ISOGENIES ON KUMMER SURFACES

MARIA CORTE-REAL SANTOS AND E. VICTOR FLYNN

Abstract. We first give a cleaner and more direct approach to the derivation of the Fast model of the Kummer surface. We show how to construct efficient (N, N)-isogenies, for any odd N, both on the general Kummer surface and on the Fast model.

1. Introduction

Various models have been constructed of the Kummer surface related to curves of genus 2, with more recent work emphasising models which are amenable to more efficient computations. There has also been considerable recent interest in computing isogenies on these surfaces. The intention of this article is to first give a more explicit derivation of the *Fast* Kummer surface model (a model introduced by Gaudry [Gau07] in the context of cryptography, and explored algorithmically by Chudnovsky and Chudnovsky [CC86] which allows for particularly fast computations), and then to construct (N, N)-isogenies on this model, for any odd N.

In Section 2, we recall the General Kummer model of the Kummer surface in $\mathbf{P}^3$, given on p.18 of [CF96], and summarise a number of associated ideas, such as the linear map induced by adding a point of order 2, certain biquadratic forms which relate to the group law on the Jacobian variety, and the Richelot isogeny.

In Section 3, we summarise the *Squared* Kummer model described by Cosset [Cos11, Ch. 4], and the linear map which relates it to the General Kummer of the Rosenhain curve given by Chung, Costello and Smith [CCS16, §3]. In Section 4, we summarise the Fast Kummer surface model, described by Gaudy [Gau07], which also allows fast computations and is $(2, 2)$-isogenous to the model in Section 3. The derivation by Gaudry [Gau07] uses several identities of theta functions as justification, an extension of the ground field was used, and there was no explicit Jacobian of which this was the Kummer Surface, but instead a $(2, 2)$-isogenous Jacobian was given over the field extension. The first main contribution of this article is that, in Section 4, we give an entirely algebraic rederivation of the Fast Kummer surface $\mathcal{K}^{\text{fast}}$ by finding a curve $\mathcal{D}$ together with a linear map from the General Kummer surface of $\mathcal{D}$ in Equation (4.2) to the Fast Kummer in Equation (4.1). Our aim is that this will make both the derivation and application of this elegant form more accessible to a wider audience who may use it over number fields for such things as height constants and descents. Everything is performed over the ground field of the parameters, and we now have an explicit Jacobian (namely the Jacobian variety of $\mathcal{D}$) of which $\mathcal{K}^{\text{fast}}$ is the Kummer surface.

1991 *Mathematics Subject Classification.* 11G30, 11G10, 14H40.
Key words and phrases. Jacobian, Abelian Variety.
14 September, 2024.

In Section 5, we describe a general method to construct (N, N)-isogenies between Kummer surfaces with efficiently computable biquadratic forms, for any odd N, using purely algebraic methods. More specifically, we show how to obtain the (N, N)-isogeny as a composition $\varphi = \lambda \circ \psi$, where ψ has the desired kernel and λ is a linear map which moves the image into the correct Kummer form. We give algorithms to compute ψ for any Kummer surface model with efficiently computable biquadratic forms. In the case of General and Fast Kummer surfaces, we explicitly show how to obtain this final linear map λ in Sections 5.1 and 5.2, respectively. This culminates in two further main contributions: an algorithm GetIsogeny (see Algorithm 7 in Section 5) which recovers the explict formulæ describing the (N, N)-isogeny φ from the N-torsion points generating the kernel; and an algorithm GetImage (see Algorithm 6 in Section 5) which computes the image of the isogeny φ. In particular, we considerably refine the scaling step of the algorithm by exploiting maps and symmetries on $\mathcal{K}^{\text{fast}}$ which allow us to avoid expensive square root operations in the ground field. These explicit algebraic derivations and descriptions will be of potential use not only in the Cryptographic community, but also to those researchers in Arithmetic Geometry who which to perform descent via isogeny over number fields. Throughout, we provide concrete complexities for all our algorithms, highlighting the main bottleneck of our work: finding degree-N homogeneous forms that are invariant under translation by an N-torsion point on the Kummer surface.

Since the Fast Kummer surfaces yield the most efficient (N, N)-isogenies, in Section 6 we provide concrete timings for running our methods on this model for odd primes $N \leq 19$.

The software accompanying this paper is written in `MAGMA` [BCP97] and Maple [Map24], and is publicly available under the MIT license. It is available at

`https://github.com/mariascrs/NN_isogenies`.

1.1. **Comparison to other methods.** We briefly compare our methods for computing (N, N)-isogenies between different models of Kummer isogenies for odd N to those in previous literature. The case $N = 3$ and 5 for the General Kummer surface was determined by Bruin, Flynn and Testa [BFT14] and by [Fly15], respectively. We recover these isogenies in Section 5. Turning to a different model, Corte-Real Santos, Costello and Smith [CCS24] give optimised formulæ for $(3, 3)$-isogenies between Fast Kummer surfaces, and touch on how these methods could be generalised to any odd N. We then extend these ideas to give a method for computing (N, N)-isogenies between Fast Kummer surfaces for any odd $N \geq 5$.

Bisson, Cosset, Lubicz, and Robert launched an ambitious program [Bis11, Cos11, LR15, LR16, Rob10, Rob21, LR22] based on the theory of theta functions [Mum84] to provide asymptotically efficient algorithms for arbitrary odd N (and beyond dimension 2 to arbitrarily high dimension). `AVIsogenies` [BCR10] is a software package written in `MAGMA` based on their results, and is publicly available.

We take a different perspective on the problem and analyse to what extent the method depicted by Corte-Real Santos, Costello and Smith [CCS24] can be optimised. Our approach allows us to describe algorithms that output the isogeny formulæ as well as the image Kummer surface, relying solely on simple linear algebra. To the best of our knowledge, the line of work relying on theta functions does not recover the formulæ describing the (N, N)-isogeny, which is interesting in its own right. In this sense, the purposes of `AVIsogenies` and our algorithm

are somewhat different. Indeed, `AVIsogenies` inputs an initial Kummer surface, isogeny kernel and a point, and outputs the image point and (theta null point of) the image Kummer surface; our algorithm inputs an initial Kummer surface and isogeny kernel, and outputs the explicit defining equations of the image Kummer surface and of the isogeny (which can then incidentally be used, if desired, to find images of specific points).

Though the algorithms developed using theta functions boast better asymptotic complexity, preliminary experimental evidence suggest that our software outperforms `AVIsogenies` for small odd N (see Section 6 for further details). However, for a precise and fair comparison between implementations, exact operation counts are needed.

Furthermore, our methods produce K-rational (N, N)-isogeny formulæ for *any* Kummer surface model given K-rational kernel generators (provided it has efficiently computable biquadratic forms). In this way, we do not require full K-rational 2-torsion in order to have a K-rational theta structure.

1.2. **Acknowledgements.** We thank Craig Costello and Sam Frengley for many fruitful discussions throughout the preparation of this paper. We further thank Benjamin Smith for discussions that lead to the statement of Proposition 5.3 and Conjecture 5.4. We are greatful to Kamal Khuri-Makdisi for kindly explaining a proof to Proposition 5.3, and allowing us to include it in this article. The first author was supported by UK EPSRC grant EP/S022503.

2. Generalities on Kummer surfaces

Let K be any field (not of characteristic 2) and consider a general curve of genus 2

$$y^2 = \mathfrak{F}(x) = f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0, \tag{2.1}$$

defined over K. We represent elements of the Jacobian variety by $\{(x_1, y_1), (x_2, y_2)\}$, as a shorthand for the divisor class of $(x_1, y_1) + (x_2, y_2) - \infty^+ - \infty^-$, where ∞^+ and ∞^- denote the points on the non-singular curve that lie over the singular point at infinity. The Kummer surface has an embedding (see p.18 of [CF96]) in $\mathbf{P}^3$ given by (k_1, k_2, k_3, k_4), where

$$k_1 = 1,\ k_2 = x_1 + x_2,\ k_3 = x_1x_2,\ k_4 = (F_0(x_1, x_2) - 2y_1y_2)/(x_1 - x_2)^2, \tag{2.2}$$

and where

$$\begin{aligned} F_0(x_1, x_2) =& 2f_0 + f_1(x_1 + x_2) + 2f_2(x_1x_2) + f_3(x_1x_2)(x_1 + x_2) \\ &+ 2f_4(x_1x_2)^2 + f_5(x_1x_2)^2(x_1 + x_2) + 2f_6(x_1x_2)^3. \end{aligned} \tag{2.3}$$

The defining equation of the Kummer surface is given by

$$\mathcal{K}^{\text{gen}} : F_1(k_1, k_2, k_3)k_4^2 + F_2(k_1, k_2, k_3)k_4 + F_3(k_1, k_2, k_3) = 0, \tag{2.4}$$

where F_1, F_2, F_3 are given by:

$$\begin{aligned}
F_1(k_1,k_2,k_3) &= k_2^2-4k_1k_3,\\
F_2(k_1,k_2,k_3) &= -2\left(2k_1^3f_0+k_1^2k_2f_1+2k_1^2k_3f_2+k_1k_2k_3f_3+2k_1k_3^2f_4\right.\\
&\quad \left.+k_2k_3^2f_5+2k_3^3f_6\right),\\
F_3(k_1,k_2,k_3) &= -4k_1^4f_0f_2+k_1^4f_1^2-4k_1^3k_2f_0f_3-2k_1^3k_3f_1f_3-4k_1^2k_2^2f_0f_4\\
&\quad +4k_1^2k_2k_3f_0f_5-4k_1^2k_2k_3f_1f_4-4k_1^2k_3^2f_0f_6+2k_1^2k_3^2f_1f_5\\
&\quad -4k_1^2k_3^2f_2f_4+k_1^2k_3^2f_3^2-4k_1k_2^3f_0f_5+8k_1k_2^2k_3f_0f_6-4k_2^4f_0f_6\\
&\quad -4k_1k_2^2k_3f_1f_5+4k_1k_2k_3^2f_1f_6-4k_1k_2k_3^2f_2f_5-2k_1k_3^3f_3f_5\\
&\quad -4k_2^3k_3f_1f_6-4k_2^2k_3^2f_2f_6-4k_2k_3^3f_3f_6-4k_3^4f_4f_6+k_3^4f_5^2.
\end{aligned}$$

We shall refer to this model as $\mathcal{K}^{\mathrm{gen}}$, the General Kummer model. We also recall from p.65 of [CF96] that there are local parameters s_1, s_2 on the Jacobian variety, and all coordinates of the Kummer surface can be written as formal power series in these paramaters. We refer the reader to [CF96] for the definition of s_1, s_2, which we shall not require here. We merely note, for future reference, that the coordinates of the General Kummer model are as follows up to the degree 6 terms, when written as formal power series in the local parameters.

(2.5)

$$\begin{aligned}
k_1 =& s_2^2-f_2s_2^4-f_6s_1^4+4f_0f_4s_2^6+8f_0f_5s_1s_2^5+18f_0f_6s_1^2s_2^4-f_1f_3s_2^6\\
&+f_1f_5s_1^2s_2^4+4f_1f_6s_1^3s_2^3+2f_2^2s_2^6+2f_2f_6s_1^4s_2^2+2f_4f_6s_1^6+\mathrm{O}(8),\\
k_2 =& 2s_1s_2+f_1s_2^4+f_3s_1^2s_2^2+f_5s_1^4+f_0f_3s_2^6+8f_0f_4s_1s_2^5+16f_0f_5s_1^2s_2^4\\
&+40f_0f_6s_1^3s_2^3-2f_1f_2s_2^6+2f_1f_4s_1^2s_2^4+6f_1f_5s_1^3s_2^3+16f_1f_6s_1^4s_2^2-f_2f_3s_1^2s_2^4\\
&+2f_2f_5s_1^4s_2^2+8f_2f_6s_1^5s_2-f_3f_4s_1^4s_2^2+f_3f_6s_1^6-2f_4f_5s_1^6+\mathrm{O}(8),\\
k_3 =& s_1^2-f_0s_2^4-f_4s_1^4+2f_0f_2s_2^6+2f_0f_4s_1^2s_2^4+4f_0f_5s_1^3s_2^3+18f_0f_6s_1^4s_2^2\\
&+f_1f_5s_1^4s_2^2+8f_1f_6s_1^5s_2+4f_2f_6s_1^6-f_3f_5s_1^6+2f_4^2s_1^6+\mathrm{O}(8),\\
k_4 =& 1,
\end{aligned}$$

where by O(8) we mean terms of degree at least 8 in s_1, s_2.

We note that if we perform a quadratic twist on the curve given in Equation (2.1) to give $cy^2=\mathfrak{F}(x)$, then this induces a linear map $(k_1,k_2,k_3,k_4)\mapsto(k_1,k_2,k_3,ck_4)$ between the Kummer surfaces.

We know (see p.22 of [CF96]) that addition by any point of order 2 gives a linear map on the Kummer surface. Specifically, if our curve has the form $y^2=(g_2x^2+g_1x+g_0)(h_4x^4+h_3x^3+h_2x^2+h_1x+h_0)$, and r_1, r_2 are the roots of $g_2x^2+g_1x+g_0$, then addition by $\{(r_1,0),(r_2,0)\}$ induces on $\mathcal{K}^{\mathrm{gen}}$ the linear map

$$\begin{pmatrix}k_1\\k_2\\k_3\\k_4\end{pmatrix}\mapsto W\begin{pmatrix}k_1\\k_2\\k_3\\k_4\end{pmatrix} \tag{2.6}$$

where W is the matrix (reproducing (3.2.10) from [CF96]):

(2.7)
$$\begin{pmatrix} g_2^2h_0 + g_0g_2h_2 - g_0^2h_4 & g_0g_2h_3 - g_0g_1h_4 & g_1g_2h_3 - g_1^2h_4 + 2g_0g_2h_4 & g_2 \\ -g_0g_2h_1 - g_0g_1h_2 + g_0^2h_3 & g_2^2h_0 - g_0g_2h_2 + g_0^2h_4 & g_2^2h_1 - g_1g_2h_2 - g_0g_2h_3 & -g_1 \\ -g_1^2h_0 + 2g_0g_2h_0 + g_0g_1h_1 & -g_1g_2h_0 + g_0g_2h_1 & -g_2^2h_0 + g_0g_2h_2 + g_0^2h_4 & g_0 \\ w_{41} & w_{42} & w_{43} & w_{44} \end{pmatrix}$$

and where the entries $w_{41}, w_{42}, w_{43}, w_{44}$ of the bottom row are:

(2.8)
$$\begin{aligned} w_{41} &:= -g_1g_2^2h_0h_1 + g_1^2g_2h_0h_2 + g_0g_2^2h_1^2 - 4g_0g_2^2h_0h_2 - g_0g_1g_2h_1h2 + g_0g_1g_2h_0h_3 - g_0^2g_2h_1h_3, \\ w_{42} &:= g_1^2g_2h_0h_3 - g_1^3h_0h_4 - 2g_0g_2^2h_0h_3 - g_0g_1g_2h_1h_3 + 4g_0g_1g_2h_0h_4 + g_0g_1^2h_1h_4 - 2g_0^2g_2h_1h_4, \\ w_{43} &:= -g_0g_2^2h_1h_3 - g_0g_1g_2h_2h_3 + g_0g_1g_2h_1h_4 + g_0g_1^2h_2h_4 + g_0^2g_2h_3^2 - 4g_0^2g_2h_2h_4 - g_0^2g_1h_3h_4, \\ w_{44} &:= -g_2^2h_0 - g_0g_2h_2 - g_0^2h_4. \end{aligned}$$

The eigenvalues of this matrix are the square roots of the resultant of $g_2x^2 + g_1x + g_0$ and $h_4x^4 + h_3x^3 + h_2x^2 + h_1x + h_0$, so this map will be diagonalisable over K exactly when this resultant is square in K. Furthermore, if $\{(r_1, 0), (r_2, 0)\}$ and $\{(r_3, 0), (r_4, 0)\}$ are points of order 2, where r_1, r_2, r_3, r_4 are distinct, then the corresponding matrices commute as affine matrices. If they have a point in common then they anticommute.

As described in Theorem 3.4.1 of [CF96], there is a 4×4 matrix of biquadratic forms B_{ij} such that, for any points $\mathfrak{A}, \mathfrak{B}$ on the Jacobian variety,

$$\Big(B_{ij}(k(\mathfrak{A}), k(\mathfrak{B}))\Big) = \Big(k_i(\mathfrak{A} + \mathfrak{B})k_j(\mathfrak{A} - \mathfrak{B}) + k_i(\mathfrak{A} - \mathfrak{B})k_j(\mathfrak{A} + \mathfrak{B})\Big), \tag{2.9}$$

where $k(\mathfrak{A})$ denotes $(k_1(\mathfrak{A}), k_2(\mathfrak{A}), k_3(\mathfrak{A}), k_4(\mathfrak{A}))$, the image of $\mathfrak{A}$ in $\mathcal{K}^{\mathrm{gen}}$, and similarly for $k(\mathfrak{B})$.

If the genus 2 curve has the form

$$y^2 = H_1(x)H_2(x)H_3(x), \tag{2.10}$$

and we define the h_{ij} by $H_j = H_j(x) = h_{j2}x^2 + h_{j1}x + h_{j0}$, then the Jacobian admits a Richelot isogeny (described on p.89 of [CF96]) to the Jacobian of the following curve:

$$y^2 = \Delta\big(H_2'H_3 - H_2H_3'\big)\big(H_3'H_1 - H_3H_1'\big)\big(H_1'H_2 - H_1H_2'\big), \tag{2.11}$$

where $\Delta = \det(h_{ij})$. This induces a $(2, 2)$-isogeny on the associated Kummer surfaces, whose kernel consists of the identity and the elements of order 2 corresponding to H_1, H_2, H_3. There is also a Richelot isogeny in the other direction (the dual isogeny), and the compisition of these gives multiplication by 2.

We note that all of the above identities are derivable purely algebraically; for example, the defining equation of the above Kummer surface can be verified directly, merely from the fact that $y_1^2 = \mathfrak{F}(x_1)$ and $y_2^2 = \mathfrak{F}(x_2)$, where we recall here that $\{(x_1, y_1), (x_2, y_2)\}$ is an element of the Jacobian variety.

3. The Squared Kummer model

In this section, we shall briefly recall a model of the Kummer surface from [Cos11, Ch. 4], which we shall call $\mathcal{K}^{\mathrm{sqr}}$, the *Squared Kummer surface*, for reasons which will become apparent in Section 4.

We define the following quantities in terms of our parameters $a, b, c, d \in K$.

$$\begin{aligned} A &= a^2 + b^2 + c^2 + d^2,\ B = a^2 + b^2 - c^2 - d^2, \\ C &= a^2 - b^2 + c^2 - d^2,\ D = a^2 - b^2 - c^2 + d^2, \end{aligned} \tag{3.1}$$

and let $\gamma = \sqrt{CD/(AB)}$. We define the Rosenhain curve

$$\mathcal{C}_{\mathrm{rosen}} : y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu), \tag{3.2}$$

where

$$\lambda = \frac{a^2c^2}{b^2d^2},\ \mu = \frac{c^2(1+\gamma)}{d^2(1-\gamma)},\ \nu = \frac{a^2(1+\gamma)}{b^2(1-\gamma)}. \tag{3.3}$$

We impose a non-degeneracy condition on a, b, c, d that the roots of $\mathcal{C}_{\mathrm{rosen}}$ are distinct, so that it is of genus 2. This curve is defined over $K(\gamma)$; even when $\gamma \notin K$, it is birationally equivalent to its $K(\gamma)/K$-conjugate.

Remark 3.1. In previous literature, for example in [CCS16], the parameters for the squared Kummer surface are often constructed from a, b, c, d rather than a^2, b^2, c^2, d^2, as above. However, since we shall be primarily interested in the fast Kummer surface model, to be described in the next section, our main aim here is just to link the two models, for which it will be convenient to assume that the parameters are squares in our ground field K.

It is noted by Chung, Costello and Smith in [CCS16, §3] that we can apply the following linear map to the coordinates k_1, k_2, k_3, k_4 of $\mathcal{K}^{\mathrm{gen}}(\mathcal{C}_{\mathrm{rosen}})$, the general Kummer model for Rosenhain curve, to obtain the coordinates X, Y, Z, T of the squared Kummer surface, namely:

$$\begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} = M \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix}, \tag{3.4}$$

where

$$M = \begin{pmatrix} a^2\mu(\lambda+\nu) & -a^2\mu & a^2(\mu+1) & -a^2 \\ b^2\lambda\nu(1+\mu) & -b^2\lambda\nu & b^2(\lambda+\nu) & -b^2 \\ c^2\nu(\lambda+\mu) & -c^2\nu & c^2(\nu+1) & -c^2 \\ d^2\lambda\mu(1+\nu) & -d^2\lambda\mu & d^2(\lambda+\mu) & -d^2 \end{pmatrix}. \tag{3.5}$$

We note that there is an error in [CCS16, §3], as some of the factors were omitted from the entries of the above matrix, and we have corrected that error here.

After applying the above linear map, the Kummer equation is transformed to the following model.

$$\begin{aligned} \mathcal{K}^{\mathrm{sqr}} : \Big(&(X^2 + Y^2 + Z^2 + T^2) - F(XT + YZ) \\ &- G(XZ + YT) - H(XY + ZT) \Big)^2 = 4E^2XYZT, \end{aligned} \tag{3.6}$$

where

$$\begin{aligned} E &= abcdABCD/\big((a^2d^2-b^2c^2)(a^2c^2-b^2d^2)(a^2b^2-c^2d^2)\big),\\ F &= (a^4-b^4-c^4+d^4)/(a^2d^2-b^2c^2),\\ G &= (a^4-b^4+c^4-d^4)/(a^2c^2-b^2d^2),\\ H &= (a^4+b^4-c^4-d^4)/(a^2b^2-c^2d^2). \end{aligned} \tag{3.7}$$

We note that $\mathcal{K}^{\text{sqr}}$ is defined over K, even when $\mathcal{K}^{\text{gen}}$ is defined over $K(\gamma)$.

If we consider the identity element and the points of order 2 on the Jacobian of $\mathcal{C}_{\text{rosen}}$ given by $\{\infty,(0,0)\}$, $\{(\mu,0),(\nu,0)\}$ and $\{(1,0),(\lambda,0)\}$, then addition by these induces on $\mathcal{K}^{\text{sqr}}$ the linear maps which take (X,Y,Z,T) to, respectively

$$(X,Y,Z,T),\ (Y,X,T,Z),\ (Z,T,X,Y),\ (T,Z,Y,X). \tag{3.8}$$

The change in coordinates given in M has therefore not diagonalised these maps, but has certainly greatly simplified them.

4. Rederivation of the Fast Kummer Surface

In this section, we recall another elegant and efficient model of the Kummer surface, described in [Gau07]:

$$\begin{aligned} \mathcal{K}^{\text{fast}} : X^4+Y^4+Z^4+T^4 - F(X^2T^2+Y^2Z^2) - G(X^2Z^2+Y^2T^2)\\ - H(X^2Y^2+Z^2T^2) + 2EXYZT = 0, \end{aligned} \tag{4.1}$$

where A,B,C,D,E,F,G,H are as defined in Equation (3.1) and Equation (3.7). We shall refer to this as $\mathcal{K}^{\text{fast}}$, the Fast model of the Kummer surface, and it will be our main focus here.

We note that there is a map $(X,Y,Z,T) \mapsto (X^2,Y^2,Z^2,T^2)$ from the Fast Kummer surface in Equation (4.1) to the Squared Kummer surface model in Equation (3.6) in the previous section, hence justifying its name.

The derivation of this model in [Gau07] uses identities of theta functions. A connection is given with the Rosenhain curve $\mathcal{C}_{\text{rosen}}$ in Equation (3.2), namely that its Jacobian is $(2,2)$-isogenous to that related to the Fast Kummer, and a rationality assumption is given, namely that $\gamma = \sqrt{CD/(AB)}$ is in the ground field K. However, it is difficult from the literature to derive the Kummer surface equation for $\mathcal{K}^{\text{fast}}$ purely algebraically or to find an explicit map from a General Kummer; even if one were to follow through the map from the Kummer of the Rosenhain curve, this would be a $(2,2)$-isogeny, rather than a linear map. Our intention in this section is to show how to derive the Fast Kummer purely algebraically, with a linear map from the General Kummer of a curve which is defined over the ground field of a,b,c,d.

To find a plausible candidate, we note that $\mathcal{K}^{\text{fast}}$ and $\mathcal{K}^{\text{sqr}}$ are $(2,2)$-isogenous via the map $(X,Y,Z,T) \mapsto (X^2,Y^2,Z^2,T^2)$, and we saw in the last section that there is a linear map between $\mathcal{K}^{\text{sqr}}$ and the General Kummer of the Rosenhain curve. So, a natural candidate for our desired curve is one which is $(2,2)$-isogenous to the Rosenhain curve. If we apply the Richelot formula in Equation (2.11) to

the Rosenhain curve $\mathcal{C}_{\text{rosen}}$ in Equation (3.2) we obtain (up to quadratic twist) the following curve.

$$\mathcal{D} : y^2 = x(x-1)(x-\rho)(x-\sigma)(x-\tau), \tag{4.2}$$

where

$$\rho = CD/(AB),\ \sigma = (ac+bd)C/((ac-bd)A),\ \tau = (ac+bd)D/((ac-bd)B). \tag{4.3}$$

There is a Richelot isogeny from $\mathcal{K}^{\text{gen}}(\mathcal{C}_{\text{rosen}})$ to $\mathcal{K}^{\text{gen}}(\mathcal{D})$, and a Richelot isogeny (the dual isogeny) from $\mathcal{K}^{\text{gen}}(\mathcal{D})$ to $\mathcal{K}^{\text{gen}}(\mathcal{C}_{\text{rosen}})$. We shall not require here the explicit equations for these Richelot isogenies, since from now onwards we shall work entirely with $\mathcal{K}^{\text{gen}}(\mathcal{D})$.

We note that, unlike $\mathcal{C}_{\text{rosen}}$, the curve $\mathcal{D}$ is automatically defined over the ground field of a, b, c, d, without the need for any rationality assumption. We do, however, impose a non-degeneracy condition on a, b, c, d that $\infty, 0, 1, \rho, \tau, \sigma$ are distinct, so that $\mathcal{D}$ is a curve of genus 2.

We also note that the Fast Kummer surface has several diagonalised involutions, namely if one negates any two of X, Y, Z, T. This suggests that this model might be derivable from the General Kummer of $\mathcal{D}$ by finding a linear map which diagonalises these involutions. Let E_0 denote the identity element of the Jacobian of $\mathcal{D}$, and note that the following points of order 2 have no overlap in support: $E_1 = \{(\sigma, 0), (\tau, 0)\}$, $E_2 = \{\infty, (0, 0)\}$ and $E_3 = \{(1, 0), (\rho, 0)\}$.

Our aim is to find linear change in coordinates which simultaneously diagonalises addition by E_1, E_2, E_3. Since $E_1 = E_2 + E_3$, it is sufficient if we simultaneously diagonalise E_2 and E_3. After using Equation (2.7) to find the matrix which gives addition by E_2 on the General Kummer, we see that it has eigenvalues $(ac + bd)CD/((ac - bd)AB)$ and $-(ac + bd)CD/((ac - bd)AB)$, each with an eigenspace of dimension 2. Similarly, the matrix which gives addition by E_3 on the General Kummer has eigenvalues $4(a^2d^2 - b^2c^2)(a^2b^2 - c^2d^2)CD/(AB(ac - bd))^2$ and $-4(a^2d^2 - b^2c^2)(a^2b^2 - c^2d^2)CD/(AB(ac - bd))^2$, each with an eigenspace of dimension 2. These commute as affine matrices and so are simultaneously diagonalisable. We find that there is a dimension 1 intersection of each eigenspace of the first matrix and each eigenspace of the second matrix, which provides four linearly independent vectors which are common eigenvectors of both matrices. We put these as the columns of the following change of basis matrix P.

$$P = \begin{pmatrix} cn_1^2A^2B^2 & -dn_1^2A^2B^2 & -an_1^2A^2B^2 & bn_1^2A^2B^2 \\ 2an_1ABm_1 & 2bn_1ABm_2 & -2cn_1ABm_3 & 2dn_1ABm_4 \\ cn_1n_2ABCD & dn_1n_2ABCD & -an_1n_2ABCD & -bn_1n_2ABCD \\ 2an_2CDm_1 & -2bn_2CDm_2 & -2cn_2CDm_3 & -2dn_2CDm_4 \end{pmatrix}, \tag{4.4}$$

where

$$\begin{aligned} &m_1 = c^2(a^4 + b^4 - c^4 + d^4) - 2a^2b^2d^2,\ m_2 = d^2(a^4 + b^4 + c^4 - d^4) - 2a^2b^2c^2, \\ &m_3 = a^2(a^4 - b^4 - c^4 - d^4) + 2b^2c^2d^2,\ m_4 = b^2(a^4 - b^4 + c^4 + d^4) - 2a^2c^2d^2, \\ &n_1 = ac - bd,\ n_2 = ac + bd. \end{aligned} \tag{4.5}$$

We define the following map:

$$\begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} = P^{-1} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix}. \tag{4.6}$$

Direct substitution (which we have verified in the Maple file in [CF24a]) now gives that General Kummer equation is converted to the Fast Kummer equation, as desired, which proves the following result.

Theorem 4.1. *The map in Equation (4.6) is a linear map from the General Kummer of $\mathcal{D}$ from Equation (4.2) to the Fast Kummer from Equation (4.1).*

We now have a direct linear map between the General Kummer and the Fast Kummer, which was our aim in this section, so that we now have a direct algebraic derivation of the Fast Kummer. It also gives a simpler approach any time computations on the associated Jacobian variety are required. Indeed, one can use the Jacobian variety of $\mathcal{D}$ if operations are required which are not defined on the Kummer surface, and can help with pseudo-addition on the Kummer surface. This approach also removes the need for any arithmetic assumption about γ, since all of our varieties and maps are defined over the ground field of a, b, c, d.

The relationship between the objects discussed so far is summarised in the following commutative diagram.

$$\begin{array}{ccc} \mathcal{K}^{\text{gen}}(\mathcal{D}) & \xrightarrow{\text{Richelot}} & \mathcal{K}^{\text{gen}}(\mathcal{C}_{\text{rosen}}) \\ \downarrow P^{-1} & & \downarrow M \\ \mathcal{K}^{\text{fast}} & \xrightarrow{\mathcal{S}} & \mathcal{K}^{\text{sqr}}, \end{array} \tag{4.7}$$

where: surfaces $\mathcal{K}^{\text{gen}}$, $\mathcal{K}^{\text{sqr}}$ and $\mathcal{K}^{\text{fast}}$ are as in Equations (2.4), (3.6) and (4.1), respectively; curves $\mathcal{C}_{\text{rosen}}$, $\mathcal{D}$ are as in Equations (3.2) and (4.2), respectively; linear maps M, P are as in Equations (3.5) and (4.4), respectively; and $\mathcal{S}$ denotes the squaring map $\mathcal{S}\colon (X, Y, Z, T) \mapsto (X^2, Y^2, Z^2, T^2)$.

Additions by E_1, E_2, E_3 on the Fast Kummer surface have all been diagonalised. Explicitly, addition by E_1 induces the map $(X, Y, Z, T) \mapsto (X, Y, -Z, -T)$, addition by E_2 induces the map $(X, Y, Z, T) \mapsto (X, -Y, Z, -T)$ and addition by E_3 induces the map $(X, Y, Z, T) \mapsto (X, -Y, -Z, T)$. If we let $E_0, \ldots, E_{15}$ denote the entire 2-torsion subgroup, the addition by these induce the map which takes (X, Y, Z, T) to, respectively:

$$\begin{aligned} &(X, Y, Z, T),\ (X, Y, -Z, -T),\ (X, -Y, Z, -T),\ (X, -Y, -Z, T), \\ &(Y, X, T, Z),\ (Y, X, -T, -Z),\ (Y, -X, T, -Z),\ (Y, -X, -T, Z), \\ &(Z, T, X, Y),\ (Z, T, -X, -Y),\ (Z, -T, X, -Y),\ (Z, -T, -X, Y), \\ &(T, Z, Y, X),\ (T, Z, -Y, -X),\ (T, -Z, Y, -X),\ (T, -Z, -Y, X). \end{aligned} \tag{4.8}$$

The identity element on $\mathcal{K}^{\text{fast}}$ is (a, b, c, d). If we apply the above maps to this, we get the complete set of 2-torsion elements on $\mathcal{K}^{\text{fast}}$, namely:

$$\begin{aligned} &(a, b, c, d),\ (a, b, -c, -d),\ (a, -b, c, -d),\ (a, -b, -c, d), \\ &(b, a, d, c),\ (b, a, -d, -c),\ (b, -a, d, -c),\ (b, -a, -d, c), \\ &(c, d, a, b),\ (c, d, -a, -b),\ (c, -d, a, -b),\ (c, -d, -a, b), \\ &(d, c, b, a),\ (d, c, -b, -a),\ (d, -c, b, -a),\ (d, -c, -b, a), \end{aligned} \tag{4.9}$$

which are the same as the list of nodes given by Gaudry in [Gau07, §3.4]. We denote the action by the two torsion point E_i by $\sigma_i : \mathcal{K}^{\text{fast}} \to \mathcal{K}^{\text{fast}}$ as described by Equation (4.8). For example, $\sigma_1 : (X, Y, Z, T) \mapsto (X, Y, -Z, -T)$, and $\sigma_4 : (X, Y, Z, T) \mapsto (Y, X, T, Z)$.

After applying the linear change of basis P, we recover the biquadratic forms given by [RS17], which are much simpler version than the biquadratic forms on the General Kummer surface, as follows.

Corollary 4.2. *Let P, Q be points on the Jacobian of $\mathcal{D}$, and let (X_P, Y_P, Z_P, T_P), (X_Q, Y_Q, Z_Q, T_Q) be their images on the Fast Kummer surface. Define the following 4×4 matrix of biquadratic forms.*

(4.10)

$$\begin{aligned}
B_{11} =&(X_P^2+Y_P^2+Z_P^2+T_P^2)(X_Q^2+Y_Q^2+Z_Q^2+T_Q^2)/(4A)\\
&+(X_P^2+Y_P^2-Z_P^2-T_P^2)(X_Q^2+Y_Q^2-Z_Q^2-T_Q^2)/(4B)\\
&+(X_P^2-Y_P^2+Z_P^2-T_P^2)(X_Q^2-Y_Q^2+Z_Q^2-T_Q^2)/(4C)\\
&+(X_P^2-Y_P^2-Z_P^2+T_P^2)(X_Q^2-Y_Q^2-Z_Q^2+T_Q^2)/(4D),\\
B_{22} =&(X_P^2+Y_P^2+Z_P^2+T_P^2)(X_Q^2+Y_Q^2+Z_Q^2+T_Q^2)/(4A)\\
&+(X_P^2+Y_P^2-Z_P^2-T_P^2)(X_Q^2+Y_Q^2-Z_Q^2-T_Q^2)/(4B)\\
&-(X_P^2-Y_P^2+Z_P^2-T_P^2)(X_Q^2-Y_Q^2+Z_Q^2-T_Q^2)/(4C)\\
&-(X_P^2-Y_P^2-Z_P^2+T_P^2)(X_Q^2-Y_Q^2-Z_Q^2+T_Q^2)/(4D),\\
B_{33} =&(X_P^2+Y_P^2+Z_P^2+T_P^2)(X_Q^2+Y_Q^2+Z_Q^2+T_Q^2)/(4A)\\
&-(X_P^2+Y_P^2-Z_P^2-T_P^2)(X_Q^2+Y_Q^2-Z_Q^2-T_Q^2)/(4B)\\
&+(X_P^2-Y_P^2+Z_P^2-T_P^2)(X_Q^2-Y_Q^2+Z_Q^2-T_Q^2)/(4C)\\
&-(X_P^2-Y_P^2-Z_P^2+T_P^2)(X_Q^2-Y_Q^2-Z_Q^2+T_Q^2)/(4D),\\
B_{44} =&(X_P^2+Y_P^2+Z_P^2+T_P^2)(X_Q^2+Y_Q^2+Z_Q^2+T_Q^2)/(4A)\\
&-(X_P^2+Y_P^2-Z_P^2-T_P^2)(X_Q^2+Y_Q^2-Z_Q^2-T_Q^2)/(4B)\\
&-(X_P^2-Y_P^2+Z_P^2-T_P^2)(X_Q^2-Y_Q^2+Z_Q^2-T_Q^2)/(4C)\\
&+(X_P^2-Y_P^2-Z_P^2+T_P^2)(X_Q^2-Y_Q^2-Z_Q^2+T_Q^2)/(4D),\\
B_{12} =&4(ab(X_PY_PX_QY_Q+Z_PT_PZ_QT_Q)-cd(X_PY_PZ_QT_Q+Z_PT_PX_QY_Q))/g_{12},\\
B_{13} =&4(ac(X_PZ_PX_QZ_Q+Y_PT_PY_QT_Q)-bd(X_PZ_PY_QT_Q+Y_PT_PX_QZ_Q))/g_{13},\\
B_{14} =&4(ad(X_PT_PX_QT_Q+Y_PZ_PY_QZ_Q)-bc(X_PT_PY_QZ_Q+Y_PZ_PX_QT_Q))/g_{14},\\
B_{23} =&4(bc(X_PT_PX_QT_Q+Y_PZ_PY_QZ_Q)-ad(X_PT_PY_QZ_Q+Y_PZ_PX_QT_Q))/g_{23},\\
B_{24} =&4(bd(X_PZ_PX_QZ_Q+Y_PT_PY_QT_Q)-ac(X_PZ_PY_QT_Q+Y_PT_PX_QZ_Q))/g_{24},\\
B_{34} =&4(cd(X_PY_PX_QY_Q+Z_PT_PZ_QT_Q)-ab(Z_PT_PX_QY_Q+X_PY_PZ_QT_Q))/g_{34},
\end{aligned}$$

where g_{ij} denotes $g_ig_j - g_kg_\ell$, where $\{i, j, k, \ell\} = \{1, 2, 3, 4\}$ and g_1, g_2, g_3, g_4 denote A, B, C, D, respectively (for example g_{12} denotes $AB - CD$). For $i > j$, we define $B_{ij} = B_{ji}$. Then these satisfy the analogous identity on the Fast Kummer as described in Equation (2.9) for the General Kummer. That is to say, if we let $(\xi_1, \xi_2, \xi_3, \xi_4)$ and $(\zeta_1, \zeta_2, \zeta_3, \zeta_4)$ denote $(X_{P+Q}, Y_{P+Q}, Z_{P+Q}, T_{P+Q})$ and

$(X_{P-Q}, Y_{P-Q}, Z_{P-Q}, T_{P-Q})$ respectively, then the above matrix $\big(B_{ij}\big)$ is projectively equal to the matrix $\big(\xi_i\zeta_j + \zeta_i\xi_j\big)$.

From the biquadratic forms, we may inductively define the division polynomials which give multiplication-by-N, as follows. We initially define

$$\begin{aligned} &\phi_X^{(0)} = a, \phi_Y^{(0)} = b, \phi_Z^{(0)} = c, \phi_T^{(0)} = d, \\ &\phi_X^{(1)} = X, \phi_Y^{(1)} = Y, \phi_Z^{(1)} = Z, \phi_T^{(1)} = T, \end{aligned} \tag{4.11}$$

and then inductively define the following, which are polynomials modulo the equation of the Fast Kummer surface.

$$\begin{aligned} \phi_X^{(2N)} &= B_{11}\big((\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/a, \\ \phi_Y^{(2N)} &= B_{22}\big((\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/b, \\ \phi_Z^{(2N)} &= B_{33}\big((\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/c, \\ \phi_T^{(2N)} &= B_{44}\big((\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/d, \end{aligned} \tag{4.12}$$

and similarly

$$\begin{aligned} \phi_X^{(2N+1)} &= B_{11}\big((\phi_X^{(N+1)}, \phi_Y^{(N+1)}, \phi_Z^{(N+1)}, \phi_T^{(N+1)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/X, \\ \phi_Y^{(2N+1)} &= B_{22}\big((\phi_X^{(N+1)}, \phi_Y^{(N+1)}, \phi_Z^{(N+1)}, \phi_T^{(N+1)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/Y, \\ \phi_Z^{(2N+1)} &= B_{33}\big((\phi_X^{(N+1)}, \phi_Y^{(N+1)}, \phi_Z^{(N+1)}, \phi_T^{(N+1)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/Z, \\ \phi_T^{(2N+1)} &= B_{44}\big((\phi_X^{(N+1)}, \phi_Y^{(N+1)}, \phi_Z^{(N+1)}, \phi_T^{(N+1)}), (\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)})\big)/T. \end{aligned} \tag{4.13}$$

The polynomials $\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)}$ are each of degree N^2 in X, Y, Z, T, and they give the multiplication-by-N map.

For odd N, we note that the 2-torsion subgroup maps to itself under multiplication by N, and the effect of addition by these 2-torsion elements, described in Equation (4.8) is preserved. Furthermore, we can see inductively that these actions are preserved on the division polynomials not merely projectively, but transparently as affine polynomials. That is to say, for example, replacing X, Y, Z, T with $X, Y, -Z, -T$ has precisely the effect of replacing $\phi_X^{(N)}, \phi_Y^{(N)}, \phi_Z^{(N)}, \phi_T^{(N)}$ with $\phi_X^{(N)}, \phi_Y^{(N)}, -\phi_Z^{(N)}, -\phi_T^{(N)}$, and similarly for all of the involutions in Equation (4.8). One consequence is that the polynomial $\phi_X^{(N)}$ only includes those monomials which are left unchanged by taking (X, Y, Z, T) to any of (X, Y, Z, T), $(X, Y, -Z, -T)$, $(X, -Y, Z, -T)$, $(X, -Y, -Z, T)$. The polynomial $\phi_Y^{(N)}$ only includes those monomials which are left unchanged by first and second of these and are negated by the third and fourth. The polynomial $\phi_Z^{(N)}$ only includes those monomials which are left unchanged by first and third of these and are negated by the second and fourth. The polynomial $\phi_T^{(N)}$ only includes those monomials which are left unchanged by first and fourth of these and are negated by the second and third. This describes a partition of all monomials of degree N^2 into four parts, and each of the degree N division polynomials only contains those from one part. We note also that the equation of the Fast Kummer surface itself involves only degree 4 monomials of the first type (left unchanged by all four of these involutions).

Addition by other points of order 2 gives further structure. For example, swapping $X \leftrightarrow Y, Z \leftrightarrow T$ will have the effect $\phi_X^{(N)} \leftrightarrow \phi_Y^{(N)}$, $\phi_Z^{(N)} \leftrightarrow \phi_T^{(N)}$.

We also note that the construction in this section, in which two elements of order 2 have their associated linear maps simultaneously diagonlised in order to simplify the Kummer surface (and its associated structures) is available whenever we start with a curve of genus 2 of the form $y^2 = H_1(x)H_2(x)H_3(x)$, provided that the resultant of H_1 and H_2H_3, and the resultant of H_2 and H_1H_3 are squares in K (it follows that the resultant of H_3 and H_1H_2 will also be square in K). This ensures that we can simultaneously diagonalise that maps for addition by the points of order 2 corresponding to H_1, H_2, H_3 since they will commute as affine matrices and their eigenvalues will be in K. The result will not necessarily be as elegant as the Fast Kummer surface model (for example, it will not typically be monic in all of X^4, Y^4, Z^4, T^4), but it will be just as sparse, consisting only of terms involving the monimials X^4, Y^4, Z^4, T^4, X^2Y^2, Z^2T^2, X^2Z^2, Y^2T^2, Z^2T^2, Y^2Z^2, and the associated biquadratic forms will also involve the same monomials as for the Fast Kummer model. Furthermore, the resulting model would be applicable to a wider set of Kummer surfaces. We demonstrate this in the following example.

Example 4.3. Consider the genus-2 hyperelliptic curve $C\colon y^2 = H_1(x)H_2(x)H_3(x)$ defined over $\mathbb{F}_{101}$ with

$$\begin{aligned} H_1(x) &= x^2 + 15x + 13, \\ H_2(x) &= x^2 + 53x + 83, \\ H_3(x) &= x^2 + 10x + 64. \end{aligned}$$

and corresponding General Kummer surface

$$\begin{aligned} \mathcal{K}^{\mathrm{gen}}\colon\ & 84k_1^4 + k_1^3k_2 + 3k_1^3k_3 + 11k_1^3k_4 + 7k_1^2k_2^2 + 93k_1^2k_2k_3 + 33k_1^2k_2k_4 + 64k_1^2k_3^2 \\ & + 96k_1^2k_3k_4 + 50k_1k_2^3 + 76k_1k_2^2k_3 + 49k_1k_2k_3^2 + 9k_1k_2k_3k_4 96k_1k_3^3 \\ & + 25k_1k_3^2k_4 + 97k_1k_3k_4^2 + 11k_2^4 + 66k_2^3k_3 + 96k_2^2k_3^2 + k_2^2k_4^2 + 18k_2k_3^3 \\ & + 46k_2k_3^2k_4 + 49k_3^4 + 97k_3^3k_4. \end{aligned}$$

None of H_1, H_2, H_3 have roots in $\mathbb{F}_{101}$, and therefore C cannot be put into Rosenhain form. As a result, $\mathcal{K}^{\mathrm{gen}}$ cannot be put into Fast Kummer form. However, the resultants $\mathrm{res}(H_1, H_2H_3)$ and $\mathrm{res}(H_2, H_1H_3)$ are squares in $\mathbb{F}_{101}$. Indeed, $\mathrm{res}(H_1, H_2H_3) = 78 = 52^2$ and $\mathrm{res}(H_2, H_1H_3) = 79 = 68^2$. On $\mathcal{K}^{\mathrm{gen}}$, addition by the 2-torsion points corresponding to the factor H_1 is given by the matrix

$$M_1 = \begin{pmatrix} 6 & 18 & 39 & 1 \\ 92 & 13 & 15 & 86 \\ 17 & 52 & 22 & 13 \\ 46 & 54 & 52 & 60 \end{pmatrix},$$

with eigenvalues 52 and 49. For the 2-torsion points corresponding to H_2 addition is given by

$$M_2 = \begin{pmatrix} 58 & 100 & 96 & 1 \\ 60 & 91 & 13 & 48 \\ 32 & 15 & 52 & 83 \\ 66 & 22 & 58 & 1 \end{pmatrix},$$

with eigenvalues 33 and 68. The matrices M_1 and M_2 commute and so can be simultaneously diagonalised using the matrix

$$P = \begin{pmatrix} 1 & 37 & 4 & 88 \\ 1 & 13 & 6 & 65 \\ 1 & 76 & 66 & 99 \\ 1 & 12 & 30 & 91 \end{pmatrix},$$

which we find by following the procedure described in the paragraphs before Equation (4.4). Setting

$$\begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} = P^{-1} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix},$$

the General Kummer surface $\mathcal{K}^{\text{gen}}$ is transformed into a Kummer surface $\mathcal{K}$ given by equation:

$$\begin{aligned} \mathcal{K}\colon\, & 50X^4 + 57Y^4 + 27T^4 + 83Z^4 + 70X^2Y^2 + 10Z^2T^2 \\ & + 54X^2Z^2 + 91Y^2T^2 + 13X^2T^2 + 44Y^2Z^2 + 90XYZT. \end{aligned}$$

We see that $\mathcal{K}$ has a sparse defining equation, similar to the Fast Kummer model.

5. Computation of (N, N)-isogenies between Kummer surfaces

In this section, we give algorithms to compute (N, N)-isogenies, for N odd, between various Kummer surface models. For our methods we assume the biquadratic forms $B_{i,j}$ (for $1 \leq i, j \leq 4$) associated to our domain Kummer surface $\mathcal{K}$ are known and efficiently computable.

Let $\mathfrak{J}$ be a Jacobian of a hyperelliptic curve of genus 2 curve with corresponding Kummer surface $\mathcal{K}$. Let $R, S \in \mathfrak{J}[N]$ be points of order N on $\mathfrak{J}$ with images $k(R), k(S)$ on $\mathcal{K}$, respectively. If $\langle R, S\rangle \subset \mathfrak{J}[N]$ is a maximal isotropic subgroup, $\langle R, S\rangle$ is the kernel of an (N, N)-isogeny $\Phi : \mathfrak{J} \to \mathfrak{J}' = \mathfrak{J}/\langle R, S\rangle$ between Jacobians. This (N, N)-isogeny descends to a morphism of Kummer surfaces

$$\begin{aligned} \varphi : \mathcal{K} &\to \mathcal{K}' \\ (k_1, k_2, k_3, k_4) &\mapsto (k_1', k_2', k_3', k_4'). \end{aligned}$$

By abuse of terminology, we say φ is an (N, N)-isogeny between Kummer surfaces with kernel generated by N-torsion points $R, S \in \mathcal{K}[N]$. We denote the coordinates of R by (R_1, R_2, R_3, R_4), and similarly for S.

For the rest of the article, our aim is to compute this (N, N)-isogeny, where the Kummer surfaces are in a desired model. This can be separated into three main algorithms:

- Computing a basis for the space of degree N homogeneous forms that are invariant under addition by the N-torsion point R. We repeat this for the other kernel generator S, to obtain two bases B_R and B_S. These are constructed using the biquadratic forms associated to the Kummer surface.

- Computing a basis for the intersection of the two spaces generated by B_R and B_S. The intersection will have expected dimension 4, and a basis for this space will give the coordinates of a map $\psi : \mathcal{K} \to \widetilde{\mathcal{K}}$, which has kernel $\langle R, S\rangle$, but $\widetilde{\mathcal{K}}$ is not necessarily in desired form.
- Compute a linear map λ that brings $\widetilde{\mathcal{K}}$ into desired form $\mathcal{K}'$.

We remark that the first two algorithms are applicable to any model of Kummer surface with efficiently computable biquadratic forms. In this article, we show how to find the final scaling for the General and Fast Kummer surfaces.
Notation. We will give precise operation counts for our algorithms, and denote a multiplication, inversion, squaring, computing square roots, and addition in K by `M`, `I`, `S`, `Sq` and `a`.

Step 1: Find forms invariant under translation by an N-torsion point. Use the biquadratic forms B_{ij} associated to $\mathcal{K}$ to define the forms in k_1, k_2, k_3, k_4 of degree N which are invariant under addition by R, and the forms of degree N which are invariant under addition by S.

For general odd $N = 2n + 1$, let $P \in \mathcal{K}$ be a point and define

$$\begin{aligned} R_{ij}^{(\ell)}(P) &= B_{ij}(P, \ell R), \text{ for } \ell \in \{1, \dots, n\}, \\ S_{ij}^{(\ell)}(P) &= B_{ij}(P, \ell S), \text{ for } \ell \in \{1, \dots, n\}. \end{aligned} \tag{5.1}$$

Then, for $i_1, \dots, i_N \in \{1, \dots, 4\}$, we take

$$F_{R,N}\big((i_1, \dots, i_N), P\big) = \sum k_{i_1}(P) R_{i_2 i_3}^{(1)}(P) R_{i_4 i_5}^{(2)}(P) \cdots R_{i_{N-1} i_N}^{(n)}(P), \tag{5.2}$$

where the sum is taken over all permutations of $i_1, \dots, i_N$.

Lemma 5.1. *Let $(i_1, \dots, i_N) \in \{1, \dots, 4\}^N$. The homogeneous form $F_{R,N}\big((i_1, \dots, i_N), P\big)$ is of degree N and is invariant under translation-by-R.*

Proof. Firstly, each $R_{i,j}^{(\ell)}$ is a homogeneous form of degree 2, and so $F_{R,N}\big((i_1, \dots, i_N), P\big)$ is of degree $2\left(\frac{N-1}{2}\right) + 1 = N$. We now show that these forms are invariant under translation-by-R. Letting $I := (i_1, \dots, i_N) \in \{1, \dots, 4\}^N$, we define the set $\mathcal{S}_I$ as

$$\mathcal{S}_I := \left\{\sigma\big((i_1, i_2, i_4, \dots, i_{N-1}, i_N, \dots, i_5, i_3)\big) : \sigma \in \langle \mathrm{id}, (2\,3), \dots, (N-1\,N)\rangle\right\}.$$

We have

$$k_{i_1} R_{i_2 i_3}^{(1)} R_{i_4 i_5}^{(2)} \cdots R_{i_{N-1} i_N}^{(n)} = \sum_{(j_1, \dots, j_N) \in \mathcal{S}_I} k_{j_1}(P) k_{j_2}(P + R) \cdots k_{j_N}(P + (N-1)R).$$

Defining

$$q_{j_1, \dots, j_N}(P) := k_{j_1}(P) k_{j_2}(P + R) \cdots k_{j_N}(P + (N-1)R),$$

we verify that

$$\begin{aligned} q_{j_1, \dots, j_N}(P + R) &= k_{j_N}(P) k_{j_1}(P + R) \cdots k_{j_{N-1}}(P + (N-1)R) \\ &= q_{j_N, j_1, \dots, j_{N-1}}(P). \end{aligned}$$

Therefore, we find that

$$\begin{aligned} F_{R,N}\big(I, P+R\big) &= \sum_{I} \sum_{(j_1,\dots,j_N)\in\mathcal{S}_I} q_{j_1,\dots,j_N}(P+R) \\ &= \sum_{I} \sum_{(j_1,\dots,j_N)\in\mathcal{S}_I} q_{j_N,j_1,\dots,j_{N-1}}(P) \\ &= \sum_{I} \sum_{(j_1,\dots,j_N)\in\mathcal{S}_{\widetilde{I}}} q_{j_1,\dots,j_N}(P) \end{aligned}$$

where $\widetilde{I} = (i_N, i_1, \dots, i_{N-1})$ for $I = (i_1, \dots, i_N)$. Then

$$\begin{aligned} F_{R,N}\big(I, P+R\big) &= \sum_{I} k_{i_N} R^{(1)}_{i_1 i_2} \cdots R^{(n)}_{i_{N-2} i_{N-1}} \\ &= \sum_{I} k_{i_1} R^{(1)}_{i_2 i_3} \cdots R^{(n)}_{i_{N-1} i_N} \\ &= F_{R,N}\big(I, P\big), \end{aligned}$$

as required. □

Remark 5.2. For ease of notation, instead of $F_{R,N}\big((i_1,\dots,i_N),P\big)$ we will write $F_{R,N}(i_1,\dots,i_N)$ when the point $P \in \mathcal{K}$ is implicit.

As an optimisation, we may restrict that $i < j$ within each $R^{(\ell)}_{ij}$, and count this form several times in the sum (rather than recomputing it multiple times). By Lemma 5.1, $F_{R,N}(i_1,\dots,i_N)$ will generate a space X_R of homogeneous degree N forms that are invariant under translation by R. We similarly define the space X_S from the homogeneous forms $F_{S,N}(i_1,\dots,i_N)$.

In the next proposition, we show that these spaces have dimension $2(N+1)$.

Proposition 5.3. *Let $R \in \mathcal{K}[N]$ be a point of order N on $\mathcal{K}$. Let X_R be the space generated by homogeneous forms of degree N on $\mathcal{K}$ that are invariant under translation by R. Then* $\dim(X_R) = 2(N+1)$.

Proof. To prove this proposition we use the theory of theta functions. Let $\mathfrak{J}$ be a Jacobian of genus 2 curve corresponding to the Kummer surface $\mathcal{K}$. Let $\mathcal{L}$ be the symmetric line bundle giving rise to the principal polarisation on $\mathfrak{J}$.

The theta functions $\{\theta_\beta\}_{\beta\in(\mathbb{Z}/2N\mathbb{Z})^2}$, as defined in, for example, [FKM24, Definition 2.1] (taking $k = 2N$), form a basis for $H^0(\mathfrak{J}, \mathcal{L}^{2N})$. Note here β lies in $(\mathbb{Z}/2N\mathbb{Z})^2$ by its identification with a maximal isotropic subgroup of $\mathfrak{J}[2N]$.

The action of $[-1]^*$ on the global sections $\{\theta_\beta\}_{\beta\in(\mathbb{Z}/2N\mathbb{Z})^2}$ is described by the general transformation formula $\theta_\beta(-z) = \theta_{-\beta}(z)$. If θ_β is fixed under this action then β is a 2-torsion point in $(\mathbb{Z}/2N\mathbb{Z})^2$, i.e., $\beta \in \{(0,0),(N,0),(0,N),(N,N)\}$. Furthermore, a sum of these global sections is invariant under this action if and only if it is of the form

$$\sum_{\mathrm{ord}(\beta)\neq 2} \big[c_\beta(\theta_\beta + \theta_{-\beta})\big] + c_1\theta_{(0,0)} + c_2\theta_{(N,0)} + c_3\theta_{(0,N)} + c_4\theta_{(N,N)},$$

for some $c_\beta, c_1, c_2, c_3, c_4 \in \mathbb{Z}$.

The space of degree N homogeneous forms on $\mathcal{K}$ is generated by the subspace of $H^0(\mathfrak{J}, \mathcal{L}^{2N})$ that is invariant under the action of $[-1]^*$, and, by the discussion above, a basis for this space is given by

$$\left\{\theta_{(0,0)}, \theta_{(N,0)}, \theta_{(0,N)}, \theta_{(N,N)}\right\} \cup \left\{ \theta_{(b_1,b_2)} + \theta_{(-b_1,-b_2)} \;\middle|\; (b_1,b_2) \neq \begin{array}{c}(0,0),(N,N),\\(0,N),(N,0)\end{array} \right\}$$

The size of this basis is $4 + \frac{1}{2}(4N^2 - 4) = 2(N^2+1)$, and therefore the dimension of the space of degree N homogeneous forms on $\mathcal{K}$ is $2(N^2+1)$.

Given this basis, we now find the forms that generate X_R, i.e., the space of degree N homogeneous forms on $\mathcal{K}$ that are invariant under translation by an N-torsion point R.

Using the identity, for a suitable primitive $2N$-th root of unity ζ_{2N}, for an N-torsion point $r = (r_1, r_2) \in (\mathbb{Z}/2N\mathbb{Z})^2$ we have

$$\theta_{(b_1,b_2)}(z + r) = (\zeta_{2N})^{r_1 b_1 + r_2 b_2} \theta_{(b_1,b_2)}(z).$$

We immediately see that

$$\theta_{(0,N)}(z + r) = (\zeta_{2N})^{N \cdot r_2} \theta_{(b_1,b_2)}(z) = \theta_{(b_1,b_2)}(z),$$

as r_2 is an N-torsion point in $\mathbb{Z}/2N\mathbb{Z}$. This also holds for $\theta_{(N,0)}$, $\theta_{(0,0)}$, and $\theta_{(N,0)}$.

Note that $(r_1, r_2) \in \{(2,0),(0,2),(2,2)\}$. Take, for example $r = (2,0)$ and consider basis elements of the form $(\theta_{(b_1,b_2)} + \theta_{(-b_1,-b_2)})(z)$. By indepedence of the θ_β, it is invariant under translation-by-r if and only if

$$(\zeta_{2N})^{2b_1} = 1 \iff 2b_1 \equiv 0 \bmod 2N \iff b_1 = 0 \text{ or } N.$$

Therefore, a basis B_R for the space X_R is given by

$$\left\{\theta_{(0,0)}, \theta_{(N,0)}, \theta_{(0,N)}, \theta_{(N,N)}\right\} \cup \left\{ \theta_{(b_1,b_2)} + \theta_{(-b_1,-b_2)} \;\middle|\; \begin{array}{l} b_1 = 0, N \text{ and } b_2 \neq 0, N \\ \text{up to } (b_1,b_2) \mapsto (b_1,-b_2) \end{array} \right\}$$

For basis elements of the form $\theta_{(b_1,b_2)} + \theta_{(-b_1,-b_2)}$, we have 2 choices for b_1 and $(2N-1)/2 = N - 1$ choices for b_2. In particular, the basis has size $\dim(X_R) = 4 + 2(N-1) = 2(N+1)$. The same holds when $r = (0,2)$ or $(2,2)$. □

Conjecture 5.4. *A basis for the space X_R, defined as in Proposition 5.3 is given by the forms described in Equation (5.2) corresponding to indices $(i_1, \ldots, i_N) \in \mathcal{I}_N$ where*

$$\begin{aligned}\mathcal{I}_N := \big\{&\{1,1,\ldots,1,1\},\{1,1,\ldots,1,2\},\ldots,\{1,2,\ldots,2,2\},\{2,2,\ldots,2,2\},\\ &\{3,3,\ldots,3,3\},\{3,3,\ldots,3,4\},\ldots,\{3,4,\ldots,4,4\},\{4,4,\ldots,4,4\}\big\}.\end{aligned}$$

For General and Fast Kummer surfaces we have confirmed this experimentally for odd $N \leq 19$, and we therefore develop our algorithms under this assumption. We highlight, however, that if this is not the case for larger N or other Kummer surface models, one can compute the invariant forms corresponding to all possible indices $(i_1, \ldots, i_N) \in \{1,2,3,4\}^N$ and then use linear algebra to find a basis for the space X_R. The indices corresponding to a basis should be independent of the N-torsion point R, and this can therefore be computed once and used for all subsequent computations.

Remark 5.5. There are other choices of bases, such as the basis given by forms corresponding to the following indices:

$$\big\{\{1,1,\dots,1,1\},\{1,1,\dots,1,4\},\dots,\{1,4,\dots,4,4\},\{4,4,\dots,4,4\},$$
$$\{2,2,\dots,2,2\},\{2,2,\dots,2,3\},\dots,\{2,3,\dots,3,3\},\{3,3,\dots,3,3\}\big\}.$$

However, we found no computational advantage in choosing another basis. If future research reveals that another choice results in a better basis, the algorithms presented in this article can be easily adapted.

By Proposition 5.3 and assuming Conjecture 5.4, given above, we obtain a simple method for generating a basis of the spaces B_R and B_S, where R, S are N-torsion points generating the kernel of the (N,N)-isogeny, which we give in Algorithm 1. In Line 7 of Algorithm 1, the sum is taken over all permutations of I, as described in Equation (5.2). For our optimised implementation attached to this article given in [CF24a], we note that $R^{(\ell)}_{i,j} = R^{(\ell)}_{j,i}$ and avoid superfluous computations by computing the summand $k_{i_1} R^{(1)}_{i_2 i_3} R^{(2)}_{i_4 i_5} \dots R^{(n)}_{i_{N-1} i_N}$ only for $(i_1,\dots,i_N)$ such that $i_j \le i_{j+1}$ for $j > 1$ and $j \equiv 0 \bmod 2$, and scalar multiply by how many such summands occur in the sum for $F_{R,N}(I)$.

Algorithm 1 FindBasis: find a basis for X_R, R is a point of odd order N on $\mathcal{K}^{\text{fast}}_{a,b,c,d}$

Input: An N-torsion point R on the Kummer surface $\mathcal{K}$ with coordinates (k_1,k_2,k_3,k_4), multiples $(2R, 3R, \dots, ((N-1)/2)R)$ of the point R, and the list of indices $\mathcal{I}_N$.
Output: A basis B_R of size $2(N+1)$ generating X_R.
1: $B_R = \{\}$
2: $n = (N-1)/2$
3: **for** ℓ from 1 to n **do**
4: Compute $R^{(\ell)}_{ij}(k_1,k_2,k_3,k_4) = B_{ij}((k_1,k_2,k_3,k_4), \ell R)$ for all $i,j \in \{1,2,3,4\}$
5: **end for**
6: **for** I in $\mathcal{I}_N$ **do**
7: $F_{R,N}(I) = \sum k_{i_1} R^{(1)}_{i_2 i_3} R^{(2)}_{i_4 i_5} \dots R^{(n)}_{i_{N-1} i_N}$
8: Add $F_{R,N}(I)$ to B_R
9: **end for**
10: **return** B_R

Proposition 5.6. *The cost of finding the basis B_R using Algorithm 1 with $\mathcal{I}_N$ as in Conjecture 5.4 is bounded by*

$$\mathsf{Cost}_{basis} \le 9(3^{(N-1)/2} - 1)\,\mathtt{M}_{poly} + 2(3^{(N-1)/2} - N - 1)\,\mathtt{a}_{poly} + \frac{N-1}{2} \cdot \mathsf{Cost}_{biquad},$$

where $\mathtt{M}_{poly}$ and $\mathtt{a}_{poly}$ represents a multiplication and an addition in $K[k_1,k_2,k_3,k_4]$ (respectively) and Cost_{biquad} is the cost of computing the evaluated biquadratic form $R^{(\ell)}_{ij} = B_{ij}((k_1,k_2,k_3,k_4), \ell R)$ for some $1 \le \ell \le (N-1)/2$.

Proof. To compute the $F_{R,N}(I)$ for all

$$I \in \big\{\{1,1,\dots,1,1\},\{1,1,\dots,1,2\},\dots,\{1,2,\dots,2,2\},\{2,2,\dots,2,2\}\big\},$$

we first compute $k_{i_1} R^{(1)}_{i_2 i_3} R^{(2)}_{i_4 i_5} \dots R^{(n)}_{i_{N-1} i_N}$ for all $(i_1, \dots, i_N)$ with $i_k \in \{1, 2\}$ for all $k = 1, \dots, N$ and $i_j \leq i_{j+1}$ for $j > 1$ and $j \equiv 0 \bmod 2$.

We start with the forms $R^{(1)}_{1,1}, R^{(1)}_{1,2}, R^{(1)}_{2,2}$. The first step is to multiply each of these forms by $R^{(2)}_{1,1}, R^{(2)}_{1,2}$, and $R^{(2)}_{2,2}$ to obtain

$$\begin{gathered} R^{(1)}_{1,1}R^{(2)}_{1,1},\ R^{(1)}_{1,1}R^{(2)}_{1,2},\ R^{(1)}_{1,1}R^{(2)}_{2,2}, \\ R^{(1)}_{1,2}R^{(2)}_{1,1},\ R^{(1)}_{1,2}R^{(2)}_{1,2},\ R^{(1)}_{1,2}R^{(2)}_{2,2}, \\ R^{(1)}_{2,2}R^{(2)}_{1,1},\ R^{(1)}_{2,2}R^{(2)}_{1,2},\ R^{(1)}_{2,2}R^{(2)}_{2,2}. \end{gathered}$$

At step m we multiply the output of the previous step with $R^{(k+1)}_{1,1}, R^{(k+1)}_{1,2}$, and $R^{(k+1)}_{2,2}$. We continue this until $m = (N-1)/2 - 1$. Each step requires 3^{m+1} multiplications in $K[k_1, k_2, k_3, k_4]$, so to run all steps we need at most $\sum_{m=1}^{(N-1)/2} 3^{m+1} = \frac{9}{2}(3^{(N-1)/2} - 1)$ multiplications in $K[k_1, k_2, k_3, k_4]$. Then, we multiply each of these forms by k_1 and then by k_2 to obtain all forms needed to compute all the $F_{R,N}(I)$. Note that representing the homogeneous forms as an array of coefficients, this final step does not require any multiplications in K. We then construct all the $F_{R,N}(I)$ from these forms using $3^{(N-1)/2} - (N+1)$ additions in $K[k_1, k_2, k_3, k_4]$.

The same argument holds for indices in

$$\big\{\{3, 3, \dots, 3, 3\}, \{3, 3, \dots, 3, 4\}, \dots, \{3, 4, \dots, 4, 4\}, \{4, 4, \dots, 4, 4\}\big\},$$

and we obtain a cost of $9(3^{(N-1)/2} - 1)$ multiplications and $2 \cdot 3^{(N-1)/2} - 2(N+1)$ additions in $K[k_1, k_2, k_3, k_4]$ to compute the basis for X_R using $\mathsf{FindBasis}$ given biquadratics $R^{(\ell)}_{ij}$ for $1 \leq \ell \leq (N-1)/2$.

The cost of computing $R^{(\ell)}_{ij}$ for all $1 \leq \ell \leq (N-1)/2$ is $\frac{N-1}{2} \cdot \mathsf{Cost}_{\text{biquad}}$. Combining these costs, we get the upper bound given in the statement of the proposition. □

Remark 5.7. In Proposition 5.6, we do not give a precise cost for $\mathtt{M}_{\text{poly}}$ and $\mathsf{Cost}_{\text{biquad}}$ in terms of K-operations. Indeed, $\mathtt{M}_{\text{poly}}$ is hard to estimate as the size of the input is not fixed along the computation. Furthermore, $\mathsf{Cost}_{\text{biquad}}$ depends on the model of the Kummer surface we are working with. In Section 5.2.7, we give a precise upper bound for the cost of $\mathsf{Cost}_{\text{biquad}}$ for Fast Kummer surfaces.

Step 2: Compute the intersection. Once we have a bases B_R and B_S for the spaces X_R and X_S, we compute the intersection $X_{R,S}$ of expected dimension 4, with basis $B_{R,S} = \{\psi_1, \psi_2, \psi_3, \psi_4\}$. These basis elements will give the coordinates of the degree-N map ψ. To find the ψ_i, we consider the equation

$$\sum_{f \in B_R} c_f \cdot f - \sum_{g \in B_S} c_g \cdot g = 0, \tag{5.3}$$

where the coefficients c_f and $c_g \in K$. As this equation must hold for any point (k_1, k_2, k_3, k_4) on $\mathcal{K}$, the coefficients of each monomial in Equation (5.3) must be identically zero. This gives us a linear system over the ground field K in the coefficients $c_f, c_g \in \mathcal{K}$. To find ψ, we compute the kernel basis vectors

$$\left\{ \begin{array}{l} (c_{f,1} : f \in B_R,\ c_{g,1} : g \in B_S), (c_{f,2} : f \in B_R,\ c_{g,2} : g \in B_S), \\ (c_{f,3} : f \in B_R,\ c_{g,3} : g \in B_S), (c_{f,4} : f \in B_R,\ c_{g,4} : g \in B_S) \end{array} \right\}, \tag{5.4}$$

of the matrix corresponding to this linear system. Then each ψ_i is given by the sum $\psi_i = \sum_{f \in B_R} c_{f,i} \cdot f$. We summarise this in Algorithm 2. In Line 13 of Algorithm 2, on input a matrix M, $\mathsf{BasisForKernel}(M)$ will compute a basis for the kernel of that matrix.

Algorithm 2 FindIntersection: find a basis for the intersection $X_{R,S}$

Input: A basis B_R for the space X_R and a basis B_S for the space X_S
Output: The generator of the intersection $X_{R,S} = X_R \cap X_S$

1: Let `mons` be the monomials in B_R and B_S
2: $B_R = \{f_1, \ldots, f_{2(N+1)}\}$
3: $B_S = \{g_1, \ldots, g_{2(N+1)}\}$
4: **for** $j = 1$ to $4(N+1)$ **do**
5: **for** $i = 1$ to $2(N+1)$ **do**
6: Let $c_{i,j}$ be the coefficient of $\texttt{mons}_j$ in f_i
7: **end for**
8: **for** $i = 2(N+1)+1$ to $4(N+1)$ **do**
9: Let $c_{i,j}$ be the coefficient of $\texttt{mons}_j$ in $-g_i$
10: **end for**
11: **end for**
12: $M = (c_{i,j})_{i=1,\ldots,4(N+1), j=1,\ldots,2(N+1)}$
13: $(d_1, \ldots, d_{2(N+1)}) = \mathsf{BasisForKernel}(M)$
14: $F = \sum_{i=1}^{2(N+1)} d_i f_i$
15: **return** F

Proposition 5.8. *The cost of computing the intersection $X_{R,S} = X_R \cap X_S$ with Algorithm 2 is*

$$\mathsf{Cost}_\cap \leq \frac{2}{3}(N+1)(2N^3 + 44N^2 + 122N + 69)\,\mathtt{M} + \frac{2}{3}(8N+5)(4N^2 + 11N + 9)\,\mathtt{a}.$$

Proof. The linear system of equations given by

$$\sum_{f \in B_R} c_f \cdot f - \sum_{g \in B_S} c_g \cdot g$$

is a system of m equations in $\#B_R + \#B_S = 4(N+1)$ unknowns, where m is the number of monomials in B_R and B_S. For R, S generating the kernel of an (N,N)-isogeny, the dimension of the solution space will be $4(N+1)$, thus it suffices to consider $4(N+1)$ of the m equations. In this way, we can obtain the coefficients $c_f, c_g \in K$ using Gaussian elimitation. Since we work projectively, we may adapt Farebrother [Far88] to remove inversions (at the cost of more multiplications), we find that we find the coefficients with $\frac{64}{3}N^3 + 72N^2 + \frac{230}{3}N + 26$ multiplications, and $\frac{64}{3}N^3 + 88N^2 + \frac{314}{3}N + 38$ additions in K.

Recalling that $\#B_R = 2(N+1)$, constructing the basis of the intersection via $\psi_i = \sum_{f \in B_R} c_{f,i} f$ requires at most $8(N+1) \cdot m \leq 8(N+1)\binom{N+3}{3}$ multiplications and $8N+4$ additions in K.

Adding this to the cost of Gaussian elimination, we obtain the costs in the statement of the proposition. □

Step 4: Find the linear map to bring the image into desired form. From Steps 1 to 3, we have calculated the degree-N map

$$\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T) : \mathcal{K} \to \widetilde{\mathcal{K}},$$

with kernel $\langle R, S\rangle$. However, $\widetilde{\mathcal{K}}$ may not be in desired Kummer suface model. Therefore, we must apply a linear map $\lambda : \widetilde{\mathcal{K}} \to \mathcal{K}'$ to obtain the (N, N)-isogeny

$$\varphi = \lambda \circ \psi : \mathcal{K} \to \mathcal{K}', \quad (k_1, k_2, k_3, k_4) \mapsto (k_1', k_2', k_3', k_4')$$

with kernel $\langle R, S\rangle$, where $\mathcal{K}'$ is in the desired form.

Step 5: Find the image Kummer surface. The final step is to find the parameters defining the image Kummer surface.

The method for finding the scaling and image Kummer surface varies depending on what model we are working with. In this article we focus on the General and Fast Kummer surface models.

Over the next two sections, we go into detail on how to find isogenies between Kummer surfaces in the General and Fast models, providing examples in the case $N = 5$.

5.1. **Construction of (N, N)-isogenies on the General Kummer model.** We consider the case where $\mathcal{K} = \mathcal{K}^{\text{gen}}$ is in General Kummer form. We want to find an (N, N)-isogeny $\varphi : \mathcal{K}^{\text{gen}} \to \hat{\mathcal{K}}^{\text{gen}}$ with kernel $\langle R, S\rangle$ for $R, S \in \mathcal{K}^{\text{gen}}[N]$. Suppose we have followed Steps 1-3 to obtain a degree-N map

$$\begin{aligned}\psi = (\psi_1, \psi_2, \psi_3, \psi_4) \colon \mathcal{K}^{\text{gen}} &\to \widetilde{\mathcal{K}},\\ (k_1, k_2, k_3, k_4) &\mapsto (\ell_1, \ell_2, \ell_3, \ell_4).\end{aligned}$$

We must now find the linear map $\lambda : \widetilde{\mathcal{K}} \to \hat{\mathcal{K}}^{\text{gen}}$ to obtain the isogeny as $\varphi = \lambda \circ \psi$.

5.1.1. *Find the final linear map.* Examine the $k_1k_4^{N-1}, k_2k_4^{N-1}, k_3k_4^{N-1}, k_4^N$ terms in each ψ_i and perform a linear map so that ℓ_1 has only a $k_1k_4^{N-1}$ term (but not the others), ℓ_2 has only a $k_2k_4^{N-1}$ term (but not the others), ℓ_3 has only a $k_3k_4^{N-1}$ term (but not the others), ℓ_4 has only a k_4^N term (but not the others).

We now wish to write the quartic which is satisfied by $\ell_1, \ell_2, \ell_3, \ell_4$. To find the coefficients of this quartic, we use the formal power series in Equation (2.5), and then check it is correct. We expect the quartic to have the form

$$(\ell_2^2 - 4\ell_1\ell_3)\ell_4^2 + \mu_1(\ell_1, \ell_2, \ell_3)\ell_4 + \mu_0(\ell_1, \ell_2, \ell_3), \tag{5.5}$$

where μ_1 is cubic and μ_0 is quartic. By this we mean: initially all coefficients in $\mu_1(\ell_1, \ell_2, \ell_3)$ and $\mu_0(\ell_1, \ell_2, \ell_3)$ should be variables. We then replace k_1, k_2, k_3, k_4 with the power series in Equation (2.5). Any true identity should make the power series in s_1, s_2 equal to zero. This gives a set linear equations in the coefficients which allow us to solve quickly for $\mu_1(\ell_1, \ell_2, \ell_3)$ and $\mu_0(\ell_1, \ell_2, \ell_3)$.

If the defining equation has terms of the form $\ell_1\ell_2^2\ell_4$, $\ell_2^3\ell_4$, $\ell_2^2\ell_3\ell_4$, we apply a further linear map $(\ell_1, \ell_2, \ell_3, \ell_4) \mapsto (\ell_1, \ell_2, \ell_3, \ell_4')$, where $\ell_4' = \ell_4 - u_1\ell_1 - u_2\ell_2 - u_3\ell_3$, so that the equation satisfied by $\ell_1, \ell_2, \ell_3, \ell_4'$ no longer has these terms.

5.1.2. *Find the image General Kummer surface.* For Step 5, we now use Equation (2.4) to read off $f_0', \ldots, f_6'$ as the coefficients of $-4k_1^3$, $-2k_1^2k_2$, $-4k_1^2k_3$, $-2k_1k_2k_3$, $-4k_1k_3^2f_4$, $-2k_2k_3^2$, $-4k_3^3$, respectively, and then check that our quartic is indeed the General Kummer equation for the target curve $y^2 = f_6'x^2 + \ldots + f_0'$.

Example 5.9. Let $\mathcal{C} : y^2 = x^5 + 3x^4 + 9x^3 + 10x^2 + 9x + 3$ over the finite field $\mathbb{F}_{11}$. If we specialise Equation (2.4), we see that the General Kummer equation is

$$\begin{aligned}\mathcal{K}^{\text{gen}}\colon\ & (7k_1k_3 + k_2^2)k_4^2 \\ & + (10k_1^3 + 4k_1^2k_2 + 4k_1^2k_3 + 4k_1k_2k_3 + 10k_1k_3^2 + 9k_2k_3^2)k_4 \\ & + 5k_1^4 + k_3^4 + 3k_1^2k_2k_3 + 8k_1k_2^2k_3 + 4k_1k_2k_3^2 + k_1^2k_3^2 \\ & + 2k_1^3k2 + 3k_1^3k_3 + 8k_1^2k_2^2 + 10k_1k_2^3 + 4k_1k_3^3 = 0.\end{aligned} \tag{5.6}$$

There are two independent points of order 5, and their images on the General Kummer are $R = (0, 1, 4, 5)$ and $S = (0, 1, 0, 0)$. Note also that $2R = (1, 8, 5, 7)$ and $2S = (1, 0, 0, 5)$. After applying Steps 1 to 3, we obtain quintics, which are invariant under addition by R and by S. After adjusting them linearly to have the correct terms for $k_1k_4^4$, $k_2k_4^4$, $k_3k_4^4$ and k_4^5, we obtain $\psi : (k_1, k_2, k_3, k_4) \mapsto (\ell_1, \ell_2, \ell_3, \ell_4)$ where

$$\begin{aligned}\ell_1 = {} & 6k_1^5 + 9k_1^4k_2 + 4k_1^4k_3 + 3k_1^4k_4 + 5k_1^3k_2^2 + 6k_1^3k_2k_3 + 4k_1^3k_2k_4 \\ & + 5k_1^3k_3^2 + 7k_1^3k_3k_4 + 10k_1^3k_4^2 + 6k_1^2k_2^3 + k_1^2k_2^2k_3 + 8k_1^2k_2^2k_4 \\ & + 7k_1^2k_2k_3^2 + 9k_1^2k_2k_3k_4 + 4k_1^2k_2k_4^2 + 10k_1^2k_3^3 + 8k_1^2k_3^2k_4 + 8k_1^2k_3k_4^2 \\ & + 7k_1^2k_4^3 + 7k_1k_2^2k_3^2 + 2k_1k_2k_3^3 + 4k_1k_2k_3k_4^2 + 9k_1k_2k_4^3 + 2k_1k_3^4 \\ & + 10k_1k_3^3k_4 + 5k_1k_3^2k_4^2 + 9k_1k_3k_4^3 + k_1k_4^4 + k_2^3k_3^2 + 3k_2^2k_3^3 + k_2^2k_3^2k_4 \\ & + 3k_2k_3^3k_4 + 8k_2k_3^2k_4^2 + 7k_3^4k_4 + 7k_3^3k_4^2 + 6k_3^2k_4^3, \\ \ell_2 = {} & 3k_1^5 + 10k_1^4k_2 + 9k_1^4k_3 + 7k_1^4k_4 + 9k_1^3k_2^2 + 7k_1^3k_2k_3 + 2k_1^3k_2k_4 + 10k_1^3k_3^2 \\ & + 8k_1^3k_3k_4 + k_1^3k_4^2 + 7k_1^2k_2^2k_4 + 6k_1^2k_2k_3^2 + 2k_1^2k_2k_3k_4 + 2k_1^2k_2k_4^2 \\ & + 8k_1^2k_3^3 + 9k_1^2k_3k_4^2 + 9k_1^2k_4^3 + 2k_1k_2^3k_3 + k_1k_2^2k_3^2 + 4k_1k_2^2k_3k_4 \\ & + 2k_1k_2k_3^3 + 2k_1k_2k_3^2k_4 + k_1k_2k_3k_4^2 + 5k_1k_2k_4^3 + 10k_1k_3^4 + 8k_1k_3^3k_4 \\ & + 9k_1k_3^2k_4^2 + 3k_1k_3k_4^3 + 2k_2^3k_3k_4 + 10k_2^2k_3^3 + 10k_2^2k_3^2k_4 + 2k_2k_3^4 + 4k_2k_3^3k_4 \\ & + 6k_2k_3^2k_4^2 + 3k_2k_3k_4^3 + k_2k_4^4 + 3k_3^5 + 9k_3^4k_4 + 8k_3^3k_4^2, \\ \ell_3 = {} & 9k_1^5 + 3k_1^4k_2 + 9k_1^4k_3 + 7k_1^3k_2k_3 + 9k_1^3k_2k_4 + 8k_1^3k_3^2 + 7k_1^3k_3k_4 + 3k_1^3k_4^2 + k_1^2k_2^3 \\ & + k_1^2k_2^2k_3 + 5k_1^2k_2k_3^2 + 8k_1^2k_2k_3k_4 + 7k_1^2k_2k_4^2 + 5k_1^2k_3^3 + 2k_1^2k_3^2k_4 + 7k_1^2k_3k_4^2 \\ & + k_1^2k_4^3 + k_1k_2^4 + 10k_1k_2^3k_3 + k_1k_2^3k_4 + 7k_1k_2^2k_3^2 + k_1k_2^2k_3k_4 + 6k_1k_2k_3^2k_4 \\ & + 7k_1k_2k_3k_4^2 + 2k_1k_2k_4^3 + 6k_1k_3^4 + 4k_1k_3^3k_4 + 9k_1k_3^2k_4^2 + 9k_1k_3k_4^3 \\ & + k_2^2k_3^2k_4 + k_2k_3^2k_4^2 + 9k_2k_3k_4^3 + 9k_3^5 + 10k_3^4k_4 + 8k_3^3k_4^2 + 6k_3^2k_4^3 + k_3k_4^4, \\ \ell_4 = {} & 10k_1^4k_2 + k_1^4k_3 + 5k_1^4k_4 + k_1^3k_2^2 + 4k_1^3k_2k_3 + 9k_1^3k_2k_4 + 3k_1^3k_3^2 + 3k_1^3k_3k_4 + 9k_1^3k_4^2 \\ & + 10k_1^2k_2^3 + 8k_1^2k_2^2k_3 + 9k_1^2k_2^2k_4 + 5k_1^2k_2k_3^2 + 10k_1^2k_2k_3k_4 + 6k_1^2k_2k_4^2 + 10k_1^2k_3^3 \\ & + 6k_1^2k_3^2k_4 + 8k_1^2k_3^3 + 4k_1k_2^4 + 5k_1k_2^3k_3 + 5k_1k_2^3k_4 + 3k_1k_2k_3^3 + k_1k_2k_3^2k_4 \\ & + 5k_1k_2k_3k_4^2 + 10k_1k_2k_4^3 + 8k_1k_3^4 + 8k_1k_3^3k_4 + 6k_1k_3^2k_4^2 + 4k_1k_3k_4^3 + 9k_1k_4^4 \\ & + k_2^5 + 10k_2^4k_3 + k_2^4k_4 + 7k_2^3k_3^2 + 10k_2^3k_3k_4 + 2k_2^2k_3^3 + 8k_2^2k_3^2k_4 + 9k_2k_3^4 \\ & + 5k_2k_3^2k_4^2 + 10k_2k_3k_4^3 + 9k_2k_4^4 + 2k_3^5 + 10k_3^4k_4 + 6k_3^3k_4^2 + 2k_3^2k_4^3 + 9k_3k_4^4 + k_4^5.\end{aligned} \tag{5.7}$$

We now find the quartic which is satisfied by $\ell_1, \ell_2, \ell_3, \ell_4$. We set up the form in Equation (5.5) where, for the moment, the coefficients of μ_1 and μ_0 are variables. We then substitute Equation (2.5) into Equation (5.7), and then substitute these into Equation (5.5), truncating the power series at the degree 10 terms in s_1, s_2. The fact that the power series is identically zero gives us a set of linear equations which we solve in the coefficients of μ_1 and μ_0. This gives the following quartic in

$\ell_1, \ell_2, \ell_3, \ell_4$.

$$\begin{aligned}&\ell_4^2(\ell_2^2+7\ell_1\ell_3)\\&+(3\ell_1^3+5\ell_1^2\ell_2+7\ell_1\ell_2^2+9\ell_1\ell_2\ell_3+7\ell_1\ell_3^2+7\ell_2^3+7\ell_2^2\ell_3+9\ell_2\ell_3^2+2\ell_3^3)\ell_4\\&+4\ell_1^4+9\ell_1^2\ell_2^2+7\ell_1^2\ell_2\ell_3+7\ell_1^2\ell_3^2+8\ell_1\ell_2^2\ell_3\\&+4\ell_1\ell_2\ell_3^2+2\ell_1\ell_3^3+8\ell_2^4+3\ell_2^3\ell_3+5\ell_2^2\ell_3^2+7\ell_2\ell_3^3+7\ell_3^4.\end{aligned}\tag{5.8}$$

If we wish, we can verify that $\ell_1, \ell_2, \ell_3, \ell_4$ indeed satisfy this equation, by substituting Equation (5.7) into Equation (5.8) and checking that the result is indeed divisible by Equation (5.6).

This is now very close to the style of a General Kummer equation, but note that we have terms $\ell_1\ell_2^2\ell_4$, $\ell_2^3\ell_4$, $\ell_2^2\ell_3\ell_4$ which do not appear in Equation (2.4). We perform the final linear map $(\ell_1, \ell_2, \ell_3, \ell_4) \mapsto (\ell_1, \ell_2, \ell_3, \ell_4')$, finding ℓ_4' as follows. Substitute $\ell_4 = \ell_4' + u_1\ell_1 + u_2\ell_2 + u_3\ell_3$ and solve the linear equations in u_1, u_2, u_3 which make these terms disappear. This gives $u_1 = u_2 = u_3 = 2$, and so we define $\ell_4' = \ell_4 - 2\ell_1 - 2\ell_2 - 2\ell_3$. The following quartic is satisfied by $\ell_1, \ell_2, \ell_3, \ell_4'$.

$$\begin{aligned}&(7\ell_1\ell_3+\ell_2^2)(\ell_4')^2\\&+(3\ell_1^3+5\ell_1^2\ell_2+6\ell_1^2\ell_3+4\ell_1\ell_2\ell_3+2\ell_1\ell_3^2+9\ell_2\ell_3^2+2\ell_3^3)\ell_4'\\&+4\ell_2^4+6\ell_2^3\ell_3+4\ell_1^2\ell_2^2+3\ell_1\ell_2^3+5\ell_1^3\ell_2\\&+8\ell_2^2\ell_3^2+\ell_1^3\ell_3+3\ell_1^2\ell_2\ell_3+2\ell_1\ell_2^2\ell_3.\end{aligned}\tag{5.9}$$

We now read off the coefficients of $-4k_1^3$, $-2k_1^2k_2$, $-4k_1^2k_3$, $-2k_1k_2k_3$, $-4k_1k_3^2f_4$, $-2k_2k_3^2$, $-4k_3^3$, to see that $f_0' = 2$, $f_1' = 3$, $f_2' = 4$, $f_3' = 9$, $f_4' = 5$, $f_5' = 1$ and $f_6' = 5$. We then confirm that Equation (5.9) is indeed the General Kummer equation for the curve $y^2 = 5x^6 + x^5 + 5x^4 + 9x^3 + 4x^2 + 3x + 2$, and the $(5,5)$-isogeny is given by $(k_1, k_2, k_3, k_4) \mapsto (\ell_1, \ell_2, \ell_3, \ell_4')$. We have determined the $(5,5)$-isogeny and the equation of the target General Kummer, as required.

5.2. **Construction of (N,N)-isogenies on the Fast Kummer model.** Suppose we are given Fast Kummer $\mathcal{K}^{\text{fast}}_{a,b,c,d}$, as in Equation (4.1). To follow notation in previous literature (e.g., [Gau07]), we will let (X, Y, Z, T) be the coordinates of $\mathcal{K}^{\text{fast}}_{a,b,c,d}$ rather than $k_1, \ldots, k_4$ as before. We wish to compute the (N,N)-isogeny $\varphi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c',d'}$, $(X, Y, Z, T) \mapsto (X', Y', Z', T')$, with kernel generated by N-torsion points $R, S \in \mathcal{K}^{\text{fast}}$.

We first note that there will be many choices of such an isogeny, since there is a rich set of linear maps between Fast Kummer surfaces, and any (N,N)-isogeny can be composed with any of these to get a variant (N,N)-isogeny.

Addition by any of the 16 points of order 2 give the linear maps σ_i defined by Equation (4.8) from $\mathcal{K}^{\text{fast}}_{a,b,c,d}$ to itself. Furthermore, there is the *Hadamard* map

$$\begin{aligned}&\mathcal{H} : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c'd'}\\&(X, Y, Z, T) \mapsto (X+Y+Z+T, X+Y-Z-T,\\&\qquad\qquad X-Y+Z-T, X-Y-Z+T),\end{aligned}\tag{5.10}$$

where $(a', b', c', d') = \mathcal{H}(a, b, c, d)$. All of the above are defined over the ground field K and, given any (N,N)-isogeny between Fast Kummers, any of the above (on the target Kummer) can be composed with it to give a variant (N,N)-isogeny.

In the special case when there exists $i \in K$, where $i^2 = -1$, (for example, when $K = \mathbb{F}_p$ for $p \equiv 1 \bmod 4$), there is also the map $(X, Y, Z, T) \mapsto (X, Y, iZ, iT)$ from $\mathcal{K}^{\text{fast}}_{a,b,c,d}$ to $\mathcal{K}^{\text{fast}}_{a,b,ic,id}$.

Since our (N, N)-isogeny has odd degree, it must be an isomorphism on the entire 2-torsion subgroup of $\mathcal{K}^{\text{fast}}_{a,b,c,d}$. After composing with a combination of the linear maps above, we can force the diagonalised elements of order 2 to map to the diagonalised elements of order 2 on the target Fast Kummer and match them up so that they have the same effect. In other words, we can choose our (N, N)-isogeny so that $(X, Y, Z, T) \mapsto (X, Y, -Z, -T)$, $(X, Y, Z, T) \mapsto (X, -Y, Z, -T)$ and $(X, Y, Z, T) \mapsto (X, -Y, -Z, T)$ have the same effect on the target coordinates X', Y', Z', T'. So, we should be able to choose our isogeny so that the monomials can be partitioned in the same way as described in the previous section for $\phi^{(N)}_X, \phi^{(N)}_Y, \phi^{(N)}_Z, \phi^{(N)}_T$. Similarly $(X, Y, Z, T) \mapsto (Y, X, T, Z)$, $(X, Y, Z, T) \mapsto (Z, T, X, Y)$ and $(X, Y, Z, T) \mapsto (T, Z, Y, X)$ should have the same effect on the target coordinates X', Y', Z', T' (if only projectively, then applying any of these involutions twice forces the scalar to be ± 1). We will use this symmetry throughout this section to find the scaling, as well as accelerate Algorithm 2 when using Fast Kummer surfaces.

Remark 5.10. The biquadratic forms corresponding to the Fast Kummer surface $\mathcal{K}^{\text{fast}}$ are the simplest, when compared to those for $\mathcal{K}^{\text{sqr}}$ and $\mathcal{K}^{\text{gen}}$. Therefore, when constructing our (N, N)-isogenies from these biqudratics, we expect the isogenies between Fast Kummer surfaces to yield the most efficient and compact maps.

We now describe a method for computing these isogenies which finds the most natural version, namely the version in which addition by the points of order 2 on the initial Fast Kummer surface have the same effect on the coordinates of the target Fast Kummer surface.

Our input is: the parameters a, b, c, d of the initial Fast Kummer, together with two independent points R, S of order N, where N is odd. We will output the formulæ defining the (N, N)-isogeny $\varphi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c',d'}$ that takes (X, Y, Z, T) to (X', Y', Z', T') with kernel $\langle R, S \rangle$.

5.2.1. *Modification to FindIntersection.* We first discuss how we can accelerate FindIntersection (see Algorithm 2) using the action of the 2-torsion points on $\mathcal{K}^{\text{fast}}$. Suppose we have followed Step 1 in Section 5 to find the space X_R of degree N homogeneous forms which are invariant under addition by R and the space X_S of those invariant under addition by S. Note that we are now using the simpler biquadratic forms in Corollary 4.2. Let B_R and B_S be a basis for these spaces, respectively.

Rather than proceeding straight to Step 2, we partition the basis B_R into four parts $B^{(1)}_R$, $B^{(2)}_R$, $B^{(3)}_R$ and $B^{(4)}_R$, as follows:

$$
\begin{aligned}
B^{(1)}_R &:= \{f_R \in B_R \mid \sigma_1(f_R) = f_R,\ \sigma_2(f_R) = f_R,\ \sigma_3(f_R) = f_R\} \\
B^{(2)}_R &:= \{f_R \in B_R \mid \sigma_1(f_R) = f_R,\ \sigma_2(f_R) = -f_R,\ \sigma_3(f_R) = -f_R\} \\
B^{(3)}_R &:= \{f_R \in B_R \mid \sigma_1(f_R) = -f_R,\ \sigma_2(f_R) = f_R,\ \sigma_3(f_R) = -f_R\} \\
B^{(4)}_R &:= \{f_R \in B_R \mid \sigma_1(f_R) = -f_R,\ \sigma_2(f_R) = -f_R,\ \sigma_3(f_R) = f_R\},
\end{aligned}
$$

where σ_i is the linear map corresponding to the action of two-torsion point E_i, as defined by Equation (4.8). Refering to the explicit basis given in Conjecture 5.4, the parts $B^{(1)}$, $B^{(2)}$, $B^{(3)}$, $B^{(4)}$ of the basis B are given by the forms corresponding to the indices in $\mathcal{I}_N^{(1)}$, $\mathcal{I}_N^{(2)}$, $\mathcal{I}_N^{(3)}$, $\mathcal{I}_N^{(4)}$, respectively, where

$$\mathcal{I}_N^{(1)} := \Big\{\{i_1,\dots,i_N\} \in \mathcal{I}_N \mid i_j \in \{1,2\} \text{ and there are an even number of 2's}\Big\},$$
$$\mathcal{I}_N^{(2)} := \Big\{\{i_1,\dots,i_N\} \in \mathcal{I}_N \mid i_j \in \{1,2\} \text{ and there are an odd number of 2's}\Big\},$$
$$\mathcal{I}_N^{(3)} := \Big\{\{i_1,\dots,i_N\} \in \mathcal{I}_N \mid i_j \in \{3,4\} \text{ and there are an even number of 4's}\Big\},$$
$$\mathcal{I}_N^{(4)} := \Big\{\{i_1,\dots,i_N\} \in \mathcal{I}_N \mid i_j \in \{3,4\} \text{ and there are an odd number of 4's}\Big\}.$$

Using this characterisation, we can compute the partition as we run FindBasis (see Algorithm 1) and immediately output $B_R^{(1)}, B_R^{(2)}, B_R^{(3)}$, and $B_R^{(4)}$ for R. We proceed similarly with S to obtain $B_S^{(1)}$, $B_S^{(2)}$, $B_S^{(3)}$ and $B_S^{(4)}$.

For each part $i = 1,\dots,4$, intersect the spaces $X_R^{(i)}$ and $X_S^{(i)}$ (generated by $B_R^{(i)}$ and $B_S^{(i)}$, respectively), to obtain $X_{R,S}^{(i)}$. We expect each $X_{R,S}^{(i)}$ to be of dimension 1, and thus generated by a homogenous form, which will give the i-th coordinate of degree-N map $\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T)$. We can find this intersection using FindIntersectionPart, which on input $B_R^{(i)}, B_S^{(i)}$, will output the generator of $X_{R,S}^{(i)}$. It runs identically to FindIntersection except the dimensions of the linear system decrease by a factor of 4. As a result, FindIntersectionPart terminates in $\frac{1}{24}(N+1)(N+12)(N-1)$ `M` and $\frac{1}{24}(N+1)(N+6)(N-1)$ `a`. Running this for each part we terminate in at most $\frac{1}{6}(N+1)(N+12)(N-1)$ `M` and $\frac{1}{6}(N+1)(N+6)(N-1)$ `a` which improves on the cost quoted in Proposition 5.8, and is concretely much faster.

5.2.2. *General method for computing the final linear map.* At this stage, we have calculated the degree-N map

$$\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T) : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \widetilde{\mathcal{K}},$$

however $\widetilde{\mathcal{K}}$ may not be in desired Fast Kummer form. Therefore, we must apply a scaling map

$$\lambda : (X, Y, Z, T) \mapsto (\lambda_X X, \lambda_Y Y, \lambda_Z Z, \lambda_T T),$$

to obtain the (N, N)-isogeny

$$\varphi = \lambda \circ \psi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c',d'}, \quad (X, Y, Z, T) \mapsto (X', Y', Z', T')$$

with kernel $\langle R, S\rangle$.

To find the scaling values, we exploit the fact that the actions

$$\sigma_4 \colon (X, Y, Z, T) \mapsto (Y, X, T, Z),$$
$$\sigma_7 \colon (X, Y, Z, T) \mapsto (Z, T, X, Y),$$
$$\sigma_{12} \colon (X, Y, Z, T) \mapsto (T, Z, Y, X),$$

of two-torsion points $E_4 = (b, a, d, c)$, $E_8 = (c, d, a, b)$, and $E_{12} = (d, c, b, a)$ on $\mathcal{K}^{\text{fast}}_{a,b,c,d}$, have the same effect on the target coordinates. From this we obtain the following linear equations, which we can solve to obtain $(\lambda_X, \lambda_Y, \lambda_Z, \lambda_T)$:

$$\varphi\big(\sigma_i^{\mathcal{K}}(X, Y, Z, T)\big) = \sigma_i^{\mathcal{K}'}\big(\varphi(X, Y, Z, T)\big), \text{ for } i = 4, 8, 12, \tag{5.11}$$

where $\sigma^{\mathcal{K}}$ denotes the action of two-torsion points on $\mathcal{K}^{\text{fast}}_{a,b,c,d}$, and $\sigma^{\mathcal{K}'}$ on $\mathcal{K}^{\text{fast}}_{a',b',c',d'}$. Importantly, however, this equality only holds modulo the equation defining the domain Kummer surface.

For $N = 3$, the scaling map is given by Corte-Real Santos, Costello and Smith in [CCS24, §4]. In this case, finding the scaling map is straightforward as the isogeny formulæ is unaffected by working modulo the equation defining $\mathcal{K}^{\text{fast}}_{a,b,c,d}$ of degree 4. However, for odd $N \geq 5$, this no longer holds, and we must take care when finding this scaling.

In this setting, we discuss three distinct methods to compute the scaling map:

(1) a method for $N = 5$ which requires 62 K-multiplications.
(2) a method for $N \geq 7$, which requires running Gaussian elimination on a system of ℓ equations in $\ell+1$ unknowns, where $\ell \leq (N-1)(N-2)(N-3)/24$.
(3) a method for $N \geq 7$, which requires the computation of 2 square roots in K, 1 inverse in K and a few K-multiplications.

5.2.3. *Final scaling for $N = 5$.* We start by describing the first method for $N = 5$, summarised by Algorithm 3. We consider the first coordinate in Equation (5.11) taking $i = 4$, to obtain the equality

$$\lambda_X \psi_X(X, Y, Z, T) = \lambda_Y \psi_Y(Y, X, T, Z) + cX\mathcal{K}^{\text{fast}}(X, Y, Z, T),$$

for some constant $c \in K$, where by abuse of notation $\mathcal{K}^{\text{fast}}(X, Y, Z, T)$ represents the equation of the Kummer surface. We say that coefficients are *transparently equal* if the corresponding monomials are unaffected by the Kummer equations. In this way, we see that the coefficients of the YZT^3 term are transparently equal. More precisely, the coefficient of YZT^3 in $\lambda_X \psi_X(X, Y, Z, T)$ should be equal to the coefficient of YZT^3 in $\lambda_Y \psi_Y(Y, X, T, Z)$, namely the coefficient of XZ^3T in $\lambda_Y \psi_Y(X, Y, Z, T)$. As we are working projectively, we can set $\lambda_X = 1$, and thus deduce the value of λ_Y. We proceed similarly using Equation (5.11) with $i = 8$ and 12, to obtain the scaling values λ_Z and λ_T, respectively.

Algorithm 3 $\mathsf{Scaling}_5$: find the scaling map when $N = 5$

Input: A quintic map $\psi : \mathcal{K}^{\text{fast}} \to \widetilde{\mathcal{K}}$
Output: The isogeny $\varphi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c',d'}$ with kernel generated by R, S

$\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T)$
Let c_Y be the coefficient of YZT^3 in ψ_X
Let d_Y be the coefficient of XZ^3T in ψ_Y
Let c_Z be the coefficient of YZT^3 in ψ_X
Let d_Z be the coefficient of XY^3T in ψ_Z
Let c_T be the coefficient of XYT^3 in ψ_Z
Let d_T be the coefficient of XYZ^3 in ψ_T
$\alpha = d_Z \cdot d_T$
$\beta = d_Y \cdot c_Z$
$(\lambda_X, \lambda_Y, \lambda_Z, \lambda_T) = (d_Y\alpha,\ c_Y\alpha,\ d_T\beta,\ c_T\beta)$
$\varphi = (\lambda_X\psi_X, \lambda_Y\psi_Y, \lambda_Z\psi_Z, \lambda_T\psi_T)$
return φ

Proposition 5.11. *Algorithm 3 terminates with* 62 M.

Proof. Lines 8 to 10 in Algorithm 3 cost 6 M. Then, constructing the output isogeny φ in Line 11 requires 56 M. □

Example 5.12. We now illustrate the method for $N = 5$ with the following example. Our input is the Fast Kummer surface defined by parameters $(a, b, c, d) = (883, 375, 1692, 1586)$ over the finite field $\mathbb{F}_{1697}$, together with the points $R = (1593, 713, 1161, 1)$ and $S = (615, 1249, 125, 1)$ of order 5. We wish to compute the $(5, 5)$-isogeny φ with kernel $\langle R, S\rangle$. After applying Steps 1 to 3, we obtain the quintic map $\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T)$, where:

$$
\begin{aligned}
\phi_X =& 1668X^5 + 708X^3Y^2 + 1282X^3Z^2 + 1487X^3T^2 \\
&+ 823X^2YZT + 646XY^4 + 760XY^2Z^2 + 502XY^2T^2 + 632XZ^4 \\
&+ 1352XZ^2T^2 + 247XT^4 + 651Y^3ZT + 1154YZ^3T + 331YZT^3, \\
\phi_Y =& 1026X^4Y + 512X^3ZT + 556X^2Y^3 + 278X^2YZ^2 \\
&+ 7X^2YT^2 + 509XY^2ZT + 289XZ^3T + 136XZT^3 + 370Y^5 \\
&+ 975Y^3Z^2 + 1564Y^3T^2 + 329YZ^4 + 1026YZ^2T^2 + 942YT^4, \\
\phi_Z =& 259X^4Z + 19X^3YT + 1373X^2Y^2Z + 396X^2Z^3 \\
&+ 686X^2ZT^2 + 1101XY^3T + 1610XYZ^2T + 371XYT^3 + 660Y^4Z \\
&+ 1520Y^2Z^3 + 1539Y^2ZT^2 + 229Z^5 + 933Z^3T^2 + 121ZT^4, \\
\phi_T =& 397X^4T + 371X^3YZ + 610X^2Y^2T + 1326X^2Z^2T \\
&+ 1464X^2T^3 + 1073XY^3Z + 945XYZ^3 + 686XYZT^2 + 80Y^4T \\
&+ 1613Y^2Z^2T + 816Y^2T^3 + 593Z^4T + 770Z^2T^3 + 708T^5.
\end{aligned}
\tag{5.12}
$$

At the stage, we only require a further scaling of each of these to obtain X', Y', Z', T' such that the $(5, 5)$-isogeny is defined by

$$\varphi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c',d'}, \quad (X, Y, Z, T) \mapsto (X', Y', Z', T').$$

Let us now apply Step 4. We first let $\lambda_X, \lambda_Y, \lambda_Z, \lambda_T$ denote the scaling factors, so that our desired X', Y', Z', T' will be

$$X' = \lambda_X\psi_X,\ Y' = \lambda_Y\psi_Y,\ Z' = \lambda_Z\psi_Z,\ T' = \lambda_T\psi_Z. \tag{5.13}$$

We now use that we are making the choice of map such that $(X, Y, Z, T) \mapsto (Y, X, T, Z)$ has the same effect on (X', Y', Z', T'). Applying $(X, Y, Z, T) \mapsto (Y, X, T, Z)$ to $\lambda_Y\psi_Y$ and equating one of its coefficients to the corresponding coefficient in $\lambda_X\psi_X$, we obtain linear equation in λ_X, λ_Y. However, some care is required, since all equalities are modulo the initial Fast Kummer equation. For quintics within the first partition, this can only mean: modulo a constant time $X\mathcal{K}^{\text{fast}}_{a,b,c,d}(X, Y, Z, T)$, where here $\mathcal{K}^{\text{fast}}_{a,b,c,d}(X, Y, Z, T)$ is the equation of the Kummer equation. So, for example, the YZT^3 term is unaffected by this. Hence the coefficients of the YZT^3 terms should be transparently equal. The coefficient of YZT^3 in $\lambda_X\psi_X$ is $331\lambda_X$. The coefficient of YZT^3 in the image of $\lambda_Y\psi_Y$ under $(X, Y, Z, T) \mapsto (Y, X, T, Z)$ (which is the same as the coefficient of XZ^3T in $\lambda_Y\psi_Y$) is $289\lambda_Y$. This gives the equation $289\lambda_Y = 331\lambda_X$ and so $\lambda_Y = 283\lambda_X$ in our field $\mathbb{F}_{1697}$. We can now check that indeed if we take the entirety of 283 times the image of $283\psi_Y$ under $(X, Y, Z, T) \mapsto (Y, X, T, Z)$ and then subtract ψ_X, the result is divisible by $\mathcal{K}^{\text{fast}}_{a,b,c,d}$,

where $a = 883, b = 375, c = 1692, d = 1586$ are our inputted initial parameter values.

Similarly, the coefficient of XYT^3 in $\lambda_Z\psi_Z$ is $371\lambda_Z$. The coefficient of XYT^3 in the image of $\lambda_T\psi_T$ under $(X, Y, Z, T) \mapsto (Y, X, T, Z)$ (which is the same as the coefficient of XYZ^3 in $\lambda_T\psi_T$) is $945\lambda_T$. Within the third partition set, these cannot be affected by multiples of $\mathcal{K}^{\text{fast}}_{a,b,c,d}$, and so $371\lambda_Z = 945\lambda_T$, giving that $\lambda_T = 1270\lambda_Z$ in our field $\mathbb{F}_{1697}$.

We can also equate the coefficient of YZT^3 in $\lambda_X\psi_X$ with the same coefficient in the image of $\lambda_Z\psi_Z$ under $(X, Y, Z, T) \mapsto (Z, T, X, Y)$ to see that $\lambda_Z = 418\lambda_X$. We have now solved for the projective array of scaling factors:

$$(\lambda_X, \lambda_Y, \lambda_Z, \lambda_T) = (1, 283, 418, 1396), \tag{5.14}$$

and we have found the final versions of our maps:

$$X' = \psi_X,\ Y' = 283\psi_Y,\ Z' = 418\psi_Z,\ T' = 1396\psi_T. \tag{5.15}$$

Thus, we have found the $(5, 5)$-isogeny

$$\varphi : (X, Y, Z, T) \mapsto (X', Y', Z', T'),$$

completing Step 4. For the final step, we compute $\varphi(a, b, c, d)$ by substituting $(X, Y, Z, T) = (883, 375, 1692, 1586)$ into (X', Y', Z', T') to give that $(a', b', c', d') = (381, 960, 69, 1199)$.

5.2.4. *Final scaling for $N \geq 7$ with Gaussian elimination.* The next method to find the scaling for $N \geq 7$ follows the same procedure, however now there are no monomials whose coefficients will be transparently equal. Rather, they will be equal modulo the equation defining the domain Kummer surface. Instead, we must now perform Gaussian elimination to solve a system of equations for the scaling values $\lambda_X, \lambda_Y, \lambda_Z$ and λ_T. More precisely, we again consider Equation (5.11) and look at the first coordinate to obtain

$$\lambda_X\psi_X(X, Y, Z, T) - \lambda_Y\psi_Y(Y, X, T, Z) + G_Y\mathcal{K}^{\text{fast}}(X, Y, Z, T) = 0, \tag{5.16}$$

$$\lambda_X\psi_X(X, Y, Z, T) - \lambda_Z\psi_Z(Z, T, X, Y) + G_Z\mathcal{K}^{\text{fast}}(X, Y, Z, T) = 0, \tag{5.17}$$

$$\lambda_X\psi_X(X, Y, Z, T) - \lambda_T\psi_T(T, Z, Y, X) + G_T\mathcal{K}^{\text{fast}}(X, Y, Z, T) = 0, \tag{5.18}$$

where G_Y, G_Z, G_T are $(N-4)$-degree forms such that $G_\bullet\mathcal{K}^{\text{fast}}$ contains monomials in the first partition (i.e., are unaffected by the action of σ_1, σ_2 and σ_3). As this equality holds for any point (X, Y, Z, T) on $\mathcal{K}^{\text{fast}}_{a,b,c,d}$, we must have that the coefficients of each monomial are identically zero. This gives us a system of equations that we solve to obtain the scaling factors $\lambda_X, \lambda_Y, \lambda_Z$, and λ_T. We remark again that we can set $\lambda_X = 1$.

We summarise this method in Algorithm 4. In Line 2, on input an integer $n \geq 3$, $\mathsf{MonomialsInFirstPartition}(n)$, outputs the monomials of the n-degree homogenous forms in the 'first partition' (i.e., those unchanged by the action of σ_1, σ_2 and σ_3). Note that these monomials can be precomputed for each N; indeed, they do not depend on the kernel generators R and S. In Line 15, on input a matrix M, the algorithm $\mathsf{EchelonForm}$, outputs the matrix M in echelon form.

Proposition 5.13. *Algorithm 4 terminates in Cost_{GE}, where*

$$\mathsf{Cost}_{GE} \leq \frac{1}{6}\ell(\ell+1)(2\ell+13)\,\mathtt{M} + \frac{1}{6}\ell(\ell+1)(2\ell+7)\,\mathtt{a},$$

where $\ell \leq (N-1)(N-2)(N-3)/24$.

Algorithm 4 $\mathsf{Scaling}_{\mathrm{GE}}$: find the scaling map when $N \geq 7$ using Gaussian elimination

Input: A map of degree N, $\psi : \mathcal{K}^{\mathrm{fast}}_{a,b,c,d} \to \widetilde{\mathcal{K}}$, for $N \geq 7$.
Output: The isogeny $\varphi : \mathcal{K}^{\mathrm{fast}}_{a,b,c,d} \to \mathcal{K}^{\mathrm{fast}}_{a',b',c',d'}$ with kernel generated by R, S.

1: $\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T)$
2: $\mathtt{mons} = \mathsf{MonomialsInFirstPartition}(N-4)$
3: **for** $k = 2$ to 4 **do**
4: $\quad G = g_1 m_1 + \cdots + g_\ell m_\ell$ for monomials $m_j \in \mathtt{mons}$
5: $\quad$ Let $\mathcal{K}(X, Y, Z, T)$ be the equation defining $\mathcal{K}^{\mathrm{fast}}_{a,b,c,d}$
6: $\quad F = \lambda_k \psi_k(Y, X, T, Z) - \lambda_1 \psi_X(X, Y, Z, T) + G\mathcal{K}$
7: $\quad$ **for** $j = 1$ to ℓ **do**
8: $\quad\quad$ Let $c_j \in K[\lambda_k, g_1, \ldots, g_\ell, \lambda_1]$ be the coefficient of m_j in F
9: $\quad$ **end for**
10: $\quad \mathbf{v} = (\lambda_k, g_1, \ldots, g_\ell, \lambda_1)$
11: $\quad$ **for** $i = 1$ to $\ell + 1$ **do**
12: $\quad\quad$ Let $c_{i,j}$ be the coefficient of $v_i \in \mathbf{v}$ in c_j
13: $\quad$ **end for**
14: $\quad M = (c_{i,j})_{i=1,\ldots,\ell+1, j=1,\ldots,\ell}$
15: $\quad M = \mathsf{EchelonForm}(M)$
16: $\quad m = \mathsf{NumberOfColumns}(M)$
17: $\quad \lambda_k = M_{1,m}$
18: **end for**
19: $\varphi = (\psi_X, \lambda_2 \psi_Y, \lambda_3 \psi_Z, \lambda_4 \psi_T)$
20: **return** φ

Proof. To obtain the scaling factors $\lambda_X, \lambda_Y, \lambda_Z$, and λ_T, we solve a system of $\ell + 1$ equations in $\ell + 1$ unknowns, where

$$\ell \leq \frac{1}{4}\binom{N-1}{3} = \frac{(N-1)(N-2)(N-3)}{24}$$

is the number of monomials of degree $N-4$ in the homogeneous forms that are left unchanged by the action of σ_1, σ_2 and σ_3. Therefore, as in the proof of Proposition 5.8, the cost of solving this system using Gaussian elimination is $\frac{1}{6}\ell(\ell+1)(2\ell+13)$ multiplications and $\frac{1}{6}\ell(\ell+1)(2\ell+7)$ additions in K.

In Line 19, we calculate $(\psi_X, \lambda_Y\psi_Y, \lambda_Z\psi_Z, \lambda_T\psi_T)$ with at most 3ℓ K-multiplications, where ℓ is the maximum of the number of monomials in ψ_Y, ψ_Z and ψ_T. Namely, $\ell \leq (N+1)(N+2)(N+3)/24$.

□

5.2.5. *Final scaling for $N \geq 7$ with square roots.* We are now ready to describe the final method for $N \geq 7$. Here, we consider Equation (5.13) when evaluated at (a, b, c, d). Namely,

$$(a', b', c', d') = \big(\lambda_X \psi_X(a,b,c,d),\ \lambda_Y \psi_Y(a,b,c,d),\ \lambda_Z \psi_Z(a,b,c,d),\ \lambda_T \psi_T(a,b,c,d)\big). \tag{5.19}$$

Applying the action σ_1, we also have that

$$(b', a', d', c') = \big(\lambda_X\psi_X(b,a,d,c), \lambda_Y\psi_Y(b,a,d,c), \lambda_Z\psi_Z(b,a,d,c), \lambda_T\psi_T(b,a,d,c)\big). \tag{5.20}$$

Looking at the first and second coordinates in the equations above, we have

$$\frac{\lambda_X\psi_X(a,b,c,d)}{\lambda_Y\psi_Y(a,b,c,d)} = \frac{a'}{b'} \text{ and } \frac{\lambda_X\psi_X(b,a,d,c)}{\lambda_Y\psi_Y(b,a,d,c)} = \frac{b'}{a'}.$$

So, we see that

$$\left(\frac{\lambda_Y}{\lambda_X}\right)^2 = \frac{\psi_X(a,b,c,d)\psi_X(b,a,d,c)}{\psi_Y(a,b,c,d)\psi_Y(b,a,d,c)}, \tag{5.21}$$

and

$$\left(\frac{b'}{a'}\right)^2 = \frac{\psi_X(b,a,d,c)\psi_Y(a,b,c,d)}{\psi_X(a,b,c,d)\psi_Y(b,a,d,c)}. \tag{5.22}$$

Looking at the third and fourth coordinates of equations Equation (5.19) and Equation (5.20), we obtain $(\lambda_T/\lambda_Z)^2$ and $(d'/c')^2$. Similarly, we can apply the action σ_3 to obtain $(\lambda_Z/\lambda_Y)^2$, $\lambda_X\lambda_Z/(\lambda_Y\lambda_T)$, $(c'/b')^2$, and $a'c'/(b'd')$.

From $(\lambda_Y/\lambda_X)^2$, $(\lambda_T/\lambda_Z)^2$, and $\lambda_X\lambda_Z/(\lambda_Y\lambda_T)$, we can obtain the scalings λ_X, λ_Y, λ_Z, and λ_T as described in Algorithm 5.

Algorithm 5 $\mathsf{Scaling}_{\mathrm{sqrt}}$: find the scaling map when $N \geq 7$ using square roots in K

Input: A map of degree N, $\psi : \mathcal{K}^{\mathrm{fast}}_{a,b,c,d} \to \widetilde{\mathcal{K}}$, for $N \geq 7$.
Output: The isogeny $\varphi : \mathcal{K}^{\mathrm{fast}}_{a,b,c,d} \to \mathcal{K}^{\mathrm{fast}}_{a',b',c',d'}$ with kernel generated by R, S.

$\psi = (\psi_X, \psi_Y, \psi_Z, \psi_T)$
$a = \psi_X(a,b,c,d)\psi_X(b,a,d,c)\big(\psi_Y(a,b,c,d)\psi_Y(b,a,d,c)\big)^{-1}$
$b = \psi_Z(a,b,c,d)\psi_Z(b,a,d,c)\big(\psi_T(a,b,c,d)\psi_T(b,a,d,c)\big)^{-1}$
$s_a = \mathsf{Sqrt}(a)$
$s_b = \mathsf{Sqrt}(b)$
$\gamma = \psi_X(a,b,c,d)\psi_Z(d,c,b,a)\big(\psi_Y(a,b,c,d)\psi_T(d,c,b,a)\big)^{-1}$
for $\alpha \in \{s_a, -s_a\}$ **do**
  for $\beta \in \{s_b, -s_b\}$ **do**
    $\lambda_Y = \alpha$
    $\lambda_Z = \alpha\beta$
    $\lambda_T = \beta\gamma$
    $\varphi = (\psi_X, \lambda_Y\psi_Y, \lambda_Z\psi_Z, \lambda_T\psi_T)$
    if $\sigma_i\big(\varphi(X,Y,Z,T)\big) = \varphi\big(\sigma_i(X,Y,Z,T)\big)$ for $i = 1,2,3$ **then**
      return φ
    end if
  end for
end for
return $\perp$

Proposition 5.14. *Algorithm 5 terminates in* Cost_{sqrt}*, where*

$$\mathsf{Cost}_{sqrt} \leq 2\ \mathtt{Sq} + 1\ \mathtt{I} + (10 + 12\ell)\ \mathtt{M},$$

where $\ell \leq (N+1)(N+2)(N+3)/24$.

Proof. Algorithm 5 requires the computation of 2 square roots in K, as well as 8 K-mutliplications and inverses of $\psi_Y(a,b,c,d)\psi_Y(b,a,d,c)$, $\psi_T(a,b,c,d)\psi_T(b,a,d,c)$ and $\psi_Y(a,b,c,d)\psi_T(d,c,b,a)$ in K. Using batched inversion [Mon87, §10.3.1], we can compute these 3 inverses with 2 multiplications and 1 inverse in K.

In Line 12, we use 3ℓ $\mathtt{M}$ to calculate $(\psi_X, \lambda_Y\psi_Y, \lambda_Z\psi_Z, \lambda_T\psi_T)$, where $\ell \leq (N+1)(N+2)(N+3)/24$, as described in the proof of Proposition 5.13. We run this at most 4 times, thus requiring at most 12ℓ K-multiplications. □

Remark 5.15. The most costly operation in Algorithm 5 is the computation of square roots using Sqrt. When $\mathrm{char}(K) = p$, and $K = \mathbb{F}_{p^m}$ is a finite field (for some $m \in \mathbb{N}$), we compute square roots using the Tonelli–Shanks algorithm [Ton91, Sha73] using Scott's optimisation in [Sco20]. This costs 2 exponentiations and a few multiplications and additions in K.

When $K = \mathbb{Q}$, we can use, for example, the 'Karatsuba Square Root' algorithm, as depicted by Brent and Zimmermann [BZ10, Algorithm 1.12], for computing square roots in $\mathbb{Z}$ (as used in the GNU Multiple Precision Arithmetic Library [GMP]) on the numerator and denominator of our rational number. This costs $O(\frac{3}{2}M(N/2))$, where $M(n)$ is the time to multiply two numbers of n limbs using Karatsuba multiplication and $O(6M(N/2))$ when using FFT multiplication.

Using the previous method depicted in Algorithm 5, it is not necessary to compute square roots in K if the only thing required is to compute the constants $E', F', G', H' \in K$ appearing in the equation defining the image Kummer surface. Indeed, from $(b'/a')^2$, $(d'/c')^2$, $(c'/b')^2$, and $a'c'/(b'd')$ we can compute (a^2, b^2, c^2, d^2) (projectively). This leads us to the definition of Algorithm 6, which on input ψ will output the constants (E', F', G', H'). In this way, we only require the square roots in order to compute the isogeny φ, and therefore to push points through the isogeny.

Algorithm 6 terminates in at most 34$\mathtt{M}$, 4$\mathtt{S}$, 1$\mathtt{I}$, and 20$\mathtt{a}$. Note, here we are computing the 7 inverses in Lines 2 to 4, Line 7, and Lines 11 to 13 using batched inversions [Mon87, §10.3.1].

5.2.6. *Compute the image constants* (a', b', c', d'). We now have the (N, N)-isogeny $\varphi = \lambda \circ \psi$ where

$$\lambda : (X, Y, Z, T) \mapsto (\lambda_X X, \lambda_Y Y, \lambda_Z Z, \lambda_T T),$$

is the scaling map obtained from the previous step. We compute (a', b', c', d') by evaluating the (N, N)-isogeny φ at $(X, Y, Z, T) = (a, b, c, d)$. This costs at most $4\left(\binom{N+3}{3} + N + 1\right) = O(N^3)$ multiplications in K. Indeed, it costs $4(N+1)$ K-multiplications to compute all powers of a, b, c, d up to a^N, b^N, c^N, d^N, and then at most $3\binom{N+3}{3}$ K-multiplications to evaluate all the monomials in φ from these. Finally, it costs at most $\binom{N+3}{3}$ K-multiplications to multiply these evaluated monomials with their coefficients.

Algorithm 6 GetImage: find the constants in the equation defining the image of an (N, N)-isogeny when $N \geq 7$

Input: A map of degree N, $\psi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \widetilde{\mathcal{K}}$, for $N \geq 7$.
Output: The constants (E', F', G', H') in the equation defining $\mathcal{K}^{\text{fast}}_{a',b',c',d'}$

1: $\psi = (\psi_1, \psi_2, \psi_3, \psi_4)$
2: $\alpha_{21} = \psi_1(b,a,d,c)\psi_2(a,b,c,d)\big(\psi_1(a,b,c,d)\psi_2(b,a,d,c)\big)^{-1}$
3: $\alpha_{43} = \psi_3(b,a,d,c)\psi_4(a,b,c,d)\big(\psi_3(a,b,c,d)\psi_4(b,a,d,c)\big)^{-1}$
4: $\alpha_{32} = \psi_2(d,c,b,a)\psi_3(a,b,c,d)\big(\psi_2(a,b,c,d)\psi_3(d,c,b,a)\big)^{-1}$
5: $\alpha_{31} = \alpha_{32} \cdot \alpha_{21}$
6: $\alpha_{41} = \alpha_{43} \cdot \alpha_{31}$
7: $\beta = \psi_1(a,b,c,d)\psi_2(d,c,b,a)\big(\psi_2(a,b,c,d)\psi_1(d,c,b,a)\big)^{-1}$
8: $(a_2, b_2, c_2, d_2) = (1, \alpha_{21}, \alpha_{31}, \alpha_{41})$
9: $A, B, C, D = \mathcal{H}(a_2, b_2, c_2, d_2)$
10: $(a_4, b_4, c_4, d_4) = \mathcal{S}(a_2, b_2, c_2, d_2)$
11: $\gamma_1 = (a_2 d_2 - b_2 c_2)^{-1}$
12: $\gamma_2 = (a_2 c_2 - b_2 d_2)^{-1}$
13: $\gamma_3 = (a_2 b_2 - c_2 d_2)^{-1}$
14: $E = \beta\gamma_1\gamma_2\gamma_3 b_2 d_2 ABCD$
15: $F = \gamma_1 \cdot (a_2^2 - b_2^2 - c_2^2 + d_2^2)$
16: $G = \gamma_2 \cdot (a_2^2 - b_2^2 + c_2^2 - d_2^2)$
17: $H = \gamma_3 \cdot (a_2^2 + b_2^2 - c_2^2 - d_2^2)$
18: **return** (E, F, G, H)

5.2.7. *Complexity of finding an (N, N)-isogeny between Fast Kummer surfaces.* Putting together all steps of the algorithm we obtain GetIsogeny, given by Algorithm 7, which on input N-torsion points R, S on Fast Kummer surface $\mathcal{K}^{\text{fast}}_{a,b,c,d}$ generating a maximal isotropic subgroup of N-torsion group of $\mathcal{K}^{\text{fast}}$, will output the (N, N)-isogeny φ with kernel $\langle R, S\rangle$. Here, we assume Conjecture 5.4, as otherwise the bases formed in Lines 4 and 5 may be wrong. The costs $\mathsf{Cost}_{\text{GE}}$, $\mathsf{Cost}_{\text{sqrt}}$ in Line 13 are as defined in Proposition 5.13 and Proposition 5.14, respectively.

Asymptotically, Algorithm 7 is dominated by the call FindBasis in Lines 4 and 5. The cost of obtaining multiples of R, namely $2R, \ldots, ((N-1)/2)R$, in Line 1 of Algorithm 7 is at most $\sum_{n=2}^{(N-1)/2} 9\lceil \log_2(n)\rceil$ S and $\sum_{n=2}^{(N-1)/2} 16\lceil \log_2(n)\rceil$ M [Gau07, Theorem 3.6]. Furthermore, reducing the basis elements in B_R and B_S modulo the Kummer equation in Line 6 costs at most a handful of additions.

Our optimised implementation shows that in the case of Fast Kummer surfaces $\mathsf{Cost}_{\text{biquad}} \leq 12\mathtt{S} + 43\mathtt{M} + 25\mathtt{a}$. Therefore, we have that the bottleneck step (asymptotically) costs

$$\mathsf{Cost}_{\text{basis}} \leq (3^{(N+3)/2} + 4 \cdot 3^{(N-1)/2} - 27)\mathtt{M}_{\text{poly}} + 6(N-1)\mathtt{S} + \frac{43}{2}(N-1)\mathtt{M} + \frac{25}{2}(N-1)\mathtt{a}$$

for such Kummer surface models.

Algorithm 7 GetIsogeny: find the isogeny φ with kernel generated by N-torsion points R, S on $\mathcal{K}^{\text{fast}}$ assuming Conjecture 5.4

Input: N-torsion points R, S on Fast Kummer surface $\mathcal{K}^{\text{fast}}_{a,b,c,d}$
Output: The isogeny $\varphi : \mathcal{K}^{\text{fast}}_{a,b,c,d} \to \mathcal{K}^{\text{fast}}_{a',b',c',d'}$ with kernel generated by R, S.

1: $\mathbf{R} = (2R, 3R, \ldots, ((N-1)/2)R)$
2: $\mathbf{S} = (2S, 3S, \ldots, ((N-1)/2)S)$
3: $\mathcal{I}_N = \{(1, \ldots, 1), (1, \ldots, 2), \ldots, (2, \ldots, 2), (3, \ldots, 3), (3, \ldots, 4), \ldots, (4, \ldots, 4)\}$
4: $B_R^{(1)}, B_R^{(2)}, B_R^{(3)}, B_R^{(4)} = \mathsf{FindBasis}(R, \mathbf{R}, \mathcal{I}_N)$
5: $B_S^{(1)}, B_S^{(2)}, B_S^{(3)}, B_S^{(4)} = \mathsf{FindBasis}(S, \mathbf{S}, \mathcal{I}_N)$
6: Reduce forms in B_R and B_S modulo the Kummer equation defining $\mathcal{K}^{\text{fast}}$
7: **for** $i = 1$ to 4 **do**
8: $\quad \psi_i = \mathsf{FindIntersection}(B_R^{(i)}, B_S^{(i)})$
9: **end for**
10: $\psi = (\psi_1, \psi_2, \psi_3, \psi_4)$
11: **if** $N = 5$ **then**
12: $\quad$ **return** $\mathsf{Scaling}_5(\psi)$
13: **else if** $\mathsf{Cost}_{\text{GE}} \leq \mathsf{Cost}_{\text{sqrt}}$ **then**
14: $\quad$ **return** $\mathsf{Scaling}_{\text{GE}}(\psi)$
15: **else**
16: $\quad$ **return** $\mathsf{Scaling}_{\text{sqrt}}(\psi)$
17: **end if**
18: **return** $\perp$

6. Implementation and Performance

In this section, we investigate the performance of our algorithms when applied to Fast Kummer surfaces, as implemented in [CF24a].

We will fix $K = \mathbb{F}_{p^m}$ to be a finite field, for some $m \in \mathbb{N}$. To set up our experiments, we consider p be a prime such that $2^4 N | p + 1$. By restricting to superspecial Fast Kummer surfaces, which are therefore defined over $\mathbb{F}_{p^2}$, we can fix $m = 2$ and our choice of prime ensures that we have full K-rational 2-torsion and K-rational N-torsion that will generate the (N, N)-isogeny. The experiments were run in `MAGMA` V.2.25-6 on Intel(R) Core$^{\text{TM}}$ i7-1065G7 CPU @ 1.30GHz $\times$ 8 with 15.4 GiB memory.

Remark 6.1. Restricting our experiments to superspecial Kummer surfaces is inspired by the setup of various cryptographic primitives constructed in isogeny-based cryptography (see, for example, [FT19, CDS20, CCS24]), and allows us to easily obtain $\mathbb{F}_{p^2}$-rational (N, N)-isogenies. We replicate this setup to demonstrate the efficiency of our algorithms, but emphasise that the methods presented in this paper hold for more general fields K. We remark, however, that the performance of our methods for large N may be hindered by coefficient blow-up for these more general fields.

6.1. Evaluating the scaling algorithms.

We first analyse Method (2) and (3) of finding the final scaling map described in Sections 5.2.4 and 5.2.5, repsectively. In particular, we investigate how $\mathsf{Cost}_{\text{GE}}$ and $\mathsf{Cost}_{\text{sqrt}}$ vary with N and p. This is necessary to precisely determine the condition in Line 13 of Algorithm 7.

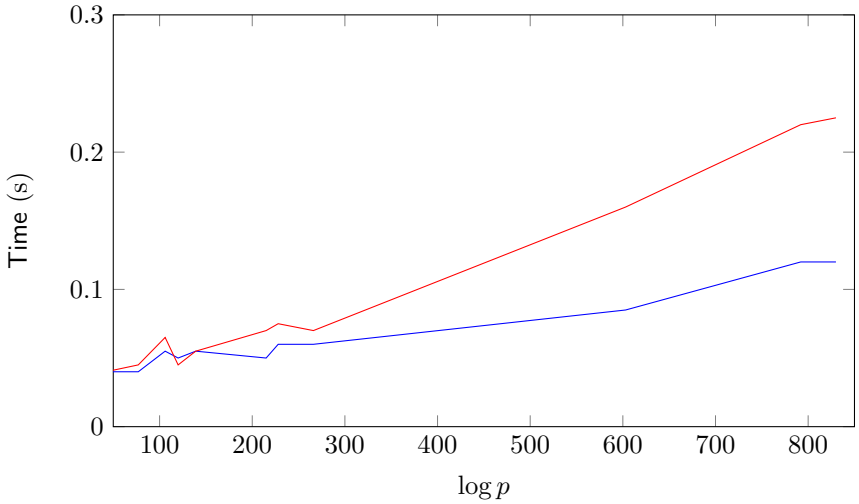

FIGURE 1. The time taken (in seconds) for GetIsogeny with method 2 of scaling $\mathsf{Scaling}_{\mathrm{GE}}$ (in blue) and with method 3 of scaling$\mathsf{Scaling}_{\mathrm{sqrt}}$ (in red) for a range of odd primes p and fixed prime $N = 7$. For each prime p, we average the time taken over 50 runs.

Using a cost metric of $1\mathtt{I} = \log_2(p)\mathtt{M}$ and $1\mathtt{a} = 0\mathtt{M}$ for $K = \mathbb{F}_{p^2}$, Proposition 5.13 tells us that

$$\mathsf{Cost}_{\mathrm{GE}} \leq \frac{N^9}{41472} + O(N^8)\,\mathtt{M}$$

Additionally, Proposition 5.14 and Remark 5.15 show that $\mathsf{Cost}_{\mathrm{sqrt}}$ is approximately equal to 4 exponentiations and 1 inversion in $\mathbb{F}_{p^2}$, for which the cost is around $(5\log_2(p) + (N+1)(N+2)(N+3)/2 + 10)\,\mathtt{M}$. From this, we expect that $\mathsf{Cost}_{\mathrm{GE}} \leq \mathsf{Cost}_{\mathrm{sqrt}}$ for N and p such that

$$N^9 - 20736N^3 - 124416N^2 - 228096N - 539136 \leq 207360\log_2(p).$$

For example, when $N = 7$, we have that $\mathsf{Cost}_{\mathrm{GE}} \leq \mathsf{Cost}_{\mathrm{sqrt}}$ for $\log_2(p) \geq 121$.

To confirm the observations from our theoretical costs, we run two sets of experiments: (1) fix $N = 7$, and vary p, where $\log_2(p)$ ranges from 12 to 850; (2) fix p where $\log_2(p) = 100$, and vary N from 7 to 19. The results are shown in Figure 1 and Figure 2, respectively.

In Figure 1, we confirm that for $N = 7$, $\mathsf{Cost}_{\mathrm{GE}} = \mathsf{Cost}_{\mathrm{sqrt}}$ for $\log_2(p) \approx 140$, which is close to our theoretical estimate. We remark that for larger N, this crossover point will be larger.

In Figure 2 we see that the cost of $\mathsf{Scaling}_{\mathrm{sqrt}}$ increases at a slower rate with N than the cost of $\mathsf{Scaling}_{\mathrm{GE}}$, as predicted by our theoretical costs. We also ran experiments for $\log_2(p) \approx 500$ and observed the same trend.

6.2. **Evaluation performance of algorithms.** In Table 1, we give timings for different sections of Algorithm 7 to compute an (N, N)-isogeny given N-torsion points R, S generating kernel on domain Kummer surface $\mathcal{K}^{\mathrm{fast}}$ defined over K for $N \geq 7$.

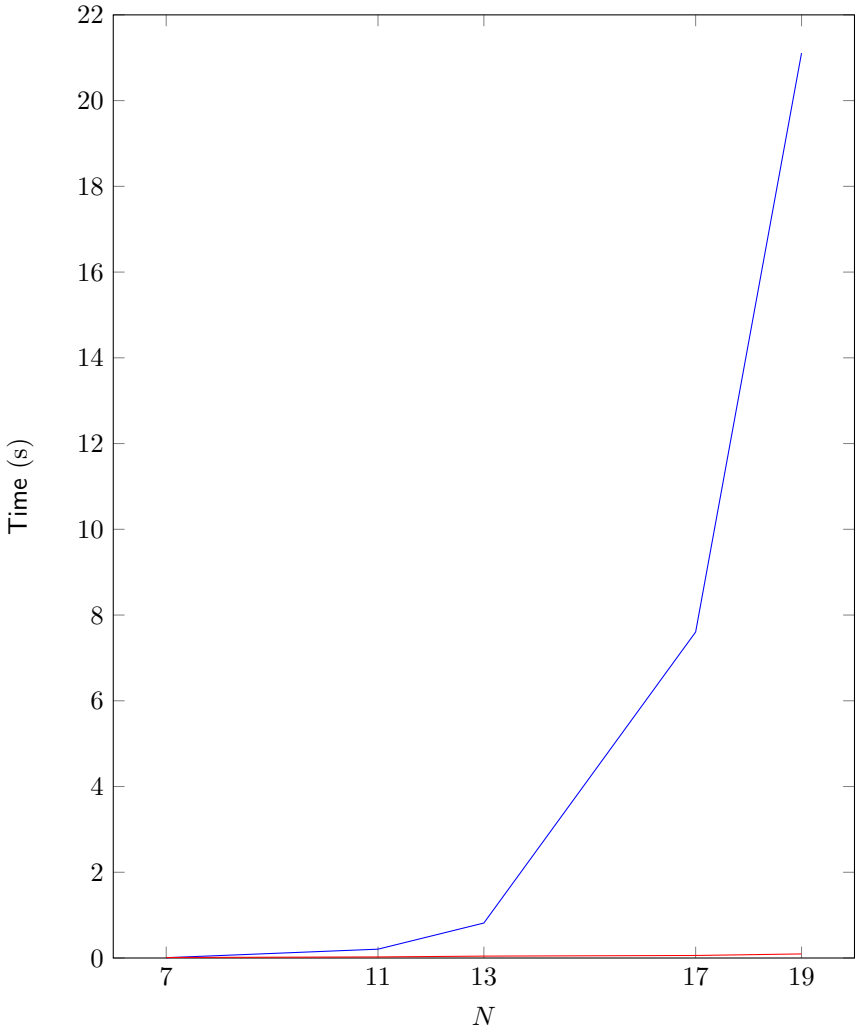

FIGURE 2. The time taken (in seconds) of method 2 of scaling $\mathsf{Scaling}_{\mathrm{GE}}$ (in blue) and method 3 of scaling $\mathsf{Scaling}_{\mathrm{sqrt}}$ (in red) for a range of odd primes N and fixed prime $\log_2(p) \approx 100$. For each N, we average the time taken over 50 runs.

As expected, the sub-algorithm FindBasis (see Algorithm 1) is the bottleneck step of GetIsogeny (see Algorithm 7). In comparison, the other sub-algorithms exhibit practical runtimes. Obtaining a method for finding the basis of spaces X_R and X_S that scales polynomially with N would have a large impact on the practicality of these methods for large N.

In Table 2, we give timings for GetImage (Algorithm 6), which computes the constants E', F', G', H' in the defining equation of the image Kummer surface $\mathcal{K}^{\mathrm{fast}}_{a',b',c',d'}$ of the (N,N)-isogeny φ with kernel generated by N-torsion points R, S, given the degree-N map $\psi : \mathcal{K}^{\mathrm{fast}}_{a,b,c,d} \to \widetilde{\mathcal{K}}$ (i.e., the (N,N)-isogeny before the final scaling).

		Time taken (s)			
N	$\lceil \log_2(p) \rceil$	FindBasis	FindIntersection	Scaling (GE/sqrt for $N \geq 7$)	GetIsogeny
5	104	0.005	0.001	0.000	0.010
7	106	0.021	0.002	0.007/0.012	0.045/0.050
11	95	0.721	0.012	0.206/0.025	1.040/0.859
13	99	4.157	0.025	0.816/0.044	5.238/4.466
17	94	85.267	0.048	7.605/0.059	93.838/86.292
19	105	416.329	0.081	21.109/0.095	439.464/418.450

TABLE 1. Comparison of time taken for different sub-algorithms of GetIsogeny for various odd $N \geq 5$ using both scaling methods for $K = \mathbb{F}_{p^2}$ with $\log_2(p) \approx 100$. We take the average over 50 runs.

N	$\lceil \log_2(p) \rceil$	Time taken for GetImage (s)
7	106	0.004
11	95	0.014
13	99	0.018
17	94	0.030
19	105	0.042

TABLE 2. The time taken for GetImage for various odd $N \geq 7$. We fix the base field $K = \mathbb{F}_{p^2}$ with $90 \leq \log_2(p) \leq 120$, and average the time taken over 50 runs.

6.3. **Performance comparison with previous works.** We now compare the performance of our methods to the software package `AVIsogenies v0.7` [BCR10] for odd prime N. For this comparison, we run the function `IsogenieG2Theta.m`, which computes an isogeny from an abelian variety (where some precompuation is done). It takes as input a kernel in theta coordiantes (i.e., the points $R, S, R+S$ in theta coordinates) and outputs the *theta null point* of the isogeneous Kummer surface. For the fast Kummer surface $\mathcal{K}^{\text{fast}}_{a,b,c,d}$, the theta null point is (a, b, c, d), and thus this function is comparable to running our algorithm GetIsogeny (chosing the best scaling method for each N and p as per Section 6.1) to obtain the (N, N)-isogeny φ and then computing $\varphi((a, b, c, d)) = (a', b', c', d')$.

To compare our software with `AVIsogenies` we therefore run the two algorithms described above for $N = 5, 7, 11, 13, 17$ for a fixed prime p with $90 \leq \log_2(p) \leq 120$, taking the average time taken over 10 runs. The results are given in Table 3. We observe that our methods are comparable and outperform `IsogenieG2Theta.m` for

small $N = 5, 7, 11$. Furthermore, there is the benefit of our method that we recover the explicit isogeny formulæ, as well as the image constants (a', b', c', d'). For larger N, our algorithm is slower, mainly due to the fact that the subroutine FindBasis scales exponentially with N, but we note again that our method also recovers the explicit isogeny formulæ, as well as the image constants (a', b', c', d').

We would like to emphasise that we do not intend timings to be a complete description of the performance of these algorithms; a fair comparison can only be made with precise operation counts. We merely present this broad comparison to showcase the interest in exploring these methods, and highlight important future work in improving the scalability of algorithm FindBasis (see Algorithm 1).

		Time taken (s)	
N	$\lceil \log_2(p) \rceil$	This work	`AVIsogenies`
5	121	0.011	0.022
7	120	0.060	0.242
11	95	0.921	1.402
13	98	4.769	0.126
17	94	95.060	0.226
19	108	423.276	12.752

TABLE 3. The time taken for `IsogenieG2Theta.m` in `AVIsogenies` and GetIsogeny with evaluation to compute the image theta constants (a', b', c', d') of an (N, N)-isogeny for various odd $N \geq 7$. We fix the base field $K = \mathbb{F}_{p^2}$ with $90 \leq \log_2(p) \leq 125$, and average the time taken over 50 runs.

Remark 6.2. Unlike the algorithms presented in this paper, the complexity of the algorithms that we use from `AVIsogenies` depends on N mod 4 (see, for example, [LR15, pg. 199]). This is shown by our experiments in Table 3, which exhbits how $N = 11$ and $N = 19$ perform comparitively worse than $N = 13$ and 17.

REFERENCES

[Bis11] G. Bisson, *Endomorphism rings in cryptography*. PhD thesis. Institut National Polytechnique de Lorraine-INPL; Technische Universiteit Eindhoven, 2011.

[BCR10] G. Bisson, R. Cosset, and D. Robert, *AVIsogenies v0.7 (Abelian Varieties and Isogenies)*, Magma package for explicit isogenies between abelian varieties (2021), `https://www.math.u-bordeaux.fr/~damienrobert/avisogenies/`.

[BCP97] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24.3-4** (1997), 235–265.

[BFT14] N. Bruin, E.V. Flynn and D. Testa, *Descent via* $(3, 3)$*-isogeny on Jacobians of genus* 2 *curves*, Acta Arith. **165** (2014), 201–223.

[CF96] J. W. S. Cassels and E. V. Flynn, *Prolegomena to a middlebrow arithmetic of curves of genus* 2, London Mathematical Society Lecture Note Series, vol. 230, Cambridge University Press, Cambridge, 1996.

[CDS20] W. Castryck, T. Decru, and B. Smith. *Hash functions from superspecial genus-2 curves using Richelot isogenies*, Journal of Mathematical Cryptology **14.1** (2020), pp. 268-292.

[CC86] D. V. Chudnovsky and G. V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests*, Advances in Applied Mathematics **7(4)** (1986), 385–434.

[CCS16] P.N. Chung, C. Costello and B. Smith, *Fast, uniform and compact scalar multiplication for elliptic curves and genus 2 Jacobians with applications to signature schemes*, https://eprint.iacr.org/2015/983 (2015)

[CCS24] M. Corte-Real Santos, C. Costello and B. Smith. *Efficient (3,3)-isogenies on fast Kummer surfaces*, https://arxiv.org/pdf/2402.01223.pdf (2024).

[CF24a] M. Corte-Real Santos and E. V. Flynn. *Github repository*, `https://github.com/mariascrs/NN_isogenies` (2024).

[Cos11] R. Cosset, *Applications of theta functions for hyperelliptic curve cryptography*, Ph.D Thesis, Universit Henri Poincar - Nancy I, November 2011.

[CR15] R. Cosset and D. Robert. *Computing (ℓ, ℓ)-isogenies in polynomial time on Jacobians of genus 2 curves*, Mathematics of Computation **84(294)** (2015), 1953–1975.

[Far88] R. W. Farebrother. *Linear Least Squares Computations*, STATISTICS: Textbooks and Monographs, Marcel Dekker (1988).

[Fly15] E. V. Flynn, *Descent via* $(5, 5)$*-isogeny on Jacobians of genus 2 curves*, J. Number Theory **153** (2015), 270–282.

[FKM24] E. V. Flynn and K. Khuri-Makdisi, *An analog of the Edwards model for Jacobians of genus 2 curves*, Res. Number Theory **10** (2024), no.2, Paper No. 32.

[FT19] E. V. Flynn and Y. B. Ti, *Genus Two Isogeny Cryptography*, PQCrypto **Vol. 11505** Lecture Notes in Computer Science, Springer (2019), 286-306.

[Gau07] P. Gaudry, *Fast Genus 2 arithmetic based on theta functions*, J. Math. Cryptol. **1.3** (2007), 243–265.

[GMP] The GMP Developers. *GMP, the GNU Multiple Precision Arithmetic Library* (Version 6.3.0), `https://gmplib.org/` (2023).

[LR15] D. Lubicz and D. Robert. *Computing separable isogenies in quasi-optimal time*, LMS Journal of Computation and Mathematics **18(1)** (2015), 198–216.

[LR16] D. Lubicz and D. Robert. *Arithmetic on abelian and Kummer varieties*, Finite Fields Their Appl. **39** (2016), 130–158.

[LR22] D. Lubicz and D. Robert. *Fast change of level and applications to isogenies*, ANTS-XV (2022).

[MOV18] A. J. Menezes, P. C. Van Oorschot and S. A. Vanstone, *Handbook of applied cryptography*, CRC press (2018).

[Mon87] P. L. Montgomery, *Speeding the pollard and elliptic curve methods of factorization*, Mathematics of Computation **48(177)** (1987), 243–264.

[Mum83] D. B. Mumford. *Tata lectures on theta. I*, Progress in Mathematics **28** (1983), xiii–235.

[Map24] Maple (2024). Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario.

[Mum84] D. B. Mumford. *Tata lectures on theta. II*, Progress in Mathematics **43** (1984), 243–265.

[RS17] J. Renes and B. Smith. *qDSA: Small and Secure Digital Signatures with Curve-Based Diffie-Hellman Key Pairs*, International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer International Publishing (2017), 273–302.

[Rob10] D. Robert. *Theta functions and cryptographic applications*, PhD thesis. Universit Henri Poincar - Nancy, 2010.

[Rob21] D. Robert. *Efficient algorithms for abelian varieties and their moduli spaces*, Habilitation Diriger des Recherches, Universit de Bordeaux (UB), 2021.

[Sco20] M. Scott, *A note on the calculation of some functions in finite fields: Tricks of the Trade*, https://eprint.iacr.org/2020/1497 (2020)

[Sha73] D. Shanks, *Five number-theoretic algorithms*, Proceedings of the Second Manitoba Conference on Numerical Mathematics (Winnipeg), 1973.

[Ton91] A. Tonelli, *Bemerkung ber die Auflsung quadratischer Congruenzen*, Gttinger Nachrichten (1891), 344–346.

[BZ10] R. P. Brent and P. Zimmermann. *Modern Computer Arithmetic*, Cambridge Monographs on Applied and Computational Mathematics **18**, Cambridge University Press (2011).

University College London, London WC1E 6BT, United Kingdom
E-mail address: maria.santos.20@ucl.ac.uk

Mathematical Institute, University of Oxford, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford OX2 6GG, United Kingdom
E-mail address: flynn@maths.ox.ac.uk