}(x)=x^4-x^2+1$ however is it a square modulo $r(u_0)$ where $u_0=-\texttt{0xd201000000010000}=-(2^{63}+2^{62}+2^{60}+2^{57}+2^{48}+2^{16})$ is the seed of the BLS12-381 curve. Considering the Legendre symbol and the law of quadratic reciprocity, -2 is a square modulo a prime r if and only if $r=\pm 1 \bmod 8$. Back to the polynomial form of $r(u)$, we deduce that $r(u_0)\equiv 1 \bmod 4$ for any u_0, and $r(u_0)\equiv 1 \bmod 8 \iff u_0 \not\equiv 2 \bmod 4$. However, this does not make a family. To design a family of embedded curves with $-D=-2$ for BLS12 curves, one example could be to write $r(x^2)=x^8-x^4+1$ (replace the variable x by x^2 everywhere i.e. assume the seed is a square) then apply Algorithm 2.1 with $\sqrt{-2}\equiv x^5+x^3-x \bmod r(x)$, a half-gcd gives directly $r(x)=(x^4-x^2+1)^2+2(x^3-x)$, and $(t,y)=(2(x^4-x^2+1),2(x^3-x))$.

4.1.2 Solving for polynomials (t,y) in the equation $r=(t^2+Dy^2)/4$

Sanso and El Housni suggest to compute a half-gcd of $r(x)$ and $W_e(x)$ to obtain candidates for $t_e(x), y_e(x)$ such that their degree is at most half the degree of $r(x)$. We recall that this strategy is well-known for example in cryptanalysis, in the descent step of a discrete logarithm computation. The first occurence of this technique (applied to polynomials) is for the initial splitting step of discrete logarithm computation in $\mathrm{GF}(2^n)$ and dates back to 1984. It is known under the name *Waterloo algorithm* from the University of Waterloo, ON, Canada, where the authors are from [BFHMV84, BMV84]. The idea is to express the target (a polynomial in $\mathbb{F}_2[x]$ of even degree $n-1$) as the ratio of two polynomials of degree $(n-1)/2$, modulo an irreducible polynomial of odd degree n. The aim is to increase the smoothness probability.

In the present case r has usually an even degree, and a half-gcd algorithm on inputs $(r(x), W_e(x))$ with $\deg r > \deg W_e$ outputs three polynomials $I(x), U(x), V(x)$ such that $I(x)r(x) = U(x) - V(x)W_e(x)$ with usually $\deg(I) = 1$, $\deg U, \deg V \le \deg r/2$. Luckily for BLS and BN, $I = 1$ and the equation $t^2 + Dy^2 = 4r$ is solved, with $t = 2U$ and $y = 2V$. But for KSS18 for example, $W_e = 2x^3 + 37$, $U = 3$, $V = -2x^3 - 37$, $I = 1372$.

4.2 Our general solution

We stick together different pieces that come from the litterature about elliptic curves and cryptography. In particular, we will explain the link with Smith technique [Smi15] and Dai, Lin, Zhao, and Zhou work [DLZZ23].

Finding exact integer solutions (t, y) to the equation

$$r = (t^2 + Dy^2)/4 \tag{3}$$

is linked to the problem of finding integer solutions to

$$\begin{cases} r = a_0^2 - a_0a_1 + (D+1)/4a_1^2 \text{ if } D = 3 \bmod 4 \\ r = a_0^2 + Da_1^2 \text{ otherwise.} \end{cases} \tag{4}$$

Dai, Lin, Zhao, and Zhou work over the integer values of the curve parameters. Their aim is to obtain an optimal formula for $\mathbb{G}_1$ subgroup membership testing that is, given a point P on $E(\mathbb{F}_p)$, check that $[r]P = \mathcal{O}$ without computing the full and costly scalar multiplication by r. For that, the endomorphism ϕ on the curve of characteristic polynomial χ_ϕ is used. This technique is known as the GLV method [GLV01]. The endomorphism ϕ has eigenvalue $\lambda_\phi \bmod r$. A Gaussian reduction gives two shorter scalars $a_0 + a_1\lambda_\phi \equiv 0 \bmod r$ however, as pointed out by Dai, Lin, Zhao, and Zhou, $[a_0]P + [a_1]\phi(P)$ might actually compute a small multiple $[sr]P$ instead of $[r]P$ and the test is not valid if s is not coprime to the curve cofactor. The authors of [DLZZ23] develop a criterion to test wether the short scalars (a_0, a_1) give a valid subgroup membership test. They propose an algorithm and a Magma implementation to compute the short scalars that pass the test.

We then observe that we face a very similar problem: with an elementary change of variables, finding (t, y) to define embedded curves correspond to finding the short scalars (a_0, a_1) to design a valid and optimal $\mathbb{G}_1$ subgroup membership testing. However as we are interested in defining families of embedded curves, we are interested in finding the scalars generically, parameterized by polynomials. For that we exploit Smith technique that dates back to an AGCT workshop at CIRM in Marseille Luminy in 2015 [Smi15].

We present our technique based on Smith idea for KSS16 and KSS18 curves. The general strategy follows the same procedure for other pairing-friendly curves. For these two curves the output is exactly what Dai, Lin, Zhao, and Zhou found with a Gaussian reduction on integers (Table 7).

Table 7: From [DLZZ23, Table 4], with $r = (x^8 + 48x^4 + 625)/61250$ for KSS16, $r = (x^6 + 37x^3 + 343)/343$ for KSS18.

Curve	$-D$	χ_ϕ	$\lambda \bmod r$	short vector (a_0, a_1)	criterion
KSS16	-4	X^2+1	$\sqrt{-1} = (x^4+24)/7$	$((31x^4+625)/8750, -(17x^4+625)/8750)$	$a_0^2 + a_1^2 = r$
KSS18	-3	X^2+X+1	$(-1+\sqrt{-3})/2 = x^3+18$	$((x/7)^3, -18(x/7)^3 - 1)$	$a_0^2 - a_0a_1 + a_1^2 = r$

4.2.1 Smith technique

Smith [Smi15] is interested in computing a ready-made short basis of the lattice whose long basis is given by the following $\vec{b}_i$, where λ_{ϕ_i} stands for the eigenvalue of the i-th endomorphism ϕ_i on the curve E.

$$\begin{cases} \vec{b}_1 &= (r, 0, \ldots, 0) \\ \vec{b}_2 &= (-\lambda_{\phi_2}, 1, 0, \ldots, 0) \\ \vec{b}_3 &= (-\lambda_{\phi_3}, 0, 1, 0, \ldots, 0) \\ &\vdots \\ \vec{b}_d &= (-\lambda_{\phi_d}, 0, \ldots, 0, 1) \end{cases}$$

In our case, there are two endomorphisms, $\phi_1 = \mathrm{Id}$ and $\phi_2 = \phi$. We recall [Smi15, Theorem 2].

Theorem 1 ([Smi15, Th. 2])**.** *Let ϕ be a non-integer endomorphism of $\mathcal{E}$ such that $\mathbb{Z}[\pi] \subset \mathbb{Z}[\phi]$, so $\pi = c\phi + b$ for some integers c and b. Suppose that we are in the situation of §1 with $\mathcal{A} = \mathcal{E}$ and $(\phi_1, \phi_2) = (1, \phi)$. The vectors*

$$\vec{b}_1 = (b-1, c) \text{ and } \vec{b}_2 = (c\deg(\phi) + (b-1)t_\phi, 1-b)$$

generate a sublattice of $\mathcal{L}$ of determinant $\#\mathcal{E}(\mathbb{F}_q)$. If $\mathcal{G} = \mathcal{E}(\mathbb{F}_q)$, then $\mathcal{L} = \langle \vec{b}_1, \vec{b}_2 \rangle$.

In [Smi15, Sect. 4], Smith provides a way for reducing the basis $(\vec{b}_1, \vec{b}_2)$ in case of small co-factors $h = 2$ for example, and provides a general framework for the technique.

We clarify that Smith's technique starts from the curve endomorphism and the curve coefficents and defines the basis in a context where the curve is of prime order. In our case, we know the pairing-friendly curve coefficients and we are looking for the embedded curve coefficients.

Another point of view is to look for a generator of a principal ideal in $\mathbb{Q}(\sqrt{-D})$ of norm r. It will be of the form $\tau = c\omega + b$. But again as we are working with parameters in polynomial form, we follow Smith technique.

We consider the pairing-friendly curve parameters (p, t, r, y) where p defines the field characteristic, t the curve trace, r the prime order of the subgroup of embedding degree k, and y such that $t^2 - 4p = -Dy^2$ with square-free D. We compute $\sqrt{-D}$ modulo $r(x)$ in polynomial form. Actually $\#E(\mathbb{F}_p) = cr = ((t-2)^2 + Dy^2)/4$ so $\sqrt{-D} = (t-2)/y \bmod r(x)$. Inverting $y(x)$ is done with an extended Euclidean algorithm on $r(x), y(x)$. Then we run a half-gcd algorithm to obtain $\sqrt{-D} \equiv U(x)/V(x)$ of reduced degrees and U, V coprime. At this point we introduce Smith basis reduction technique. The first vector of the basis is $\vec{b}_1 = (U(x), -V(x))$. We need to complete the basis: the second vector is $(DV(x), -U(X))$. Observe that the determinant of

$$B = \begin{bmatrix} U(x) & -V(x) \\ DV(x) & -U(x) \end{bmatrix}$$

is $\det(B) = U^2(x) + DV^2(x)$ and is a multiple of $r(x)$. For each factor ℓ of the determinant, we reduce the basis. It consists in finding a left kernel of B in $\mathbb{Z}/\ell\mathbb{Z}$. At the end of this process we expect to obtain a reduced basis whose determinant is exactly $r(x)$.

For $D = 3 \bmod 4$ and characteristic polynomial $\chi = X^2 + t_\phi X + \deg_\phi$ of discriminant $t_\phi^2 - 4\deg_\phi = -D$ with $t_\phi = 1$ and $\deg_\phi = (D+1)/4$, a variant can be used (to avoid a factor 4). Compute $(-t_\phi + \sqrt{-D})/2 = \lambda$ as $U(x)/V(x)$ modulo $r(x)$. The first vector is $(U(x), -V(x))$. The second vector is $(t_\phi U(x) + \deg\phi V(x), U(X))$ so that the determinant of the basis matrix is $U^2(x) + t_\phi U(x)V(x) + \deg\phi V^2(x)$. Once the matrix is reduced of determinant exactly r, we obtain the embedded curve coefficients from the formulas (1).

4.2.2 Application to KSS18

A curve like KSS18 with j-invariant 0 has complex multiplication (CM) by $\mathbb{Z}[(-1+\sqrt{-3})/2]$. The Frobenius is $\pi = (-t + y\sqrt{-3})/2$ so that $\pi\overline{\pi} = (t^2 + 3y^2)/4$. For the embedded curve parameters we are looking for (t_e, y_e) such that $(t_e^2 + 3y_e^2)/4 = r$. We denote $\tau = (t_e + y_e\sqrt{-3})/4$. The endomorphism ϕ on KSS18 has characteristic polynomial $\chi = X^2 + X + 1$ and its eigenvalue is $\lambda_\phi = (-1 + \sqrt{-3})/2$. We obtain $\lambda = x^3 + 18$, already of degree $\deg r/2$. No half-gcd is required. The first basis vector is $\vec{b}_1 = (x^3 + 18, -1)$ and a second vector can be $\vec{b}_2 = (1, x^3 + 19)$. We define the basis

$$\begin{bmatrix} \lambda & -1 \\ \deg\phi & \lambda+1 \end{bmatrix} = \begin{bmatrix} x^3+18 & -1 \\ 1 & x^3+19 \end{bmatrix}$$

whose determinant is $343r(x) = 7^3 \cdot r$. The aim is to reduce this basis by a factor 7^3. We are looking for a linear combination

$$(i\vec{b}_1 + j\vec{b}_2)/343 = ((j + 18i + i \cdot x^3)/343, (19j - i + j \cdot x^3)/343)$$

such that the denominator 343 will simplify and the coefficients will be integers. Note that $x \equiv 14 \bmod 21$ hence $7 \mid x$, $343 \mid x^3$ and we are looking for $i, j \in \mathbb{Z}/343\mathbb{Z}$ satisfying

$$j + 18i \equiv 0 \bmod 343 \iff 19j - i = 0 \bmod 343 \text{ indeed } 1/18 = -19 \bmod 343 \ .$$

We have a degree of freedom on j as $i = 19j \bmod 343$. We test all $1 \le j < 343$, and keep the pairs such that $\vec{b}_{i,j} = (i\vec{b}_1 + j\vec{b}_2)/343 = (a_0, a_1)$ satisfies $a_0^2 + a_0a_1 + a_1^2 = r$ (with exactly r, not a multiple). Finally we obtain a solution whose coefficients are integer-valued assuming $x \equiv 14 \bmod 21$ like for KSS18 curves.

$$(i,j) = (19,1),\ \vec{b} = (19\vec{b}_1+\vec{b}_2)/343 = ((1+19\lambda_r)/7^3, (\lambda_r+1)-19) = (19(x/7)^3+1, (x/7)^3) \ . \tag{5}$$

The pair $(a_0, a_1) = (19(x/7)^3 + 1, (x/7)^3)$ corresponds to a twist of the embedded curve given by Dai, Lin, Zhao, and Zhou parameters.

4.2.3 Application to KSS16

For KSS16 curves, the endomorphism has characteristic polynomial $\chi = X^2 + 1$. One obtains, with $\lambda_\phi = (x^4 + 24)/7$,

$$\vec{b}_1 = (1, \lambda_\phi) = (1, (x^4 + 24)/7), \quad \vec{b}_2 = (\lambda_\phi, -1) = ((x^4 + 24)/7, -1) \ .$$

The determinant of the matrix made of $\vec{b}_1$, $\vec{b}_2$ is $\det \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} = -1250r(x)$ and we are looking for a linear combination to simplify by $1250 = 2 \cdot 5^4$,

$$(i\vec{b}_1 + j\vec{b}_2)/1250 = (i + j(x^4 + 24)/7, i(x^4 + 24)/7 - j)/1250$$

such that the denominator 1250 will simplify and the coefficients will be integers. Note that $x \equiv 25, 45 \bmod 70$ hence $x \equiv 5 \bmod 10$, $5^4 \mid x^4$. With $x = 10x_0 + 5 = 5(2x_0 + 1)$),

$$\begin{aligned}(i\vec{b}_1 + j\vec{b}_2) &= (i + j(5^4(2x_0 + 1)^4 + 24)/7, i(5^4(2x_0 + 1)^4 + 24)/7 - j) \\ &= (i + j(5^4 + 24)/7, i(5^4 + 24)/7 - j) \bmod 1250\end{aligned}$$

and we are looking for $i, j \in \mathbb{Z}/2 \cdot 5^4\mathbb{Z}$ satifying

$$i + (5^4 + 24)/7j \equiv 0 \bmod 2 \cdot 5^4 \iff i + 807j \equiv 0 \bmod 2 \cdot 5^4 \ .$$

(Note that $((5^4+24)/7)^2 = -1 \bmod 2\cdot 5^4$ so that the two constraints are equivalent). We have a degree of freedom on j as $i = -807j = 443j \bmod 2\cdot 5^4$. We test the pairs (i, j) and keep those such that $(a_0, a_1) = (i\vec{b}_1 + j\vec{b}_2)$ satisfies $a_0^2 + a_1^2 = r(x)$. We obtain integer valued parameters for $x \equiv \pm 25 \bmod 70$ for KSS16:

$$\begin{aligned}(i, j) =&(31, 17),\\ \vec{b} =&(31\vec{b}_1 + 17\vec{b}_2)/1250 = ((31 + 17\lambda_\phi)/1250, (31\lambda_\phi - 17)/1250)\\ =&((17(x/5)^4 + 1)/14, (31(x/5)^4 + 1)/14)\ .\end{aligned} \tag{6}$$

Algorithm 4.1: Generating embbeded curve families with KSS16 and $-D = -4$

$r(u) \leftarrow (u^8 + 48u^4 + 625)/61250$, a KSS16 curve order;
$(t_1(u), y_1(u)) \leftarrow ((31(u/5)^4 + 1)/7, -(17(u/5)^4 + 1)/14)$;
for $(t_e(u), y_e(u))$ *in the set of 4 twist parameters of* $(t(u), y(u))$ **do**
    $q_e(u) \leftarrow r + 1 - t_e$;
    if $q_e(u)$ *is irreducible* **then**
        Append (t_e, y_e, q_e) to the list of families;
return the list of families

We give in Table 8 the results of Alg. 4.1 applied to KSS16 parameters.

Table 8: Embedded curves for KSS16, parameters (t_e, y_e) such that $r = (t_e^2 + 4y_e^2)/4$ with $-D = -4$. A first pair is $(t_1, y_1) = ((31(u/5)^4 + 1)/7, -(17(u/5)^4 + 1)/14)$ and the other pairs are for the quadratic and quartic twists. The polynomials for the orders are all irreducible but have cofactors 2, 2, 32, and 20.

	(t_e, y_e) s.t. $r = (t_e^2 + 4y_e^2)/4$	$q = r + 1 - t_e$	family
t, y	$(31(u/5)^4 + 1)/7, (-17(u/5)^4 - 1)/14$	$(u^8 - 386u^4 + 5^5 \cdot 17)/61250$	(yes, 2)
$-t, y$	$(-31(u/5)^4 - 1)/7, (-17(u/5)^4 - 1)/14$	$(u^8 + 482u^4 + 5^4 \cdot 113)/61250$	(yes, 2)
$2y, t/2$	$(-17(u/5)^4 - 1)/7, (31(u/5)^4 + 1)/14$	$(u^8 + 286u^4 + 5^4 \cdot 113)/61250$	(yes, 32)
$-2y, t/2$	$(17(u/5)^4 + 1)/7, (31(u/5)^4 + 1)/14$	$(u^8 - 190u^4 + 5^5 \cdot 17)/61250$	(yes, 20)

5 Conclusion

In this preliminary note we generalized Sanso and El Housni work on families of embedded curves for SNARKs. The source code will be made available later. Examples of embedded cycles and embedded Montgomery curves for BLS24 will be provided in a later version of this work.

Acknowledgements. This work follows a discussion with Carla Ràfols at the IMACC'23 conference at Royal Holloway in December 2023. I would like to thank Carla Ràfols, Simon Masson, Youssef El Housni and Antonio Sanso for raising to me this interesting problem of families of embedded curves, and Anca Nitulescu, Javier Silva, and Nikitas Paslis for the fruitful discussions and many shared references.

References

[AEHG22] Diego F. Aranha, Youssef El Housni, and Aurore Guillevic. A survey of elliptic curves for proof systems. *Des. Codes Cryptogr.*, Special Issue: Mathematics of Zero-Knowledge:1–46, December 2022. arXiv:2022/586, doi:10.1007/s10623-022-01135-y.

[Azt] Aztec Protocol. Aztec connect specifications. https://aztecprotocol.github.io/aztec-connect/primitives.html.

[BCG+20] Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu. ZEXE: Enabling decentralized private computation. In *2020 IEEE Symposium on Security and Privacy*, pages 947–964. IEEE Computer Society Press, May 2020. doi:10.1109/SP40000.2020.00050.

[BFHMV84] Ian F. Blake, Ryoh Fuji-Hara, Ronald C. Mullin, and Scott A. Vanstone. Computing logarithms in finite fields of characteristic two. *SIAM Journal on Algebraic Discrete Methods*, 5(2):276–285, 1984. doi:10.1137/0605029.

[BGH19] Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Report 2019/1021, 2019. https://eprint.iacr.org/2019/1021.

[BMV84] Ian F. Blake, Ronald C. Mullin, and Scott A. Vanstone. Computing logarithms in GF(2^n). In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 73–82. Springer, Heidelberg, August 1984.

[CFH+15] Craig Costello, Cédric Fournet, Jon Howell, Markulf Kohlweiss, Benjamin Kreuter, Michael Naehrig, Bryan Parno, and Samee Zahur. Geppetto: Versatile verifiable computation. In *2015 IEEE Symposium on Security and Privacy*, pages 253–270. IEEE Computer Society Press, May 2015. doi:10.1109/SP.2015.23.

[DLZZ23] Yu Dai, Kaizhan Lin, Chang-An Zhao, and Zijian Zhou. Fast subgroup membership testings for $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ on pairing-friendly curves. *Designs, Codes and Cryptography*, 91(10):3141–3166, Oct 2023. ePrint 2022/348. doi:10.1007/s10623-023-01223-7.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, April 2010. doi:10.1007/s00145-009-9048-z.

[GG23] Jean Gasnier and Aurore Guillevic. An algebraic point of view on the generation of pairing-friendly curves. HAL archive hal-04205681, September 2023.

[GLV01] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 190–200. Springer, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_11.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_11.

[Hop17a] Daira Hopwood. Jubjub supporting evidence. https://github.com/daira/jubjub, 2017.

[Hop17b] Daira Hopwood. Tweedledum/tweedledee supporting evidence. https://github.com/daira/tweedle, 2017.

[Hop20a] Daira Hopwood. Halo optimizations and constructing graphs of elliptic curves. https://github.com/daira/halographs/, 2020.

[Hop20b] Daira Hopwood. The pasta curves for halo 2 and beyond. https://electriccoin.co/blog/the-pasta-curves-for-halo-2-and-beyond/, 2020.

[Hop21] Daira Hopwood. Pluto-eris hybrid cycle of elliptic curves, 2021. https://github.com/daira/pluto-eris.

[KZM+15] Ahmed Kosba, Zhichao Zhao, Andrew Miller, Yi Qian, Hubert Chan, Charalampos Papamanthou, Rafael Pass, abhi shelat, and Elaine Shi. C∅c∅: A framework for building composable zero-knowledge proofs. Cryptology ePrint Archive, Report 2015/1093, 2015. https://eprint.iacr.org/2015/1093.

[Mec20] Izaak Meckler. O(1) labs fork of zexe: implementation of BN382-plain, 2020. https://github.com/o1-labs/zexe/tree/master/algebra/src/bn_382.

[MSZ21] Simon Masson, Antonio Sanso, and Zhenfei Zhang. Bandersnatch: a fast elliptic curve built over the BLS12-381 scalar field. Cryptology ePrint Archive, Report 2021/1152, 2021. https://eprint.iacr.org/2021/1152.

[Poe18] Andrew Poelstra. Curve with group order $2^{255} - 19$? Post on the Modern Crypto – Curves mailing list https://moderncrypto.org/mail-archive/curves/2018/000992.html, March 21 2018.

[SEH24] Antonio Sanso and Youssef El Housni. Families of prime-order endomorphism-equipped embedded curves on pairing-friendly curves. ePrint:2023/1662, 2024.

[SG18] Michael Scott and Aurore Guillevic. A new family of pairing-friendly elliptic curves. In Lilya Budaghyan and Francisco Rodríguez-Henríquez, editors, *Arithmetic of Finite Fields*, pages 43–57, Cham, 2018. Springer. arXiv:2018/193, doi:10.1007/978-3-030-05153-2_2.

[Smi15] Benjamin Smith. Easy scalar decompositions for efficient scalar multiplication on elliptic curves and genus 2 Jacobians. *Contemporary mathematics*, 637:15, May 2015. https://hal.inria.fr/hal-00874925. URL: https://hal.inria.fr/hal-00874925, doi:10.1090/conm/637/12753.