AN IMPROVEMENT TO THE KLPT^2 ALGORITHM

ABEL LAVAL

Université libre de Bruxelles
abel.laval@ulb.be

Abstract. In this note, we present a tweak to the KLPT^2 algorithm that reduces the size of the output isogeny from $O(p^{15.5})$ to $\tilde{O}(p^{14.5})$. Then, we show that the recent improvement to the KLPT algorithm by Nakagawa, Yoshizumi and Onuki is compatible with the Hermitian module splitting approach, further lowering the reduced degree bound to $\tilde{O}(p^{14})$.

1. Introduction

After the series of attacks on SIDH in August 2022 [CD22, MMP+23, Rob22], isogeny-based cryptography has seen a broad effort using the mathematical tools at the core of the attacks to systematically explore and exploit more efficient algorithms on abelian varieties in higher dimension. This led to more efficient versions of the SQIsign signature scheme [DLRW23, NOC+24, BFD+24] — the only isogeny-based competitor in the NIST competition for additional post-quantum signature schemes, but also new algorithmic methods for class group evaluation, new ideal-to-isogeny conversion or even permitted the emergence of completely new protocols altogether. One of the tools waiting to be generalized to higher dimension was the Kohel-Lauter-Petit-Tignol algorithm for the algebraic ℓ-isogeny path problem [KLPT14], which was notably used in the early versions of SQIsign and was at the heart of the algorithmic reduction between the isogeny path problem and the endomorphism ring problem [Wes21]. The higher dimension version of the algebraic ℓ-isogeny path problem is stated as follows

Problem 1. *Given an algebraic representation of two superspecial principally polarized abelian varieties (A_1, λ_1) and (A_2, λ_2) with dimension d over $\mathbb{F}_{p^2}$, find a polynomial-time algorithm that computes a polarized isogeny $\varphi : (A_1, \lambda_1) \to (A_2, \lambda_2)$ whose reduced degree is a power of a small prime $\ell \neq p$.*

Recall that a polarized isogeny $\varphi : (A_1, \lambda_1) \to (A_2, \lambda_2)$ is an isogeny between the underlying unpolarized varieties such that $\hat{\varphi}\lambda_2\varphi = N\lambda_1$ for some integer N called the reduced degree and where the map $\hat{\varphi} : \hat{A}_2 \to \hat{A}_1$ is the dual isogeny.
By *algebraic representation* we mean either the Ibukiyama-Katsura-Oort representation [IKO86] that encodes principally polarized abelian surfaces as 2-by-2 matrices or the unimodular Hermitian module point-of-view developed in [Rob24]. The former was used by [CDK+25] to provide a first solution to Problem 1 in the dimension 2 case and has not been formally generalized to other dimensions at this point; the latter is at the base of the improvement brought by [PRS26] and encompasses abelian varieties of any dimension.
We refer the reader to [CDK+25, Section 2] for an exposition of the Ibukiyama-Katsura-Oort correspondence and to [Rob24] for unimodular Hermitian modules. In Sections 2 and 3, we heavily rely on notations introduced in [CDK+25, Sections 2 and 3] and [PRS26, Section 7], respectively.

A variant of Problem 1 asks to find an isogeny whose norm is powersmooth. All solutions proposed thus far solved both variants with minimal adaptation and we follow the same route by focusing solely on the power-of-a-prime variant.

Besides the usual discussion on the asymptotically expected running time and the potential resort to heuristics, performance of algorithms solving Problem 1 is measured in terms of the size of the output isogeny. Finding a better solution to the problem often means reducing the size of said output.

1.1. State-of-the-art and contributions

Besides the dimension one case that was solved by the original KLPT paper in 2014, Problem 1 has to this day only been solved in dimension $d = 2$. The first solution was given in [CDK+25] with an algorithm outputting a very large isogeny, whose reduced degree was $O(p^{25})$. Later, Page, Robert and Soumier proposed a different approach exploiting extra degrees of freedom provided by the abstract unimodular Hermitian module representation, while reusing several subroutines of the original algorithm. The result was an improved reduced degree of size $O(p^{15.5})$ [PRS26]. The variant we present in this note preserves the structure of the algorithm as found in [PRS26] and optimizes one of the original subroutines, leading to a new output bound of $\tilde{O}(p^{14.5})$.

In parallel, Nakagawa and Yoshizumi [NY26] discovered a new variant of dimension one KLPT they called Δ-KLPT. Their new method computes a connecting ideal between two arbitrary maximal orders with reduced norm $\tilde{O}(p^4)$, compared to the $O(p^{4.5})$ that constituted the state of the art since the first SQIsign paper. The downside of their algorithm is that its input ideals have to satisfy several constraining requirements, but a few months later, Nakagawa and Onuki [NO26] updated the algorithm — and the name, which became Δ-KLPT$^+$ — with a phase of pre-processing and rerandomization of the output, removing said requirements. As we will see in Section 3, this new version of KLPT can be incorporated into KLPT^2, offering a final output size of $\tilde{O}(p^{14})$.

The two improvements are summarized in the diagrams below, highlighting the contribution of each step to the size of the output isogeny.
The solution described in [PRS26] gave the following composition of (module) isogenies:

$$(M_1, H_{M_1}) \xrightarrow[\text{Splitting}]{O(p^{5.5})} (I_1 \oplus J_1) \xrightarrow[\text{KLPT}]{O(p^{4.5})} (I_2 \oplus J_2) \xrightarrow[\text{Splitting}]{O(p^{5.5})} (M_2, H_{M_2})$$

This work turns it into:

$$(M_1, H_{M_1}) \xrightarrow[\text{Splitting}]{\tilde{O}(p^{5})} (I_1 \oplus J_1) \xrightarrow[\text{Δ-KLPT}^+]{\tilde{O}(p^{4})} (I_2 \oplus J_2) \xrightarrow[\text{Splitting}]{\tilde{O}(p^{5})} (M_2, H_{M_2})$$

which gives a reduction from $O(p^{15.5})$ to $\tilde{O}(p^{14})$ as stated.

2. The improvement

In this section, we show how we gain a factor $\tilde{O}(\sqrt{p})$ off of the two splitting isogenies by adding a small tweak to the subroutine solving the following problem:

Problem 2. *Given an IKO matrix $g = \left(\begin{smallmatrix} s & r \\ \bar{r} & t \end{smallmatrix}\right)$ with $s, t \in \mathbb{Z}_{\geqslant 0}$ and $r \in \mathcal{O}_0$ such that $s = O(\sqrt{p})$ is a prime that does not divide $pt\,\mathsf{n}(r)$ and ℓ does not divide $\mathsf{n}(r)$, find $a, c \in \mathcal{O}_0$ such that*

$$s\,\mathsf{n}(a) + t\,\mathsf{n}(c) + \mathsf{tr}(ar\bar{c}) = \ell^e \tag{1}$$

for some positive integer e.

The non-divisibility condition on s is required to ensure Equation 1 can be solved, while the condition $\ell \nmid \mathsf{n}(r)$ guarantees that the solution gives a primitive vector later on.

A *good* solution for this problem is one that minimizes ℓ^e for prescribed inputs s, t, r. In the context of KLPT^2 with Hermitian modules, solving Problem 2 makes it possible to compute a vector $v = \mathcal{O}_0(a, c)$ inducing a splitting isogeny[1] of smooth degree between a principally polarized abelian surface (A, λ) and a product of elliptic curves $E_0 \times E$, where $E_0 : y^2 = x^3 + x$ has j-invariant 1728.

[1]Which — through the contravariant equivalence between the categories of unimodular Hermitian modules and that of principally polarized abelian varieties — comes from a smooth similitude $\mathcal{O}_0 x + \mathcal{O}_0 y \to (M, H_M)$. Note that we omit the canonical Hermitian form in the notation of the left side.

The reduced degree of that splitting isogeny is exactly ℓ^e.[2] Our new method lowers the reduced degree of that splitting isogeny by a factor $\sqrt{p}/\log(p)^2$, which ultimately leads to a factor $\tilde{O}(p)$ gain for the whole algorithm.

Solving Problem 2. To solve Equation 1, the usual approach consists in forcing the trace to vanish by taking $a \in \mathbb{Z}[i]$ and $c \in \mathbb{Z}[i]jr$. Write $\mathsf{n}(a) = a_1^2 + a_2^2$ and $\mathsf{n}(c) = \mathsf{n}((c_1 + c_2 i)jr) = p\,\mathsf{n}(r)(c_1^2 + c_2^2)$ so Equation 1 only has four integral unknowns (a_1, a_2, c_1, c_2). By reducing the equation modulo s, the dependence in a disappears and we are left with only (c_1, c_2) in a simpler equation we can solve directly:

$$pt\,\mathsf{n}(r)(c_1^2 + c_2^2) = \ell^e \mod s \tag{2}$$

Solving this equation efficiently is the heart of our improvement. Among the several admissible tuples (c_1, c_2), we will look for one such that $\frac{1}{s}(\ell^e - pt\,\mathsf{n}(r)(c_1^2 + c_2^2))$ is a prime equal to 1 mod 4, so that we can ultimately compute (a_1, a_2) using Cornacchia's algorithm:

$$a_1^2 + a_2^2 = \frac{\ell^e - pt\,\mathsf{n}(r)(c_1^2 + c_2^2)}{s} \tag{3}$$

It is clear that this equation admits a solution only if the right-hand side is positive, which means we can take e to be the smallest integer such that

$$e \geqslant \lceil \log_\ell \left(pt\,\mathsf{n}(r)(c_1^2 + c_2^2)\right) \rceil \tag{4}$$

With $t = O(p^{1.5})$, $\mathsf{n}(r) = O(p^2)$ as in [PRS26] and with the bound $c_1^2 + c_2^2 = O(\sqrt{p}\log(p)^2)$ constructed below, we obtain the estimate $\ell^e = O(p^5 \log(p)^2) = \tilde{O}(p^5)$.

In [CDK+25], it was shown that Equation 2 has in fact many solutions and the coarse bound $c_i < s$ was used to obtain $c_1^2 + c_2^2 < 2s^2 = O(p)$. The algorithm picks random solutions until Equation 3 is solvable. This step required heuristically $O(1)$ iterations with no effect on the size of $c_1^2 + c_2^2$. Additionally, the original KLPT^2 algorithm imposes the constraint that $\mathsf{n}(a)$ and $\mathsf{n}(c)$ be coprime, which induces at worst a $\log(s)$ time overhead as several candidates for c have to be tested before this extra requirement is met.

In what follows, an integer N is said to be *Cornacchiable*[3] if it is a prime equal to 1 modulo 4, as such integers can easily be represented as a sum of two squares with Cornacchia's algorithm.
Even though a much larger class of integers is also compatible with the algorithm, this one has the practical advantage of having an efficient membership test that does not require factoring N.

2.1. Our improvement

Because s is assumed to be prime, Equation 2 can be rewritten as

$$c_1^2 + c_2^2 = \ell^e (pt\,\mathsf{n}(r))^{-1} \mod s \tag{5}$$

Equation 5 is well-defined if and only if s does not divide $(pt\,\mathsf{n}(r))$, which is the case, by hypothesis. For any lift $x \in \mathbb{Z}$ of the right-hand side $\tilde{x} = \ell^e (pt\,\mathsf{n}(r))^{-1} \in \mathbb{Z}/s\mathbb{Z}$, we have

$$a_1^2 + a_2^2 = \frac{\ell^e - pt\,\mathsf{n}(r)x}{s} \tag{6}$$

As we can see, the only degree of freedom we have in Equation 6 is on x. Write $\mathsf{int} : \mathbb{Z}/s\mathbb{Z} \to \mathbb{Z}$ for the lift $[x] \mapsto x$ where $[x]$ is the unique representative of x modulo s in the interval $\{0, \ldots, s-1\}$.
Our strategy is to compute a sequence of tuples $((x_k, y_k))_{k\in\mathbb{N}}$ with $x_k \in \mathbb{Z}$ and $y_k = \frac{1}{s}(\ell^e - pt\,\mathsf{n}(r)x_k)$ such that all the following conditions are met:

(1) for all $k \geqslant 0$, $x_k = \ell^e (pt\,\mathsf{n}(r))^{-1} \mod s$
(2) for all $k \geqslant 0$, $x_k = 1 \mod 4$ and $y_k = 1 \mod 4$

[2]Note that Problem 2 was first formulated and solved in the original KLPT^2 paper, but with a different goal in mind. Instead of a smooth splitting isogeny, the vector (a, c) acted as the first column of a smooth isogeny between two IKO matrices: $u' : \left(\begin{smallmatrix} s & r \\ \bar{r} & t \end{smallmatrix}\right) \mapsto \left(\begin{smallmatrix} \ell^e & \bullet \\ \bullet & \bullet \end{smallmatrix}\right)$. Then, Lemma 3.3 of [CDK+25] was used to connect the two ends.

[3]Some might argue that, Giuseppe Cornacchia being Italian, we should say *Cornacchiabile* instead. We leave this problem open for future work.

(3) Both x_k and y_k are prime

The first condition and the definition of y_k follow from the problem itself. The second and third conditions together imply exactly that both x_k and y_k are Cornacchiable.
Our strategy is to construct the sequence $((x_k, y_k))_{k\in\mathbb{N}}$ such that the first and second conditions are met by construction for all k. Then it suffices to increase k until the third is met as well.
Once a valid tuple (x_k, y_k) is found, we can compute $c_1^2 + c_2^2 = \mathsf{Cornacchia}(x_k)$ and $a_1^2 + a_2^2 = \mathsf{Cornacchia}(y_k)$, before outputting $(a_1 + a_2 i, (c_1 + c_2 i)jr)$ as a solution to Problem 2.

Forcing x_k and y_k into the correct congruence class modulo 4. Let $x_0 = \mathsf{int}(\tilde{x}) + \varepsilon s \in \mathbb{Z}$, where $\varepsilon \in \{0, 1, 2, 3\}$ is chosen so that $x_0 = 1 \bmod 4$. Taking

$$x_k = x_0 + 4ks, \forall k \in \mathbb{Z}$$

ensures that x_k verifies (1) and $x_k = 1 \bmod 4$, for all k.
Now, let us see how $y_k \bmod 4$ can be controlled. We have:

$$y_k = \frac{\ell^e - pt\,\mathsf{n}(r)x_k}{s} = s(\ell^e + t\,\mathsf{n}(r)) \mod 4$$

since $s = s^{-1} \bmod 4$, $p = 3 \bmod 4$ and $x_k = 1 \bmod 4$.
Because ℓ is a fixed parameter, we have no control over it and the only degree of freedom at our disposal is the input IKO matrix $g = \left(\begin{smallmatrix} s & r \\ \bar{r} & t \end{smallmatrix}\right)$. Even before starting the actual computation of (a, c) a pre-processing step can check whether or not $s(\ell^e + t\,\mathsf{n}(r))$ equals 1 modulo 4. If it is the case, we can proceed. Otherwise, we re-randomize g by applying to it an automorphism $u_\alpha = \left(\begin{smallmatrix} 1 & 0 \\ \alpha & 1 \end{smallmatrix}\right)$:

$$g' = \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix} \begin{pmatrix} s & r \\ \bar{r} & t \end{pmatrix} \begin{pmatrix} 1 & \bar{\alpha} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} s & s\bar{\alpha} + r \\ s\alpha + \bar{r} & \frac{\mathsf{n}(s\alpha + \bar{r}) + d}{s} \end{pmatrix}$$

where $d = \det(g)$ is a power of ℓ. A rigorous but tedious case-by-case analysis would give us one suitable value for $\alpha \in \mathcal{O}_0$ for any possible combination of $s, t, \mathsf{n}(r), \ell, d$ modulo 4. Instead, we heuristically expect to find a suitable α after $O(1)$ attempts by enumerating quaternions of small norm.
Applying the automorphism u_α has repercussions on our solution (a, c), which was meant to induce a splitting similitude $\mathcal{O}_0(a, c) \oplus J \hookrightarrow M$, where $M = \mathcal{O}_0 b_1 + \mathcal{O}_0 b_2$ is the Hermitian module corresponding to our IKO matrix g. But after the change of basis, what we have is instead

$$\mathcal{O}_0(a, c) \oplus J' \hookrightarrow M' \xrightarrow{\sim} M$$

where M' is just M written with a different basis (b_1', b_2') induced by $g' = u_\alpha g u_\alpha^*$. The change of basis is explicit: $b_1' = b_1$, $b_2' = \alpha b_1 + b_2$. Thus, the vector we are really looking for is

$$ab_1' + cb_2' = ab_1 + c(\alpha b_1 + b_2) = (a + c\alpha)b_1 + cb_2$$

Note that the case where y_k is already equal to 1 modulo 4 without requiring any randomization corresponds to $\alpha = 0$ and the isometry is just the identity.
Once the randomization step is done, both x_k and y_k equal 1 modulo 4 for any value of k, and both conditions (1) and (2) are satisfied.

Probing in the arithmetic progression $x_k = x_0 + 4ks$. Note that at this point, y_k is entirely determined by x_k. Our approach to meet the last condition is simple: increase k until both x_k and y_k are prime. At each increment, the size of x_k increases and so does $c_1^2 + c_2^2$. Our whole improvement to KLPT^2 comes down to the fact that we can find a prime pair (x_k, y_k) before the size of $c_1^2 + c_2^2$ exceeds $O(p)$.

The prime number theorem for arithmetic progressions states that the number of primes smaller than N in $(x_k)_{k\geqslant 0}$ is heuristically

$$\pi(N) \sim \frac{1}{\varphi(4s)} \cdot \frac{N}{\log(N)} \approx \frac{N}{2s\log(N)}$$

where φ is Euler's totient function. We obtain the heuristic probability for an integer $n \leqslant N$ to be a prime by dividing by the number of integers in the progression, which gives

$$\mathbb{P}(n \leqslant N \text{ is prime}) \approx \frac{N}{2s\log(N)} \cdot \frac{4s}{N} \approx \frac{2}{\log(N)}$$

This probability applies to both of our conditions and the conjunction is given by

$$\mathbb{P}(x_k \leqslant N_1 \text{ is prime}) \cdot \mathbb{P}(y_k \leqslant N_2 \text{ is prime}) \approx \frac{4}{\log(N_1)\log(N_2)} \tag{7}$$

where N_1, N_2 are the respective bounds for each condition. Note that here we assumed that heuristically, x_k and y_k behave as independent integers.
Our last task will be to estimate a lower bound for N_1, N_2. Since we are conducting a single search for both primes, we would like to work with a uniform bound N by keeping the largest of the two, that is taking $N = \max(N_1, N_2)$. Then, inverting Equation 7 with this uniform bound, we obtain the expected number of attempts k before both the values are prime:

$$k \approx \frac{\log(N)^2}{4}$$

Because $x_0 < 4s$ and since each attempt increases x_k by $4s$, its expected final size is

$$x_k \approx 4s\frac{\log(N)^2}{4} = s\log(N)^2 \tag{8}$$

Moreover, y_k is by definition smaller than $\lfloor \ell^e/s \rfloor$, which — in the setting of Page, Robert and Soumier's solution to KLPT^2 — is $\tilde{O}(p^{4.5})$. Indeed, in [PRS26, Section 7.1] the entries of the input IKO matrix verify: $s = O(p^{0.5})$, $t = O(p^{1.5})$ and $\mathsf{n}(r) = O(p^2)$. Thus, ℓ^e being positive, we have the estimate

$$\ell^e \geqslant pt\,\mathsf{n}(r)x_k = O(p \cdot p^{1.5} \cdot p^2 \cdot p^{0.5}\log(p)^2) = \tilde{O}(p^5) \tag{9}$$

and $N \approx \ell^e/s = \tilde{O}(p^{4.5})$[4] .
From this analysis we deduce that picking $N \approx p^{4.5}$ both guarantees that all potential candidates for y_k can be tested while being large enough for the search to succeed with overwhelming probability.
Combining everything, we obtain:

$$x_k \approx s\log(p^{4.5})^2 \approx \frac{81}{4}\sqrt{p}\log(p)^2 \tag{10}$$

and we have $c_1^2 + c_2^2 = O(\sqrt{p}\log(p)^2)$ as stated. This reduction of $c_1^2 + c_2^2$ is the modification that produces the bound $\ell^e = \tilde{O}(p^5)$, to be compared with the $O(p^{5.5})$ of [PRS26].

Outputting the smooth splitting similitude. Once x_k and y_k satisfying all of our three conditions are found, we obtain our vector v inducing a smooth splitting similitude $\mathcal{O}_0 v \oplus J \hookrightarrow M$ from direct computations. First, compute $(c_1, c_2) = \mathsf{Cornacchia}(x_k)$ and $(a_1, a_2) = \mathsf{Cornacchia}(y_k)$, then recombine the integers into quaternions as $a = a_1 + a_2 i$, $c = (c_1 + c_2 i)jr$, and finally return the vector in the modified basis computed in the preprocessing phase

$$\mathcal{O}_0 v = \mathcal{O}_0(a + c\alpha, c)$$

Everything so far is summed up in Algorithm 1. One extra feature of our method is that it guarantees for free that $\mathsf{n}(a)$ and $\mathsf{n}(c)$ are coprime. This condition is necessary for v to be primitive and for the unimodular Hermitian module M to be equal to the direct sum $\mathcal{O}_0 v \oplus (\mathcal{O}_0 v)^\perp$.

[4]There is a seemingly circular argument here because the size of N depends on y_k and ultimately on x_k but the size of x_k itself depends on N... In fact, we have a positive feedback loop of logarithmic intensity between the two, but with a rapid dampening effect. In practice, taking $N \approx p^{4.5}$ makes this effect unnoticeable.

Algorithm 1: Compute_ac

Input : $g = \left(\begin{smallmatrix} s & r \\ \bar{r} & t \end{smallmatrix}\right)$ as in Problem 2, encoding a Hermitian module (M, H_M). A basis (b_1, b_2) of M.

Output: A vector $v = (a + c\alpha)b_1 + cb_2$ inducing a smooth similitude $\mathcal{O}_0 v \oplus (\mathcal{O}_0 v)^\perp \hookrightarrow M$

$e \leftarrow \lceil \log_\ell(p^5 \log(p)^2) \rceil$

$\alpha \leftarrow 0$ `// default initialization`

`// Pre-processing phase`

While $s(\ell^e + t\,\mathsf{n}(r)) \neq 1 \bmod 4$ **do**

  $\alpha \xleftarrow{\$} \mathcal{O}_0$ `// Enumerate non-zero quaternions with small norm`

  $g' = \left(\begin{smallmatrix} s' & r' \\ \bar{r}' & t' \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} 1 & 0 \\ \alpha & 1 \end{smallmatrix}\right)\left(\begin{smallmatrix} s & r \\ \bar{r} & t \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 & \bar{\alpha} \\ 0 & 1 \end{smallmatrix}\right)$

  If *($s'(\ell^e + t'\,\mathsf{n}(r')) = 1 \bmod 4$ and g' satisfies the hypotheses of Problem 2)* **then**

    $(s, r, t) \leftarrow (s', r', t')$

`// Finding prime` x_k `and` y_k

While *True* **do**

  $x \leftarrow \mathsf{int}(\ell^e (pt\,\mathsf{n}(r))^{-1} \mod s)$

  $x \leftarrow x + \varepsilon s$ `// with` $\varepsilon \in \{0, 1, 2, 3\}$ `such that` $x + \varepsilon s = 1 \mod 4$

  While $(\ell^e \geqslant pt\,\mathsf{n}(r)x)$ **do**

    If $\left(x \text{ and } \frac{\ell^e - pt\,\mathsf{n}(r)x}{s} \text{ are Cornacchiable}\right)$ **then**

      $(c_1, c_2) \leftarrow \mathsf{Cornacchia}(x)$

      $(a_1, a_2) \leftarrow \mathsf{Cornacchia}((\ell^e - tp\,\mathsf{n}(r)x)/s)$

      $a = (a_1 + ia_2), \quad c = (c_1 + ic_2)jr$

      Return $(a + c\alpha)b_1 + cb_2$

    $x \leftarrow x + 4s$

  $(e \leftarrow e + 1)$ if $\ell = 1 \bmod 4$, $\quad (e \leftarrow e + 2)$ otherwise `//` $\ell^e \bmod 4$ `stays fixed.`

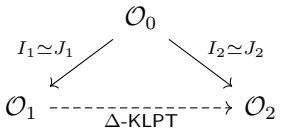
3. Delta-KLPT+

In this section, we show that the new one-dimensional KLPT variant proposed in [NY26] — named Δ-KLPT — and its enhancement Δ-KLPT$^+$ [NO26] fit in the KLPT^2 framework, allowing for another reduction of the output size. While Δ-KLPT requires very specific ideal inputs to work and is not readily usable in PRS, the addition of Δ-KLPT$^+$ makes it possible to compute smooth connecting ideals $I : \mathcal{O}_1 \to \mathcal{O}_2$ between two generic maximal orders as long as arbitrary connecting ideals $I_1 : \mathcal{O}_0 \to \mathcal{O}_1$ and $I_2 : \mathcal{O}_0 \to \mathcal{O}_2$ are provided. With a few extra preprocessing steps, Δ-KLPT$^+$ becomes compatible with the output of Algorithm 1 and brings another reduction of the KLPT^2 output, from $\tilde{O}(p^{14.5})$ to $\tilde{O}(p^{14})$.

The inputs of Δ-KLPT are two connecting ideals $I_i : \mathcal{O}_0 \to \mathcal{O}_i$, that is, both ideals are left $\mathcal{O}_0$-ideals and respectively have $\mathcal{O}_1$ and $\mathcal{O}_2$ as their right-order and are such that $\mathsf{n}(I_1) = \mathsf{n}(I_2) = N$, for some prime N congruent to 3 modulo 4. On top of that, a quadratic residue condition detailed in Nakagawa and Onuki's paper is required. What Δ-KLPT$^+$ does is taking the generic ideals J_1, J_2 obtained by the Hermitian module splitting routine and computing $I_1 \simeq J_1$, $I_2 \simeq J_2$ that meet the Δ-KLPT requirements.

Let us now take the SplittingModule algorithm from ([PRS26, Algorithm 5]) as a blackbox and have a closer look at what it really gives us. The interactions between the different routines can be summed

up in a single diagram:



Interpreting the output of the splitting algorithm. Let (A_1, λ_1), (A_2, λ_2) be our two input principally polarized abelian surfaces for the KLPT^2 algorithm. They correspond to two unimodular Hermitian modules (M_1, H_{M_1}) and (M_2, H_{M_2}). The SplittingModule algorithm outputs ℓ^e-similitudes $(\mathcal{O}_0 v_i \oplus J_i, \frac{1}{\ell^e} H_{M_i|\mathcal{O}_0 v_i} \oplus \frac{1}{\ell^e} H_{M_i|J_i}) \hookrightarrow (M_i, H_{M_i})$, for $i \in \{1, 2\}$ and where $H_{M|M'}$ is the restriction of H_M to $M' \subseteq M$. The rank 1 modules $\mathcal{O}_0 v_i$ are principal ideals by construction and therefore, through the equivalence of categories described in [Rob24, Corollary 4.27], they correspond to the curve E_0 itself. On the other hand, the orthogonal rank 1 modules $J_i \simeq M_i/\mathcal{O}_0 v_i$ technically correspond not to a single ideal but rather to one of the two classes of non-principal ideals with right order $\mathcal{O}_i$. This is because the splitting similitude determines a class of isogenies $A_i \to E_0 \times E_i$ instead of a particular isogeny and we are actually free to choose the one we want (see Lemma A.1 below)[5]. Unless $\mathcal{O}_i$ is $\mathfrak{p}$-extremal (that is, if it contains a copy of $\mathbb{Z}[j]$ as a subring), there are two such classes: one encodes isogenies with codomain E_i; the other, isogenies with codomain the Galois conjugate E_i^π. Without loss of generality, we call E_i the codomain curve associated to the ideal class J_i.
For the sake of clarity, we will abuse notation and keep writing J_i for what is in reality a whole class of ideals. This notation choice can be interpreted as an implicit fixed choice of an ideal in the class, and indeed, it was shown in [CDK+25, Lemma 3.7] that a complementary left-ideal of some $\mathcal{O}_0 v \subset M$ for $v = (a, c)$ was simply $J = \mathcal{O}_0\, \mathsf{n}(c) + \mathcal{O}_0 \bar{c} a$[6]. This means that after obtaining $v = (a, c)$ with Algorithm 1, the orthogonal ideal J such that $\mathcal{O}_0 v \oplus J \hookrightarrow M$ is immediately given with a representation suited for Δ-KLPT$^+$ and we are left with two products we have to connect:

$$E_0 \times E_1 \xrightarrow{\Phi} E_0 \times E_2$$

This step corresponds to the central arrow of the diagrams in Section 1.1.

Computing a smooth isogeny between the products. After the splitting phase, all that is left to do is to compute an $(\ell^{e'}, \ell^{e'})$-isogeny $\Phi : E_0 \times E_1 \to E_0 \times E_2$. The simplest way to do that is to take

$$\Phi = \begin{pmatrix} \varphi_3' & 0 \\ 0 & \varphi_3 \end{pmatrix}$$

where both $\varphi_3' \in \mathrm{End}(E_0)$ and $\varphi_3 : E_1 \to E_2$ have degree $\ell^{e'}$. Computing such an isogeny φ_3 is exactly what Δ-KLPT$^+$ does, with $\ell^{e'} = O(p^4)$; and finding a smooth endomorphism of E_0 is a simple norm equation to solve which is always possible in this case since the required degree is much larger than p^2. Combining our tweak in Algorithm 1 and the replacement of SQIsign's KLPT by Δ-KLPT$^+$, we solve the algebraic ℓ-isogeny path problem and output a connecting matrix of reduced degree $\tilde{O}(p^{14})$.

References

[BFD+24] Andrea Basso, Luca De Feo, Pierrick Dartois, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D-West: The Fast, the Small, and the Safer. Cryptology ePrint Archive, Paper 2024/760, 2024.

[CD22] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. Cryptology ePrint Archive, Paper 2022/975, 2022.

[5] Theoretically, J_i has a negligible chance to degenerate into a principal ideal, when the splitting is $A_i \to E_0^2$. In this case we don't even need a general KLPT algorithm and the Petit-Smith [PS18] variant gives a cleaner and shorter solution.

[6] The paper actually considers the right $\mathcal{O}_0$-ideal $\mathsf{n}(c)\mathcal{O}_0 + a\bar{c}\mathcal{O}_0$ but since we want left ideals for Δ-KLPT$^+$, we take its conjugate, which is $\mathcal{O}_0\, \mathsf{n}(c) + \mathcal{O}_0 \bar{c} a$ in our left-module convention.

[CDK+25] Wouter Castryck, Thomas Decru, Péter Kutas, Abel Laval, Christophe Petit, and Yan Bo Ti. KLPT^2: Algebraic pathfinding in dimension two and applications. 2025.

[DLRW23] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: New dimensions in cryptography. Cryptology ePrint Archive, Paper 2023/436, 2023.

[IKO86] Tomoyoshi Ibukiyama, Toshiyuki Katsura, and Frans Oort. Supersingular curves of genus two and class numbers. *Compositio Mathematica*, 57(2):127–152, 1986. `http://eudml.org/doc/89752`.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014. `https://doi.org/10.1112/S1461157014000151`.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. Cryptology ePrint Archive, Paper 2023/640, 2023.

[NO26] Kohei Nakagawa and Hiroshi Onuki. Δ-KLPT$^+$: Improved norm bounds for the quaternion isogeny path problem and an application to ring signatures. Cryptology ePrint Archive, Paper 2026/2108, 2026.

[NOC+24] Kohei Nakagawa, Hiroshi Onuki, Wouter Castryck, Mingjie Chen, Riccardo Invernizzi, Gioella Lorenzon, and Frederik Vercauteren. SQIsign2D-East: A new signature scheme using 2-dimensional isogenies, 2024.

[NY26] Kohei Nakagawa and Ryo Yoshizumi. Δ-SQIsign: A new isogeny-based signature scheme using degree challenges. Cryptology ePrint Archive, Paper 2026/1166, 2026.

[PRS26] Aurel Page, Damien Robert, and Julien Soumier. On the conversion of module representations for higher dimensional supersingular isogenies. Cryptology ePrint Archive, Paper 2026/276, 2026.

[PS18] Christophe Petit and Spike Smith. An improvement to the quaternion analogue of the ℓ-isogeny problem. *Presentation at MathCrypt*, 2018.

[Rob22] Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, Paper 2022/1038, 2022.

[Rob24] Damien Robert. The module action for isogeny based cryptography. Cryptology ePrint Archive, Paper 2024/1556, 2024.

[Wes21] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. Cryptology ePrint Archive, Paper 2021/919, 2021.

Appendix A. The orthogonal of $\mathcal{O}_0 v$

As a goal for future work, it is interesting to remark that the choice of J by [CDK+25] is not canonical. In fact, J is only one of the many representatives of the ideal class $(\mathcal{O}_0 v)^\perp$.
If one were able to select a specific ideal in the class that fits the Δ-KLPT requirements, the algorithm could be applied right away without requiring the pre-processing introduced in Δ-KLPT$^+$. In this section, we characterize $(\mathcal{O}_0 v)^\perp$ as an ideal class and see whether or not picking specific ideal representatives could let us get rid of Δ-KLPT$^+$ altogether.
To express the ideal class, we use Lemma A.1 below but to make its exposition understandable, let us recall some points of the theory of Hermitian modules.

Concrete PPAS representation from unimodular Hermitian modules. Let (A, λ) be a principally polarized superspecial abelian surface. The module we work with is $M = \mathrm{Hom}(A, E_0)$, equipped with the Hermitian form $H_M(x, y) = x\bar{y} \in \mathcal{O}_0$. The solution to the Principal Ideal Problem presented in [PRS26, Section 4] makes it possible to compute an explicit basis (b_1, b_2) for the left $\mathcal{O}_0$-module M. As elements of M, the b_i are maps from A to E_0 and they admit an adjoint $\tilde{b}_i : E_0 \to A$, which is compatible with the polarization λ. This adjoint is actually closely related to λ since the Gram matrix $S = (H_M(b_i, b_j))_{i,j\in\{1,2\}}$ can be written as

$$S = \begin{pmatrix} s & r \\ \bar{r} & t \end{pmatrix}$$

with $s, t \in \mathbb{Z}_+, r \in \mathcal{O}_0$ and $\det(S) = st - \mathsf{n}(r) = 1$. This matrix S is called an IKO matrix (see [CDK+25, Section 2]) and it encodes the polarization λ.

The Hermitian form H_M can be expressed in the basis (b_1, b_2). Concretely, if $x = x_1 b_1 + x_2 b_2$ and $y = y_1 b_1 + y_2 b_2$, then

$$H_M(x, y) = (x_1, x_2) S (\bar{y}_1, \bar{y}_2)^t$$

Note that if we take $v = ab_1 + cb_2$, then $H_M(v, v) = s\,\mathsf{n}(a) + t\,\mathsf{n}(c) + \mathsf{tr}(ar\bar{c})$ as in Equation 1.
In an attempt to simplify the proof of Lemma A.1, we also introduce a change of basis (b_1^*, b_2^*) that incorporates S in x.

$$\begin{aligned} H_M(x, y) &= (x_1, x_2) \begin{pmatrix} s & r \\ \bar{r} & t \end{pmatrix} \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix} \\ &= \underbrace{(x_1 s + x_2 \bar{r})}_{x_1' :=} \bar{y}_1 + \underbrace{(x_1 r + x_2 t)}_{x_2' :=} \bar{y}_2 \end{aligned}$$

The change of variables $(x_1, x_2) \to (x_1', x_2')$ is bijective since S has determinant 1. In particular, we have

$$x_1 = x_1' t - x_2' \bar{r}, \quad x_2 = -x_1' r + x_2' s$$

Hence, x is expressed in the basis (b_1, b_2) as

$$\begin{aligned} x &= (x_1' t - x_2' \bar{r}, -x_1' r + x_2' s)(b_1, b_2)^t \\ &= (x_1', x_2') \left(\begin{smallmatrix} t & -r \\ -\bar{r} & s \end{smallmatrix}\right) (b_1, b_2)^t \\ &= (x_1', x_2') \underbrace{S^{-1} (b_1, b_2)^t}_{(b_1^*, b_2^*)^t} \end{aligned}$$

With this change of variable, we can simply write:

$$H_M(x, y) = x_1 \bar{y}_1 + x_2 \bar{y}_2$$

for $x = x_1 b_1^* + x_2 b_2^*$ and $y = y_1 b_1 + y_2 b_2$.
Let us now use this machinery to prove the following result:

Lemma A.1. *Let $M = \mathcal{O}_0 b_1 + \mathcal{O}_0 b_2$ be a unimodular Hermitian module. Let $v = (a, c) \in M$ with $\mathsf{n}(a)$ and $\mathsf{n}(c)$ coprime (which implies that v is primitive). A vector $x = x_1 b_1^* + x_2 b_2^* \in M$ is orthogonal to $\mathcal{O}_0 v$ if and only if $H_M(x, v) = x_1\bar{a} + x_2\bar{c}$ is zero. The set of solutions is generated by the two vectors $(-\bar{c}a, \mathsf{n}(a))$ and $(\mathsf{n}(c), -\bar{a}c)$ and we have:*

$$(\mathcal{O}_0 v)^\perp = \mathcal{O}_0(-\bar{c}a, \mathsf{n}(a)) + \mathcal{O}_0(\mathsf{n}(c), -\bar{a}c)$$

Proof. We check with a direct computation that $(-\bar{c}a, \mathsf{n}(a)) \cdot (\bar{a}, \bar{c})^t$ and $(\mathsf{n}(c), -\bar{a}c) \cdot (\bar{a}, \bar{c})^t$ equal 0 and therefore both $K_1 := \mathcal{O}_0(-\bar{c}a, \mathsf{n}(a))$ and $K_2 := \mathcal{O}_0(\mathsf{n}(c), -\bar{a}c)$ are included in $(\mathcal{O}_0 v)^\perp$, which gives the first inclusion $K_1 + K_2 \subseteq (\mathcal{O}_0 v)^\perp$.
Now, we show the opposite inclusion by considering $x = (x_1, x_2) \in M$ such that $x_1\bar{a} + x_2\bar{c} = 0$ and express it as an element in $K_1 + K_2$. By Bézout's theorem, let us write $u\,\mathsf{n}(a) + w\,\mathsf{n}(c) = 1$ for $u, w \in \mathbb{Z}$. Then we can write

$$\begin{aligned} x &= (x_1, x_2)(u\,\mathsf{n}(a) + w\,\mathsf{n}(c)) \\ &= (ux_1\,\mathsf{n}(a) + wx_1\,\mathsf{n}(c),\ ux_2\,\mathsf{n}(a) + wx_2\,\mathsf{n}(c)) \end{aligned}$$

Now, the orthogonality condition gives

$$\begin{cases} x_1\bar{a} &= -x_2\bar{c} \\ -x_1\bar{a} &= x_2\bar{c} \end{cases} \implies \begin{cases} x_1\,\mathsf{n}(a) &= -x_2\bar{c}a \\ x_2\,\mathsf{n}(c) &= -x_1\bar{a}c \end{cases}$$

Substituting this into the previous equality yields

$$\begin{aligned} x &= (ux_2(-\bar{c}a) + wx_1\,\mathsf{n}(c),\ ux_2\,\mathsf{n}(a) + wx_1(-\bar{a}c)) \\ &= \underbrace{ux_2(-\bar{c}a, \mathsf{n}(a))}_{\in K_1} + \underbrace{wx_1(\mathsf{n}(c), -\bar{a}c)}_{\in K_2} \end{aligned}$$

which proves the other inclusion and therefore $(\mathcal{O}_0 v)^\perp = K_1 + K_2$. □

The lemma expresses $(\mathcal{O}_0 v)^\perp$ as a submodule of M. Δ-KLPT instead requires a left $\mathcal{O}_0$-ideal, which is obtained for instance by projecting onto the first coordinate, which gives $J = \mathcal{O}_0\,\mathsf{n}(c) + \mathcal{O}_0\bar{c}a$ as in [CDK+25, Lemma 3.7]. Choosing a non-zero vector $x \in (\mathcal{O}_0 v)^\perp$ induces a choice of an ideal J in the class. Whether or not it is possible to choose this vector in a way that makes the pre-processing of Δ-KLPT$^+$ superfluous is unclear and left for future work.