Return of the Kummer: a Toolbox for Genus-2 Cryptography

Maria Corte-Real Santos[1] and Krijn Reijnders[2]

[1] University College London
maria.santos.20@ucl.ac.uk
[2] Radboud University, Nijmegen, The Netherlands
krijn@cs.ru.nl

Abstract. This work expands the machinery we have for isogeny-based cryptography in genus 2 by developing a toolbox of several essential algorithms for *Kummer surfaces*, the dimension-2 analogue of x-only arithmetic on elliptic curves. Kummer surfaces have been suggested in hyperelliptic curve cryptography since at least the 1980s and recently these surfaces have reappeared to efficiently compute $(2,2)$-isogenies. We construct several essential analogues of techniques used in one-dimensional isogeny-based cryptography, such as pairings, deterministic point sampling and point compression and give an overview of $(2,2)$-isogenies on Kummer surfaces. We furthermore show how Scholten's construction can be used to transform isogeny-based cryptography over elliptic curves over $\mathbb{F}_{p^2}$ into protocols over Kummer surfaces over $\mathbb{F}_p$.
As an example of this approach, we demonstrate that SQIsign verification can be performed completely on Kummer surfaces, and, therefore, that one-dimensional SQIsign verification can be viewed as a two-dimensional isogeny between products of elliptic curves. Curiously, the isogeny is then defined over $\mathbb{F}_p$ rather than $\mathbb{F}_{p^2}$. Contrary to expectation, the cost of SQIsign verification using Kummer surfaces does not explode: verification costs only $1.5\times$ more in terms of finite field operations than the SQIsign variant AprèsSQI, optimised for fast verification. Furthermore, it is plausible that arithmetic on Kummer surfaces can be efficiently vectorised, giving Kummer-based protocols over $\mathbb{F}_p$ a potential performance boost on modern architectures, possibly surpassing the performance of elliptic-curve analogues over $\mathbb{F}_{p^2}$.

Keywords: post-quantum cryptography, isogenies, Kummer surface, SQIsign, genus 2

1 Introduction

Post-quantum cryptography aims to develop cryptographic primitives that are secure when the adversary has access to a classical and quantum computer.

Author list in alphabetical order; see https://www.ams.org/profession/leaders/CultureStatement04.pdf. The first author has been supported by UK EPSRC grant EP/S022503/1. Date of this document: 2024-08-14.

Due to the growing investment into quantum computing, this field has garnered a significant amount of attention in the last decade, culminating in the NIST standardisation of the key encapsulation mechanism Kyber [7] and the digital signature schemes Dilithium [31], Falcon [33], and SPHICNS+ [5]. Nevertheless, post-quantum signatures still deserve more attention: we rely mostly on lattice-based security assumptions, and the signature sizes are significantly larger than pre-quanum signatures. Due to this, NIST is still actively seeking new post-quantum secure signature schemes [58].

Isogeny-based cryptography offers an answer to these problems. SQIsign [16, 29, 30] relies on the hardness of the general isogeny problem and boasts the smallest combined signature and public key size of any signature scheme in Round 1 of NIST's alternate call for signatures.

The main disadvantage of isogeny-based primitives is their speed. The signing operation in SQIsign and variants is a few orders of magnitude slower than the lattice-based alternatives, and verification requires at least a few milliseconds. Therefore, there has been a surge of recent research that aims to improve the efficiency of SQIsign. We highlight two schemes in particular. First, SQIsignHD [26], a new scheme that offers much more competitive signing times and improved security reductions, at the cost of verification speed. Second, AprèsSQI [22], a variant of SQIsign optimised for verification speed, with additional trade-offs between verification time and signature size.

Recent works [2, 32, 46] have shown incredible advancements in the performance of higher-dimensional SQIsign, achieving relatively fast signing and verification in a few milliseconds. These approaches verify using a 2-dimensional $(2^n, 2^n)$-isogeny over $\mathbb{F}_{p^2}$ between products of elliptic curves, using the machinery developed after the SIDH attacks [14, 45, 51] and in particular using fast isogeny formulas derived from theta structures [27].

1.1 Our contributions

In this work, we make a step towards having a full-fledged toolbox for isogeny-based cryptography in genus 2. We give an overview of Kummer surfaces (including improvements to crucial maps between different Kummer surface *models*) and give a detailed concrete explanation on how pairings and isogenies of Kummer surfaces work. We then show how to apply these in the context of isogeny-based cryptography by developing several algorithms including CheckOrigin, PointDifference and PointCompression, whose analogues on elliptic curves have existed for years.

We further describe how to exploit Scholten's construction [56] to identify a dimension 2 Kummer surface over $\mathbb{F}_p$ to any elliptic curve over $\mathbb{F}_{p^2}$ that has $\mathbb{F}_{p^2}$-rational 2-torsion. We also detail the extension of this construction due to Costello [24], which depicts how isogenies $\phi : E_1 \to E_2$ of degree 2 between elliptic curves can be associated to $(2,2)$-isogenies $\varphi : \mathcal{K}_1 \to \mathcal{K}_2$ between the corresponding Kummer surfaces. In this way, isogeny-based primitives using supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ can be modified to work with *super-special* Kummer surfaces instead, with all computations now over $\mathbb{F}_p$. Addition-

ally, by restricting to Kummer surfaces that arise from level-2 theta structure, we can potentially exploit vectorisation for the core Kummer arithmetic, shown by Bernstein, Chuengsatiansup, Lange, and Schwabe [4] to be very efficient on modern architectures.

We show that these special $(2,2)$-isogenies between Scholten Kummer surfaces, first described by Costello [24], are a natural restriction of the general $(2,2)$-isogenies defined for these objects. Compared to the $(2,2)$-isogenies derived from theta structures [27], in this work, we use the more geometric interpretation from Costello [24], which views such $(2,2)$-isogenies as morphisms between Jacobians of hyperelliptic curves, and in particular their Kummer surfaces. Fundamentally, our approach relies on theta structures too, but the geometric interpretation using hyperelliptic curves allows us to more naturally develop similar techniques as those used for elliptic curves.

As a showcase of these tools and techniques, we show that the original SQIsign verification [16, 29, 30], or its AprèsSQI variant [22], can also be performed completely over Kummer surfaces defined over $\mathbb{F}_p$. Using Scholten's construction [56], this turns the one-dimensional response isogeny into a two-dimensional isogeny between products over elliptic curves over $\mathbb{F}_p$, instead of $\mathbb{F}_{p^2}$. We analyze the viability and performance of such an approach. At its core, SQIsign verification requires the computation of an isogeny of degree 2^e between supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, where $e \approx 1000$ for NIST Level I security. Using Scholten's construction, we perform this verification on superspecial Kummer surfaces instead. To achieve this, a number of tools need to be developed. Indeed, in dimension 1, we require pairing-based techniques, point compression, and optimised isogeny formulæ, all of which are underdeveloped in the Kummer surface literature[3].

In more detail, we do the following.

- Section 2 describes general Kummer surfaces and squared Kummer surfaces, including those that arise from Scholten's construction and their twists. We explicitly construct maps between these, if they exist, and give improved maps between the squared Kummer and the Jacobian. We also categorise elliptic Kummer surfaces.
- Section 3 describes the use of the generalised Tate pairing to describe the image of isogenies, which allows us to generalise previous results [22, Thm. 2] to the dimension 2 (and higher) case in Theorem 2.
- Section 4 develops essential tools for cryptography on Kummer surfaces, namely CheckOrigin and PointDifference, and apply those to efficiently sample and compress points in PointCompression.
- Sections 5 and 6 contain an overview and improvement to the computation of the fifteen $(2,2)$-isogenies between squared Kummer surfaces, including a new derivation of the three elliptic $(2,2)$-isogenies between elliptic Kummer surfaces described by Costello [24]. We show that the codomain, and its

[3] We choose to do SQIsign on Kummer surfaces "not because it is easy, but because it is hard; because that goal will serve to organise and measure the best of our energies and skills".

Rosenhain invariants, cost[4] at most 11**M** and 32**a** with point evaluation being at most 8**M** and 16**a**. This improves on the state of the art [27] for these specific $(2,2)$-isogenies.
– Section 7 combines all of the above to enable SQIsign verification to be performed on Kummer surfaces for both compressed and uncompressed signatures. We furthermore provide benchmarks in terms of finite-field operations.

Software. Alongside the theory developed in this article, we provide accompanying software, written in `MAGMA` [9], Python and SageMath [59]. Our code is available under the MIT license in the following repository:

`https://github.com/Krijn-math/return-of-the-kummer`

The source code contains the following:

– Optimised Python code that implements the compressed and uncompressed variants of SQIsign verification on Kummer surfaces, used for benchmarking. To implement PointCompression, an algorithm to be performed in signing, we use SageMath for Jacobian arithmetic.
– All the algorithms and maps in this article are implemented in `MAGMA`, with the aim to allow a reader to verify many of the claims made throughout and gain an understanding on how the various objects behave. To this end, we have documented the `MAGMA` code to expose various useful tricks and insights.

1.2 Related work

Kummer surfaces of genus-2 Jacobians were first introduced to cryptography by Chudnovsky and Chudnovsky [18], who gave a variant of Lenstra's ECM factoring algorithm. Gaudry [35] then proposed these Kummer surfaces as a setting for efficient discrete-logarithm-based cryptosystems. Many later works built on this to demonstrate that high-speed, high-security Kummer-based implementations of Diffie–Hellman key exchange [4, 8, 49] and signature schemes [49, 50] give significant improvements over elliptic curves in many contexts. In particular, we highlight a work by Bernstein, Chuengsatiansup, Lange, and Schwabe [4] which develops several new techniques for efficient vectorization of Kummer surface computations, leading to new speed records for high-security constant-time (hyper)elliptic curve Diffie–Hellman. In parallel to this, Lubicz and Robert [43] developed algorithms for efficient arithmetic on Kummer surfaces using the theory of theta functions of level 2. Further, Lubicz and Robert [44] give efficient algorithms for pairing computation, which were later improved on by Robert [52]

More recently, in a work by Costello [24], Kummer surfaces were introduced to isogeny-based cryptography in the context of SIDH. In particular, Costello extended Scholten's construction to transport any chain of 2-isogenies between elliptic curves over $\mathbb{F}_{p^2}$ to a chain of $(2,2)$-isogenies between Kummer surfaces, now defined over $\mathbb{F}_p$. Kummer surfaces are also implicitely used in the work

[4] Using the notation **M** for $\mathbb{F}_p$-multiplications, **S** for squarings and **a** for additions.

by Dartois, Maino, Pope, and Robert [27], who give algorithms to compute $(2,2)$-isogenies between theta structures of level 2. These have since been used to accelerate verification in SQIsign variants. We note that the three $(2,2)$-isogenies described by Costello, which are also rederived in this work, are somewhat special, in the sense that they arise from 2-isogenies between elliptic curves. We therefore call these *elliptic* isogenies. We observe that this subset of isogenies can be computed more efficiently than the general formulæ given in [27].

Very recently, new variants of SQIsign have emerged [2, 32, 46] that use higher-dimensional techniques to achieve fast signing and verification for SQIsign. The breakthroughs are spectacular and shift the focus in SQIsign from one-dimensional to two-dimensional verification. Our work is different in that it transforms the one-dimensional verification of SQIsign into a two-dimensional isogeny. In this way, we demonstrate that one-dimensional SQIsign can itself be viewed as having two-dimesional verification. More generally, the aim of this work is to show that it is possible to transform one-dimensional protocols into two-dimensional protocols, whilst still relying on the one-dimensional hardness assumptions which are arguably better understood. Furthermore, because of the generality of the techniques developed in this work, we believe it can be applied in the context of two-dimensional SQIsign, and our exposition of genus-2 cryptography from the perspective of Kummer surfaces may clarify and complement the description of two-dimensional SQIsign for some readers.

Acknowledgements. We thank Craig Costello for general advice on the technicalities and writing of this paper. We thank Damien Robert for helpful advice on technical details. We thank Lars Ran for helpful advice on the use of Gröbner bases in the derivation of our results. We thank Peter Schwabe for helpful discussions on vectorised implementations of Kummer arithmetic.

2 Kummer Surfaces

This work concerns itself with different models of Kummer surfaces, associated to the same (or isomorphic) hyperelliptic curve $\mathcal{C}$ defined over a field $\Bbbk$. Given a hyperelliptic curve $\mathcal{C}$ of genus 2 with Jacobian $\mathcal{J}_\mathcal{C}$, the corresponding *Kummer surface* is given by the quotient $\mathcal{K} := \mathcal{J}_\mathcal{C}/\langle \pm 1 \rangle$. The Kummer surface has a quartic model in $\mathbb{P}^3$, so that $\mathcal{K}$ can be embedded into projective space with coordinates $(X_1 : X_2 : X_3 : X_4) \in \mathbb{P}^3$. Furthermore, $\mathcal{K}$ has sixteen point singularities, called *nodes*, given by the images of the 2-torsion points of $\mathcal{J}_C$ under this quotient, as these are precisely the points fixed by -1. The quotient map destroys the group law on the Jacboian $\mathcal{J}_C$ and thus $\mathcal{K}$ only inherits scalar multiplication from $\mathcal{J}_C$. However, as we see in Section 2.6, we still have a *pseudo-group* law on $\mathcal{K}$. For example, to add two points $P, Q \in \mathcal{K}$, we require knowledge of $P - Q$.

There are many types of Kummer surface models. For any hyperelliptic curve $\mathcal{C}$, we can construct the *general* Kummer surface $\mathcal{K}_\mathcal{C}^{\text{gen}}$, which we discuss in Section 2.1. When $\mathcal{C}$ is isomorphic over a field $\Bbbk$ to a curve in Rosenhain form then $\mathcal{C}_{\lambda,\mu,\nu}$ also admits a *canonical* and *squared* Kummer surface. These two

other Kummer surfaces are closely related; we describe these in Sections 2.3 and 2.4, and give maps $\mathcal{K}^{\text{gen}}_{\mathcal{C}} \to \mathcal{K}^{\text{gen}}_{\lambda,\mu,\nu}$ and $\mathcal{K}^{\text{gen}}_{\lambda,\mu,\nu} \to \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$. We then introduce and classify squared Kummer surfaces with $\nu = \lambda \cdot \mu$ in Section 2.5. These allow several optimizations beyond regular squared Kummer surfaces and we show their connection to elliptic curves through Scholten's construction in Section 2.8.

In many cases, the literature may call any of these models *the* Kummer surface $\mathcal{K}$. In this work, as we deal with several different isomorphic curves and their associated Kummers, we avoid this and use the following explicit notation. For a general hyperelliptic curve $\mathcal{C}$ of genus 2, with Jacobian $\mathcal{J}_{\mathcal{C}}$, we denote

- the general Kummer surface by $\mathcal{K}^{\text{gen}}_{\mathcal{C}}$,
- the squared Kummer surface by $\mathcal{K}^{\text{Sqr}}_{\mathcal{C}}$, if it exists.

For a curve $\mathcal{C}_{\lambda,\mu,\nu}$ in Rosenhain form, with Jacobian $\mathcal{J}_{\lambda,\mu,\nu}$, we denote

- the associated general Kummer surface by $\mathcal{K}^{\text{gen}}_{\lambda,\mu,\nu}$,
- the associated squared Kummer surface by $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$,
- the associated elliptic Kummer surface by $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\lambda\mu}$, only if $\nu = \lambda\mu$,

For a curve $\mathcal{C}_\alpha$ associated to an elliptic curve E_α through Scholten's construction, with Jacobian $\mathcal{J}_\alpha$, we denote

- the general Kummer surface by $\mathcal{K}^{\text{gen}}_{\alpha}$,
- the squared Kummer surface by $\mathcal{K}_\alpha$, a special form of $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\lambda\mu}$.

Any hyperelliptic curve of genus 2 has five or six x-values w_i where the curve intersects with $y = 0$. These are called the Weierstrass points $(w_i, 0)$. In a degree 5 model, we consider the (single) point at infinity as the Weierstrass point ∞. For a curve in Rosenhain form, the six Weierstrass points are $w_1 = \infty$, $w_2 = 0$, $w_3 = 1$, $w_4 = \lambda$, $w_5 = \mu$ and $w_6 = \nu$. This numbering is strictly and often used throughout this work whenever we work with curves in Rosenhain form, in particular to describe two-torsion.

With our main motivation being Kummer surfaces for use in isogeny-based cryptography, throughout this work we often restrict to *superspecial* Jacobians of genus-2 curves and their associated Kummer surfaces, the natural analogue of supersingular elliptic curves to arbitrary dimension [10, 13, 41]. We refer to a hyperelliptic curve as superspecial when its Jacobian is superspecial. We furthermore require rational 2-torsion on the Jacobians and Kummer surfaces.

2.1 General Kummer surfaces

We begin by discussing the general Kummer surface in more detail. Readers only interested in the cryptographically relevant Kummer surfaces used later in this work can skip directly to Section 2.4.

Construction of Kummer surface. Consider a genus-2 curve $\mathcal{C}$ defined over field $\Bbbk$. We follow Cassels and Flynn [12, §3] to compute the corresponding general Kummer surface. Let

$$\mathcal{C} : y^2 : c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6,\ c_i \in \Bbbk,$$

where c_6 can equal 0 if $\mathcal{C}$ is in the degree 5 model.[5] The Kummer surface $\mathcal{K}^{\text{gen}}$ corresponding to curve $\mathcal{C}$ is defined over $\Bbbk$ and is given by elements $(X_1 : X_2 : X_3 : X_4) \in \mathbb{P}^3$ such that

$$\mathcal{K}^{\text{gen}} : K_2X_4^2 + K_1X_4 + K_0 = 0,$$

where

$$\begin{aligned}
K_2 &:= X_2^2 - 4X_1X_3,\\
K_1 &:= -2\begin{pmatrix} 2c_0X_1^3 + c_1X_1^2X_2 + 2c_2X_1^2X_3 + c_3X_1X_2X_3 \\ +2c_4X_1X_3^2 + c_5X_2X_3^2 + 2c_6X_3^3 \end{pmatrix},\\
K_0 &:= (c_1^2 - 4c_0c_2)X_1^4 - 4c_0c_3X_1^3X_2 - 2c_1c_3X_1^3X_3 - 4c_0c_4X_1^2X_2^2\\
&\quad + 4(c_0c_5 - c_1c_4)X_1^2X_2X_3 + (c_3^2 + 2c_1c_5 - 4c_2c_4 - 4c_0c_6)X_1^2X_3^2\\
&\quad - 4c_0c_5X_1X_2^3 + 4(2c_0c_6 - c_1c_5)X_1X_2^2X_3 + 4(c_1c_6 - c_2c_5)X_1X_2X_3^2\\
&\quad - 2c_3c_5X_1X_3^3 - 4c_0c_6X_2^4 - 4c_1c_6X_2^3X_3 - 4c_2c_6X_2^2X_3^2\\
&\quad - 4c_3c_6X_2X_3^3 + (c_5^2 - 4c_4c_6)X_3^4.
\end{aligned}$$

The identity point $\mathbf{o} \in \mathcal{K}^{\text{gen}}$ is given by $\mathbf{o} = (0 : 0 : 0 : 1)$.

Maps to the General Kummer surface. We can map pairs of points (x_1, y_1), (x_2, y_2) lying on $\mathcal{C}$ to $\mathcal{K}^{\text{gen}}$, where $x_1 \neq x_2$, as follows:

$$\rho : \mathcal{C}^{(2)} \to \mathcal{K}^{\text{gen}}, \quad ((x_1, y_1), (x_2, y_2)) \mapsto (X_1 : X_2 : X_3 : X_4),$$

where

$$X_1 := 1, \quad X_2 := x_1 + x_2, \quad X_3 := x_1x_2, \quad X_4 := \frac{F(x_1, x_2) - 2y_1y_2}{(x_1 - x_2)^2},$$

with

$$\begin{aligned}
F(x_1, x_2) &= 2c_0 + c_1(x_1 + x_2) + 2c_2x_1x_2 + c_3(x_1 + x_2)x_1x_2 + 2c_4(x_1x_2)^2\\
&\quad + c_5(x_1 + x_2)(x_1x_2)^2 + 2c_6(x_1x_2)^3.
\end{aligned}$$

We construct the map $\tilde{\rho} : \mathcal{J}_{\mathcal{C}} \to K^{\text{gen}}$, from the Jacobian $\mathcal{J}_{\mathcal{C}}$ of $\mathcal{C}$ to $\mathcal{K}^{\text{gen}}$, by exploiting the fact that, given a divisor[6] $\langle x^2 + a_1x + a_0, b_1x + b_0\rangle \in \mathcal{J}_{\mathcal{C}}$, we can

[5] Equivalently, called the odd degree model.

[6] In this work, we always use the *Mumford representation* for elements of Jacobians.

construct all the rational functions in the map ρ above in terms of a_0, a_1, b_0, b_1. The map $\tilde{\rho}$ is given as follows.

$$\tilde{\rho} : \langle x^2 + a_1 x + a_0, b_1 x + b_0 \rangle \mapsto (X : Y : Z : T)$$

where

$$X_1 := 1, \quad X_2 := -a_1, \quad X_3 := a_0, \quad X_4 := \frac{F'(a_0, a_1) - 2(b_1^2 a_0 - b_0 b_1 a_1 + b_0^2)}{a_1^2 - 4a_0},$$

with

$$F'(a_0, a_1) = 2c_0 - c_1 a_1 + 2c_2 a_0 - c_3 a_0 a_1 + 2c_4 a_0^2 - c_5 a_1 a_0^2 + 2c_6 a_0^3.$$

We emphasise that this construction of $\mathcal{K}^{\text{gen}}$ associated with $\mathcal{C}$ applies to any hyperelliptic curve $\mathcal{C}$ of genus 2.

Points of order 2 on $\mathcal{K}^{\text{gen}}$. The map $\tilde{\rho} : \mathcal{J} \to \mathcal{K}$ is of order 2 except at the sixteen points in $\mathcal{J}[2]$ which map precisely to the sixteen nodes $\mathcal{K}[2] \subset \mathcal{K}$.

The elements of order 2 in $\mathcal{J}$ are given by divisors $D_{i,j}$, where the index i and j refer to pairs of Weierstrass points $(w_i, 0) + (w_j, 0)$ in the support of $D_{i,j}$ for $1 \leq i < j \leq 6$. The Mumford representation of $D_{i,j}$ is $\langle x^2 - (w_i + w_j)x + w_i \cdot w_j, 0 \rangle$ whenever $w_i, w_j \neq \infty$. Whenever ∞ is a Weierstrass point (i.e., when using the degree 5 model) we consider $w_1 = \infty$ and the Mumford representation of $L_{1,j}$ simply ignores this factor $(x - w_1)$.

Using $\tilde{\rho}$, we find the sixteen points $L_{i,j}$ of order 2 on $\mathcal{K}$ given by

$$\begin{aligned} L_{i,j} &= (1 : w_i + w_j : w_i w_j : F(w_i, w_j)/(w_i - w_j)^2), && \text{when } w_i, w_j \neq \infty, \\ L_{1,j} &= (0 : 1 : w_j : w_j^2), && \text{where } w_1 = \infty. \end{aligned}$$

Addition by points of order 2 on $\mathcal{K}^{\text{gen}}$. Addition of points of order 2 on Kummer surfaces is well-defined, and yields a linear map from $\mathcal{K}^{\text{gen}}$ to itself. For $L_{i,j} \in \mathcal{K}[2]$ of order 2, we can represent the translation by $L_{i,j}$, e.g. $P \mapsto P + L_{i,j}$, as a 4×4 matrix $W_{i,j}$ over $\Bbbk$. As these maps are involutions on $\mathcal{K}^{\text{gen}}$, we get $W_{i,j}^2 = c \cdot I_4$ for some $c \in \Bbbk$. These matrices are computed and described by Cassels and Flynn [12], we provide a compact presentation in Appendix A.

2.2 Rosenhain form of a hyperelliptic curve

To construct the canonical and squared Kummer surfaces, we require a hyperelliptic curves in Rosenhain form $\mathcal{C}_{\lambda,\mu,\nu}$ defined over $\Bbbk$. These Kummer surfaces models are theta structures of level 2, which implies that $\mathcal{K}[2]$ plays an important role in their arithmetic. We will see this in later sections when we give efficient algorithms to compute 2-pairings on and $(2,2)$-isogenies between these Kummer surfaces using their points of order 2.

Definition 1. *A hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$ is in* Rosenhain form *over a field $\Bbbk$, when*

$$\mathcal{C}_{\lambda,\mu,\nu} : y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu) \quad \textit{with } \lambda, \mu, \nu \in \Bbbk\,.$$

The values λ, μ and ν are called the Rosenhain invariants *of $\mathcal{C}_{\lambda,\mu,\nu}$.*

The Rosenhain form of a hyperelliptic curve of genus 2 can be viewed as an analogue to the Montgomery form of elliptic curves. Whereas a general elliptic curve in (short) Weierstrass form admits x-only arithmetic, this x-only arithmetic is much more efficient for curves in Montgomery form. We find a similar situation in genus 2: the general Kummer surface can be constructed for any hyperelliptic curve $\mathcal{C}$, yet high-speed cryptography requires the use of the more efficient Kummer surfaces, which arise from the theory of theta functions.

The original idea to use such Kummer surfaces in cryptography is due to the Chudnovsky brothers [18] in 1986. These more efficient Kummer surfaces come in two forms: the *canonical* Kummer surface, as described by Gaudry [35]; and the closely related *squared* Kummer surface, described by Bernstein [3].

2.3 Canonical Kummer surface

Following Gaudry [35], the canonical Kummer surface associated to a hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$ over $\Bbbk$ in Rosenhain form is defined by four *fundamental theta constants* which can be computed from the Rosenhain invariants of $\mathcal{C}_{\lambda,\mu,\nu}$. Given a hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$ with Rosenhain invariants $\lambda, \mu, \nu \in \Bbbk$, we define the fundamental theta constants $a, b, c, d \in \overline{\Bbbk}$ and *dual fundamental theta constants* $A, B, C, D \in \overline{\Bbbk}$ such that

$$\begin{aligned} A^2 &= a^2 + b^2 + c^2 + d^2, \quad B^2 = a^2 + b^2 - c^2 - d^2, \\ C^2 &= a^2 - b^2 + c^2 - d^2, \quad D^2 = a^2 - b^2 - c^2 + d^2. \end{aligned} \tag{1}$$

The theta constants are related to the Rosenhain invariants in the following way

$$\lambda = \frac{a^2c^2}{b^2d^2},\ \mu = \frac{c^2e^2}{d^2f^2},\ \nu = \frac{a^2e^2}{b^2f^2},$$

where $e, f \in \overline{\Bbbk}$ such that $e^2/f^2 = (AB + CD)/(AB - CD)$. Note that the constants a, b, c, d are defined up to sign, but the resulting Kummer surfaces are isomorphic. The canonical Kummer surface $\mathcal{K}^{\text{can}}_{\lambda,\mu,\nu}$ defined over $\overline{\Bbbk}$ is then given by the following equation.

$$\mathcal{K}^{\text{can}}_{\lambda,\mu,\nu} : \begin{array}{c} T_1^4 + T_2^4 + T_3^4 + T_4^4 + 2E \cdot T_1T_2T_3T_4 \\ = \\ F \cdot (T_1^2T_4^2 + T_2^2T_3^2) + G \cdot (T_1^2T_3^2 + T_2^2T_4^2) + H \cdot (T_1^2T_2^2 + T_3^2T_4^2). \end{array}$$

where

$$E := \frac{(16ABCD)^2 \cdot abcd}{(a^2d^2 - b^2c^2)(a^2c^2 - b^2d^2)(a^2b^2 - c^2d^2)}, \text{ and}$$

$$F := \frac{a^4 - b^4 - c^4 + d^4}{a^2d^2 - b^2c^2}, \quad G := \frac{a^4 - b^4 + c^4 - d^4}{a^2c^2 - b^2d^2}, \quad H := \frac{a^4 + b^4 - c^4 - d^4}{a^2b^2 - c^2d^2}.$$

Note that as the A^2, B^2, C^2, D^2 are linear combinations of a^2, b^2, c^2, d^2, the equation for $\mathcal{K}^{\text{can}}$ is determined entirely by a, b, c, d. The identity point $\mathbf{o} \in \mathcal{K}^{\text{can}}$ is given by $\mathbf{o} = (a : b : c : d) \in \mathcal{K}^{\text{can}}$.

As this article does not use this model of Kummer surface, we defer a discussion of the arithemtic on these surfaces to Gaudry [35].

Points of order 2 on $\mathcal{K}^{\text{can}}_{\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}}$. The 16 points of order 2 on $\mathcal{K}^{\text{can}}$ are

$$\begin{aligned}
\mathbf{o} = &(a : b : c : d), \quad (a : b : -c : -d), \quad (a : -b : c : -d), \quad (a : -b : -c : d),\\
&(b : a : d : c), \quad (b : a : -d : -c), \quad (b : -a : d : -c), \quad (b : -a : -d : c),\\
&(c : d : a : b), \quad (c : d : -a : -b), \quad (c : -d : a : -b), \quad (c : -d : -a : b),\\
&(d : c : b : a), \quad (d : c : -b : -a), \quad (d : -c : b : -a), \quad (d : -c : -b : a).
\end{aligned}$$

For the addition matrices for these points, see Appendix A.2.

2.4 Squared Kummer surface

The squared Kummer surface has been the subject of interest in hyperelliptic curve cryptography [3, 4, 8, 49] as it boasts the fastest arithmetic when one can stay on a single surface.

Construction of the squared Kummer surface. The squared Kummer surface corresponding to $\mathcal{C}_{\lambda,\mu,\nu}/\Bbbk$ is related to the canonical Kummer surface corresponding to $\mathcal{C}_{\lambda,\mu,\nu}/\Bbbk$ via the squaring map $(T_1 : T_2 : T_3 : T_4) \to (T_1^2 : T_2^2 : T_3^2 : T_4^2)$.

The squared Kummer surface $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ is defined by four constants

$$(\mu_1 : \mu_2 : \mu_3 : \mu_4) := (a^2 : b^2 : c^2 : d^2),$$

where now $\mu_i \in \Bbbk$ as long as $\mathcal{C}_{\lambda,\mu,\nu}$ is defined over $\Bbbk$. From this, we obtain a relation between the μ_i and the dual fundamental theta constants as follows

$$\begin{aligned}
A^2 &= \mu_1 + \mu_2 + \mu_3 + \mu_4, \quad B^2 = \mu_1 + \mu_2 - \mu_3 - \mu_4\\
C^2 &= \mu_1 - \mu_2 + \mu_3 - \mu_4, \quad D^2 = \mu_1 - \mu_2 - \mu_3 + \mu_4.
\end{aligned} \tag{2}$$

The equation defining the squared Kummer is given by

$$\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu} : E \cdot X_1X_2X_3X_4 = \left(\begin{array}{l} X_1^2 + X_2^2 + X_3^2 + X_4^2 - F \cdot (X_1X_4 + X_2X_3) \\ -G \cdot (X_1X_2 + X_2X_4) - H \cdot (X_1X_2 + X_3X_4) \end{array} \right)^2.$$

with

$$\begin{aligned}
F &:= \frac{\mu_1^2 - \mu_2^2 - \mu_3^2 + \mu_4^2}{\mu_1\mu_4 - \mu_2\mu_3}, \quad G := \frac{\mu_1^2 - \mu_2^2 + \mu_3^2 - \mu_4^2}{\mu_1\mu_3 - \mu_2\mu_4}, \quad H := \frac{\mu_1^2 + \mu_2^2 - \mu_3^2 - \mu_4^2}{\mu_1\mu_2 - \mu_3\mu_4},\\
E &:= 4\mu_1\mu_2\mu_3\mu_4 \left(\frac{A^2B^2C^2D^2}{(\mu_1\mu_1 - \mu_3\mu_4)(\mu_1\mu_3 - \mu_2\mu_4)(\mu_1\mu_4 - \mu_2\mu_3)} \right)^2.
\end{aligned}$$

The identity point $\mathbf{o} \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ is $\mathbf{o} = (\mu_1 : \mu_2 : \mu_3 : \mu_4)$. As given by Bos, Costello, Hisil, and Lauter [8], we can derive the constants $\mu_1, \mu_2, \mu_3, \mu_4$ from the Rosenhain invariants $\lambda, \mu, \nu \in \Bbbk$ as

$$\mu_4 = 1, \quad \mu_3 = \sqrt{\frac{\lambda\mu}{\nu}}, \quad \mu_2 = \sqrt{\frac{\mu(\mu-1)(\lambda-\nu)}{\nu(\nu-1)(\lambda-\mu)}}, \quad \mu_1 = \mu_2\mu_3\frac{\nu}{\mu}. \tag{3}$$

Mapping to $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda,\mu,\nu}}$. The map $\rho^{\mathrm{Sqr}} : \mathcal{J}_{\lambda,\mu,\nu} \to \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ is given by [19, 23]:

$$D = \langle x^2 + u_1 x + u_0, v_1 x + v_0 \rangle \mapsto (X_1 : X_2 : X_3 : X_4),$$

where

$$\begin{aligned}
X_1 &= \mu_1 \cdot \left(u_0(w_3w_5 - u_0)(w_4 + w_6 + u_1) - v_0^2\right), \\
X_2 &= \mu_2 \cdot \left(u_0(w_4w_6 - u_0)(w_3 + w_5 + u_1) - v_0^2\right), \\
X_3 &= \mu_3 \cdot \left(u_0(w_3w_6 - u_0)(w_4 + w_5 + u_1) - v_0^2\right), \\
X_4 &= \mu_4 \cdot \left(u_0(w_4w_5 - u_0)(w_3 + w_6 + u_1) - v_0^2\right).
\end{aligned} \tag{4}$$

Here, $w_3 = 1$, $w_4 = \lambda$, $w_5 = \mu$, and $w_6 = \nu$ are the Weierstrass points with our fixed numbering.

Points of order 2 on $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda,\mu,\nu}}$. The 16 points in $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}[2]$, which we denote $\mathcal{K}[2]$, correspond precisely to the 16 elements $D_{i,j}$ of order 2 on $\mathcal{J}_{\lambda,\mu,\nu}[2]$. To fill a gap in the literature on this topic, we give a full description of $\mathcal{K}[2]$.

Suppose $\mu_1, \mu_2, \mu_3, \mu_4$ are the theta constants of $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ with Rosenhain invariants $\lambda, \mu, \nu \in \Bbbk$.[7] Let τ and $\tilde{\tau}$ denote the roots of $x^2 - Gx + 1$ (where G is the same as in the defining equation), so that $\tilde{\tau} = 1/\tau$. Let $L_{i,j}$ be the element in $\mathcal{K}[2]$ corresponding to

$$D_{i,j} := \langle (x - w_i)(x - w_j), 0 \rangle \in \mathcal{J}_{\lambda,\mu,\nu}[2].$$

If $w_i = w_1 = \infty$, then $D_{1,j} = \langle (x - w_j), 0 \rangle$, and similarly if $w_j = \infty$. We have

$$\begin{aligned}
\mathbf{o} &= (\mu_1 \colon \mu_2 \colon \mu_3 \colon \mu_4), & L_{1,2} &= (\mu_2 \colon \mu_1 \colon \mu_4 \colon \mu_3), \\
L_{5.6} &= (\mu_3 \colon \mu_4 \colon \mu_1 \colon \mu_2), & L_{3,4} &= (\mu_4 \colon \mu_3 \colon \mu_2 \colon \mu_1), \\
L_{4,5} &= (\mu \cdot \mu_4 \colon \mu_3 \colon 0 \colon 0), & L_{3,6} &= (\mu_3 \colon \mu \cdot \mu_4 \colon 0 \colon 0), \\
L_{4,6} &= (0 \colon 0 \colon \mu \cdot \mu_4 \colon \mu_3), & L_{3,5} &= (0 \colon 0 \colon \mu_3 \colon \mu \cdot \mu_4), \\
L_{2,3} &= ((\nu-1)\mu_2 \colon 0 \colon (\mu-1)\mu_4 \colon 0), & L_{1,4} &= ((\mu-1)\mu_4 \colon 0 \colon (\nu-1)\mu_2 \colon 0), \\
L_{1,3} &= (0 \colon (\nu-1)\mu_2 \colon 0 \colon (\mu-1)\mu_4), & L_{2,4} &= (0 \colon (\mu-1)\mu_4 \colon 0 \colon (\nu-1)\mu_2), \\
L_{2,5} &= (\tau \colon 0 \colon 0 \colon 1), & L_{1,6} &= (1 \colon 0 \colon 0 \colon \tau), \\
L_{2,6} &= (0 \colon 1 \colon \tau \colon 0), & L_{1,5} &= (0 \colon \tau \colon 1 \colon 0).
\end{aligned}$$

[7] Do not confuse the Rosenhain invariant μ with the theta constants μ_i.

Remark 1. We can also write these 2-torsion points in terms of dual constants A, B, C, D rather than Rosenhain invariants. Indeed, following Gaudry in the proof of Lemma 4.2 in [35] and [35, §7.5], we have that

$$\frac{AB+CD}{AB-CD} = \mu \cdot \frac{\mu_4}{\mu_3}, \quad \frac{AC+BD}{AC-BD} = \frac{(\nu-1)\mu_2}{(\mu-1)\mu_4}, \quad \frac{AD+BC}{AD-BC} = \frac{(\mu-\lambda)\mu_2}{(\mu-1)\mu_3}.$$

Addition by points of order 2 on $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda,\mu,\nu}}$. As far as we are aware, the linear maps $W_{i,j}$ that represent addition by a point $L_{i,j}$ of order 2 on the squared Kummer surface do not appear in the literature. We present an approach to derive such matrices, with more details given in Appendix A.2. The resulting matrices $W_{i,j}$ are available in our code.

This approach differs from the usual algebraic approach and a similar approach works for any Kummer surface or, generally, any projective linear map.

Let $D_{i,j} \in \mathcal{J}[2]$ denote the pre-image of $L_{i,j} \in \mathcal{K}[2]$. Our goal is to find a description of $W_{i,j}$. On the Jacobian, it is easy to add $D_{i,j}$ to any random element $D_Q \in J$. Thus, we can take a large enough sequence of points $\mathbf{A}_\mathcal{J} = [D_1, \ldots, D_m]$ and apply this translation to get $\mathbf{B}_\mathcal{J} = [D_1 + D_{i,j}, \ldots, D_m + D_{i,j}]$.

We map both sequences $\mathbf{A}_\mathcal{J}$ and $\mathbf{B}_\mathcal{J}$ down to $\mathcal{K}^{\mathrm{Sqr}}$ to get $\mathbf{A}_\mathcal{K}$ and $\mathbf{B}_\mathcal{K}$. Let A_n denote the n-th element of $\mathbf{A}_\mathcal{K}$, and similarly for B_n. Then $W_{i,j}$ must map A_n to $\lambda_n B_n$ for some $\lambda_n \in \Bbbk$, as points on the Kummer are defined up to a scalar. Hence, we get

$$W_{i,j}\mathbf{A}_\mathcal{K} = \mathbf{B}_\mathcal{K}\Lambda,$$

where Λ denotes the diagonal matrix of size $m \times m$ with λ_n on the diagonal. Assuming $W_{i,j}$ and Λ are unknown, with m large enough, a Gröbner basis computation readily yields a solution for $W_{i,j}$, up to some unknown scaling factor. Given such a solution for $W_{i,j}$, we normalise the top-left corner to 1. By performing this for a few different concrete instantiations of curves $\mathcal{C}$ and primes p, we are able to determine the coefficients of each $W_{i,j}$ in terms of the theta constants μ_i and the Rosenhain invariants λ, μ, ν. We then verify the correctness of the resulting matrices $W_{i,j}$.

2.5 Elliptic Kummer surfaces

In this article, we will work with a special squared Kummer surface which arises from a Rosenhain curve $\mathcal{C}_{\lambda,\mu,\nu}$ with $\nu = \lambda\mu$. Such Kummer surfaces have strong ties to elliptic curves, as we show in Lemma 1. This also becomes apparent when we introduce Scholten's construction in Section 2.8. We therefore call such Kummer surfaces *elliptic*. In this section, we discuss special properties of the elliptic Kummer surface, which are needed to construct our toolbox for Kummer surfaces in Sections 3 and 4.

Lemma 1. *The Jacobian of a curve $\mathcal{C}/\overline{\Bbbk}$ is $(2,2)$-isogenous to a product of elliptic curves $E \times E'$ if and only if $\mathcal{C}$ has Rosenhain invariants satisfying $\nu = \lambda\mu$.*

Proof. Let $\mathcal{J}_{\lambda,\mu,\lambda\mu}$ be the Jacobian of a Rosenhain curve $\mathcal{C}_{\lambda,\mu,\lambda\mu}$ of genus 2. In particular, here $\nu = \lambda\mu$. The quadratic splitting (see [57, § 8.2]) of $\mathcal{C}_{\lambda,\mu,\lambda\mu}$ by $G_1 = x$, $G_2 = (x-1)(x-\nu)$, $G_3 = (x-\lambda)(x-\mu)$ has determinant 0, hence $\mathcal{J}_{\lambda,\mu,\lambda\mu}$ is $(2,2)$-isogenous to a product of elliptic curves. For the other direction, let $\mathcal{J}_{\lambda,\mu,\nu}$ be $(2,2)$-isogenous to a product of elliptic curves. Then, up to some reordering, this Richelot isogeny is given by the splitting $G_1 = x$, $G_2 = (x-1)\cdot(x-w_i)$, $G_3 = (x-w_j)(x-w_k)$ for some assignment of i,j,k to $\{4,5,6\}$. By $\det(G_1,G_2,G_3) = 0$ we get $w_i = w_j \cdot w_k$. Permuting the Rosenhain invariants, this gives $\nu = \lambda\mu$. □

Lemma 1 suggests that the moduli space of elliptic Kummers given by the Igusa-Clebsch invariants [38, p. 620] is a 'nice' subspace of the general moduli space.

Construction of the elliptic Kummer surface $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda,\mu,\lambda\mu}}$. We construct an elliptic Kummer surface $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\lambda\mu}$ in the same way as any squared Kummer surface, depicted in Section 2.4. However, given $\nu = \lambda\cdot\mu$, Equation (3) tells us that $\mu_3 = \mu_4 = 1$, greatly simplifying the equation defining $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\lambda\mu}$. For example, in this case $G = \mu_1 + \mu_2$.

Points of order 2 on $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda,\mu,\lambda\mu}}$. In the case where $\nu = \lambda\mu$, we have that $\mu_3 = \mu_4 = 1$ and $\tau = \frac{(\mu-1)\mu_4}{(\nu-1)\mu_2}$. In this way, twelve of the sixteen two-torsion points simplify as follows:

$$\begin{aligned}
\mathbf{o} &= (\mu_1 : \mu_2 : 1 : 1), & L_{1,2} &= (\mu_2 : \mu_1 : 1 : 1),\\
L_{5,6} &= (1 : 1 : \mu_1 : \mu_2), & L_{3,4} &= (1 : 1 : \mu_2 : \mu_1),\\
L_{4,5} &= (\mu : 1 : 0 : 0), & L_{3,6} &= (1 : \mu : 0 : 0),\\
L_{4,6} &= (0 : 0 : \mu : 1), & L_{3,5} &= (0 : 0 : 1 : \mu),\\
L_{2,3} &= (1 : 0 : \tau : 0), & L_{1,4} &= (\tau : 0 : 1 : 0),\\
L_{1,3} &= (0 : 1 : 0 : \tau), & L_{2,4} &= (0 : \tau : 0 : 1).
\end{aligned}$$

The other points are as before:

$$\begin{aligned}
L_{2,5} &= (\tau : 0 : 0 : 1), & L_{1,6} &= (1 : 0 : 0 : \tau),\\
L_{2,6} &= (0 : 1 : \tau : 0), & L_{1,5} &= (0 : \tau : 1 : 0).
\end{aligned}$$

We highlight the symmetry between the left and right column. This symmetry positively impacts the efficiency of arithmetic on the elliptic Kummer surface.

Addition by points of order 2 on $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda,\mu,\lambda\mu}}$. The matrices that describe addition by points of order 2 on $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\lambda\mu}$ are easily derived giving the matrices $W_{i,j}$ for addition by $L_{i,j}$ on $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, specialised to $\mu_3 = \mu_4 = 1$. Their concrete form is found in Appendix A.2.

2.6 Arithmetic of Squared Kummer surfaces

As with elliptic curves, the construction of the Kummer surface $\mathcal{K}$ as the quotient of $\mathcal{J}$ by ± 1 implies that we only have a *pseudo*-group structure on $\mathcal{K}$. Nevertheless, this is enough to compute scalar multiplications $P \mapsto [n]P$ and differential addition, and in general is rich enough for cryptographic applications. The pseudo-group structure on squared Kummer surfaces looks particularly nice due to surprisingly elegant symmetries.

Basic morphisms. The arithmetic on Kummer surfaces is constructed from four basic morphisms fron $\mathbb{P}_3$ to $\mathbb{P}_3$, namely the squaring map $\mathtt{S}$, the Hadamard involution $\mathtt{H}$, the scaling map $\mathtt{C}_P$, and the inversion map $\mathtt{Inv}$, defined as follows:

$$\begin{aligned}
\mathtt{S}\colon (X:Y:Z:T) &\mapsto (X^2:Y^2:Z^2:T^2),\\
\mathtt{H}\colon (X:Y:Z:T) &\mapsto (X+Y+Z+T:X+Y-Z-T:\\
&\qquad\qquad X-Y+Z-T:X-Y-Z+T),\\
\mathtt{C}_{(P_1:\,P_2:\,P_3:\,P_4)}\colon (X:Y:Z:T) &\mapsto (P_1\cdot X:P_2\cdot Y:P_3\cdot Z:P_4\cdot T),\\
\mathtt{Inv}\colon (X:Y:Z:T) &\mapsto (1/X:1/Y:1/Z:1/T)\\
&\qquad\qquad = (YZT:XZT:XYT:XYZ).
\end{aligned}$$

The map $\mathtt{S}$ costs $4\mathbf{M}$, $\mathtt{H}$ costs $8\mathbf{a}$, $\mathtt{C}_P$ costs $4\mathbf{M}$, and $\mathtt{Inv}$ costs $6\mathbf{M}$.

Doubling, scalar multiplication and other arithmetic operations. The four basic morphisms $\mathtt{S}, \mathtt{H}, \mathtt{C}$ and $\mathtt{Inv}$ are enough to define the curve operations doubling $\mathtt{xDBL} : P \mapsto 2P$, differential addition $\mathtt{xADD} : P, Q, P-Q \mapsto P+Q$, scalar multiplication $\mathtt{xMUL} : P \mapsto [n]P$ (see [19, Appendix A]), and the three-point ladder $P, Q, P-Q, s \mapsto P + sQ$.

2.7 Maps between Kummer surfaces

Maps induced from $\kappa : \mathcal{C} \to \mathcal{C}_{\lambda,\mu,\nu}$. When the Weierstrass points of $\mathcal{C}$ are $\Bbbk$-rational, there is a $\Bbbk$-rational isomorphism $\kappa : \mathcal{C} \to \mathcal{C}_{\lambda,\mu,\nu}$ between a general hyperelliptic curve $\mathcal{C}$ and a curve $\mathcal{C}_{\lambda,\mu,\nu}$ in Rosenhain form [55]. It is given by five values $(a, b, c, d, e) \in \Bbbk$, namely

$$\kappa : \mathcal{C} \to \mathcal{C}_{\lambda,\mu,\nu}, \quad (x, y) \mapsto \left(\frac{ax+b}{cx+d}, \frac{ey}{(cx+d)^2}\right)$$

as described by Costello [24, §2]. This map κ induces a map $\kappa_* : \mathcal{J} \to \mathcal{J}_{\lambda,\mu,\nu}$ between their Jacobians [24, §3]. From κ_*, we construct the induced map $\kappa_{**} : \mathcal{K}^{\text{gen}} \to \mathcal{K}^{\text{gen}}_{\lambda,\mu,\nu}$ between their general Kummer surfaces, given in terms of the values $a, b, c, d, e^2 \in \Bbbk$. For simplicity, we normalise the inputs and outputs to the first coordinate.

$$\kappa_{**} : (1 : X_2 : X_3 : X_4) \mapsto (1 : X_2' : X_3' : X_4')$$

with

$$X_2' = \frac{2acX_3 + (ad+bc)K_2 + 2bd}{c^2X_3 + cdX_2 + d^2}, \quad X_3' = \frac{a^2X_3 + abX_2 + b^2}{c^2X_3 + cdX_2 + d^2}$$

and X_4' computed in terms of the defining polynomials K_1, K_2, K_3 for the domain and K_1', K_2', K_3' for the codomain as

$$X_4' = -\frac{K_1'(1, X_2', X_3') + 4v'}{2(X_2'^2 - 4X_3')}$$

where

$$v' = -\frac{e^2 \cdot (K_1(1, X_2, X_3) + 2X_4K_2(1, X_2, X_3))}{4(c^2X_3 + cdX_2 + d^2)^3}.$$

Map between general and squared Kummer surface. Let $\mathcal{K}^{\text{gen}}_{\lambda,\mu,\nu}$ and $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ be the general and squared Kummer surface associated to the same hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$. The isomorphism between $\mathcal{K}^{\text{gen}}_{\lambda,\mu,\nu}$ and $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ is given by Chung, Costello, and Smith [19] as a linear map $\mathbf{M}$, by interpolating image points under both $\rho^*_{\lambda,\mu,\nu}$ and ρ^{Sqr} of divisors of $D \in \mathcal{J}_{\lambda,\mu,\nu}$.

Map from Kummer to Jacobian. For several applications later in this work (see Sections 3 and 4), we require a (partial) inverse of the map $\rho^{\text{Sqr}} : \mathcal{J}_{\lambda,\mu,\nu} \to \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$, i.e., a map that computes $D, -D \in \mathcal{J}_{\lambda,\mu,\nu}$ given a point $P = (X_1 : X_2 : X_3 : X_4) \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ such that $\rho^{\text{Sqr}}(D) = \rho^{\text{Sqr}}(-D) = P$. Often, recovering the values u_0, u_1 or the value v_0^2 of the Mumford representation is enough. Such maps were originally given by Gaudry [35], making use of additional theta constants and functions. When working on a single Kummer surface, these constants and functions are fixed, and we can easily precompute these. In isogeny-based cryptography, however, we no longer have this luxury, and the computation of these additional constants and functions is rather expensive.

Using algebraic tools, detailed in Appendix C, we find more elegant and efficient maps

$$(\rho^{\text{Sqr}})^{-1} : \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu} \to \mathcal{J}_{\lambda,\mu,\nu}, \qquad P = (X_1 : X_2 : X_3 : X_4) \to \{D, -D\}$$

using three polynomials $F_0, F_1, F_2 \in \Bbbk[X]$ to recover u_0 and u_1, given by

$$\begin{aligned} F_0 =& (w_4 - w_6)\tilde{X}_1 + (w_3 - w_5)\tilde{X}_2 - (w_4 - w_5)\tilde{X}_3 - (w_3 - w_6)\tilde{X}_4, \\ F_1 =& (w_3 - w_5)w_4w_6\tilde{X}_1 + (w_4 - w_6)w_3w_5\tilde{X}_2 \\ & - (w_3 - w_6)w_4w_5\tilde{X}_3 - (w_4 - w_5)w_3w_6\tilde{X}_4, \\ F_2 =& -(\tilde{X}_1 + \tilde{X}_2 - \tilde{X}_3 - \tilde{X}_4)(w_3w_4 - w_5w_6). \end{aligned} \tag{5}$$

We then recover u_0 and u_1 as

$$u_0 = F_1(\tilde{X})/F_0(\tilde{X}), \quad u_1 = F_2(\tilde{X})/F_0(\tilde{X}), \tag{6}$$

where $\tilde{X}_i = X_i/\mu_i$. This recovery works regardless of the projective representation of P, i.e. we recover the same u_0 and u_1 if we consider $P = (\omega X_1 : \omega X_2 : \omega X_3 : \omega X_4)$ for any $\omega \in \Bbbk$ as the image of D under ρ^{Sqr}. In particular, we can apply this to any randomly sampled point on $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$. To recover v_0^2, we require an additional polynomial $G \in \Bbbk[X]$, which allow us to recover ω.

$$\begin{aligned} G &= -(w_3 - w_4)(w_5 - w_6)(w_3 w_4 - w_5 w_6) F_1(\tilde{X}), \\ \omega &= G(\tilde{X})/F_0^2(\tilde{X}). \end{aligned} \tag{7}$$

Together with u_0 and u_1, we recover v_0^2 by computing

$$v_0^2 = u_0(w_3 w_5 - u_0)(w_4 + w_6 + u_1) - \omega \tilde{X}_1. \tag{8}$$

For a number of applications, having u_0, u_1, v_0^2 is already sufficient. However, to recover the points $\{D, -D\}$, we need to compute the corresponding v_0, v_1. To do so, we compute the two roots x_1, x_2 of $x^2 + u_1 x + u_0$ and get the corresponding y-values y_1, y_2 such that $(x_1, \pm y_1)$ and $(x_2, \pm y_2)$ lie on the curve $\mathcal{C}_{\lambda,\mu,\nu}$. Noting that there are two possible y-values for each x_i, we set

$$\widetilde{v}_0 = \frac{(y_1 - x_2)(y_2 - x_2)}{x_2 - x_1}$$

and choose the y-values such that $(\widetilde{v}_0)^2$ matches the v_0^2 computed above. We then compute v_1 as $v_1 = (y_1 - y_2)/(x_1 - x_2)$.

2.8 Scholten's construction

In 2003, Scholten [56] introduced a specific Kummer surface $\mathcal{K}_\alpha$ defined over $\mathbb{F}_p$ associated to a given elliptic curve E_α over $\mathbb{F}_{p^2}$ with rational 2-torsion. This construction provides a tool to translate cryptographic protocols defined between elliptic curves to one between Kummer surfaces, which we will exploit in Section 7. The Kummer surfaces $\mathcal{K}_\alpha$ derived in this construction have the property that they are isomorphic to *elliptic* Kummer surfaces, that is, they can be described as the squared Kummer surfaces of a curve $\mathcal{C}$ with Rosenhain invariants $\lambda, \mu, \lambda\mu$ (see Section 2.5). This subsection describes Scholten's derivation.

As this class of Kummer surfaces will be of most interest to us, from this point forward in the article, we will restrict to $\Bbbk = \mathbb{F}_p$ or $\Bbbk = \mathbb{F}_{p^2}$. For simplicity, we restrict to $p \equiv 3 \bmod 4$ and write $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$ where i is a root of $x^2 + 1 \in \mathbb{F}_p[x]$, though we remark that our results hold more generally than this.

Relation to an elliptic curve. Let $E_\alpha/\mathbb{F}_{p^2} : y^2 = x(x - \alpha)(x - \frac{1}{\alpha})$ be an elliptic curve where $x = x_0 + ix_1$, $y = y_0 + iy_1$, and $\alpha = \alpha_0 + i\alpha_1$. The *Weil restriction* W_α of E_α is an abelian surface defined over $\mathbb{F}_p$ given by

$$W_\alpha := \mathrm{Res}^{\mathbb{F}_{p^2}}_{\mathbb{F}_p}(E_\alpha) = V(W_0, W_1),$$

with W_0 and W_1 given by the *real* and *imaginary* part (respectively) of

$$(y_0 + iy_1)^2 - (x_0 + ix_1)\left((x_0 + ix_1) - (\alpha_0 + i\alpha_1)\right)\left((x_0 + ix_1) - 1/(\alpha_0 + i\alpha_1)\right).$$

Scholten [56] showed that the Weil restriction of $E_\alpha/\mathbb{F}_{p^2}$ is $(2,2)$-isogenous to the Jacobian of a hyperelliptic curve $\mathcal{C}_\alpha/\mathbb{F}_p$ with defining equation given by Costello [24, Prop. 1]. To the Jacobian $\mathcal{J}_\alpha$ of $\mathcal{C}_\alpha$, we can associate a general Kummer surface $\mathcal{K}_\alpha^{\text{gen}}$. As we prefer to work with the squared Kummer model, the next proposition follows Costello [24, §5] to present the isomorphism that maps $\mathcal{C}_\alpha$ to a curve in Rosenhain form. We exploit the fact that, by construction, $\mathcal{C}_\alpha$ has 6 $\mathbb{F}_p$-rational Weierstrass points. We emphasise that for this construction it is necessary that the associated Jacobian $\mathcal{J}_\alpha$ is superspecial [10, 13, 41].

Proposition 1 ([24]). *Consider the hyperelliptic curve $\mathcal{C}_\alpha/\mathbb{F}_p$ of genus 2 with superspecial Jacobian J_α, and let $\beta = \beta_0 + i\beta_1$, $\gamma = \gamma_0 + i\gamma_1 \in \mathbb{F}_{p^2}$ such that $\gamma^2 = \alpha$ and $\beta^2 = (\alpha^2 - 1)/\alpha$. Then $\mathcal{C}_\alpha$ is isomorphic over $\mathbb{F}_p$ to $\mathcal{C}_{\lambda,\mu,\nu}$ where*

$$\lambda = -\frac{(\beta_0\gamma_1 + \beta_1\gamma_0)(\beta_0\gamma_0 + \beta_1\gamma_1)}{(\beta_0\gamma_1 - \beta_1\gamma_0)(\beta_0\gamma_0 - \beta_1\gamma_1)},\ \mu = \frac{(\beta_0\gamma_0 + \beta_1\gamma_1)(\beta_0\gamma_0 - \beta_1\gamma_1)}{(\beta_0\gamma_1 + \beta_1\gamma_0)(\beta_0\gamma_1 - \beta_1\gamma_0)},$$

and $\nu = -(\beta_0\gamma_0 + \beta_1\gamma_1)^2/(\beta_0\gamma_1 - \beta_1\gamma_0)^2$.

One can verify that $\nu = \lambda\mu$ and hence, by Lemma 1, $\mathcal{K}_{\lambda,\mu,\nu}$ is elliptic. By Section 2.7, the isomorphism $\kappa : \mathcal{C}_\alpha \to \mathcal{C}_{\lambda,\mu,\nu}$ will induce an isomorphism $\kappa_{**} : \mathcal{K}_\alpha^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}}$. Composing this with $\mathbf{M} : \mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{Sqr}}$, we obtain a map $\mathcal{K}_\alpha^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{Sqr}}$. The theta constants can be computed using Equation (3) combined with γ_0 and γ_1 as

$$\mu_1 = \sqrt{\lambda} \cdot \left(\frac{\gamma_0^2 - \gamma_1^2}{\gamma_0^2 + \gamma_1^2}\right), \quad \mu_2 = \mu_1/\lambda, \quad \mu_3 = \mu_4 = 1. \tag{9}$$

Mapping points from E_α to $\mathcal{K}_\alpha$. Explicit maps between the Weil restriction of E_α and $\mathcal{J}_\alpha$ are given by Bernstein and Lange [6]. Costello [24, §3] gives this $(2,2)$-isogeny $\eta : W_\alpha(\mathbb{F}_p) \to \mathcal{J}_\alpha(\mathbb{F}_p)$ as a composition of several maps, with the map $E_\alpha(\mathbb{F}_{p^2}) \to W_\alpha(\mathbb{F}_p)$ implicitly assumed. By extending η with the map $\rho^{\text{Sqr}} \circ \kappa_\star$ from Section 2.7 to get $\bar{\eta} := \rho^{\text{Sqr}} \circ \kappa_\star \circ \eta$, we can map points $P \in E_\alpha(\mathbb{F}_{p^2})$ to $\bar{\eta}(P) \in \mathcal{K}_\alpha(\mathbb{F}_p)$. We summarise this in Figure 1.

2.9 Elliptic twists of elliptic Kummer surfaces

We now investigate elliptic Kummer surfaces arising from an elliptic curve E_α and its twist $E_{-\alpha}$. We describe how this influences our choice of map η (i.e., what constant e we choose) from Section 2.7. To the best of our knowledge, the discussion on twists in this section does not appear in previous literature.

Definition 2. *The* elliptic twist *of a squared Kummer surface $\mathcal{K}^{Sqr}$, denoted $\mathcal{K}^T$ is defined as the surface with theta coordinates $(-\mu_1, -\mu_2, \mu_3, \mu_4)$, where μ_i are the theta coordinates of $\mathcal{K}^{Sqr}$.*

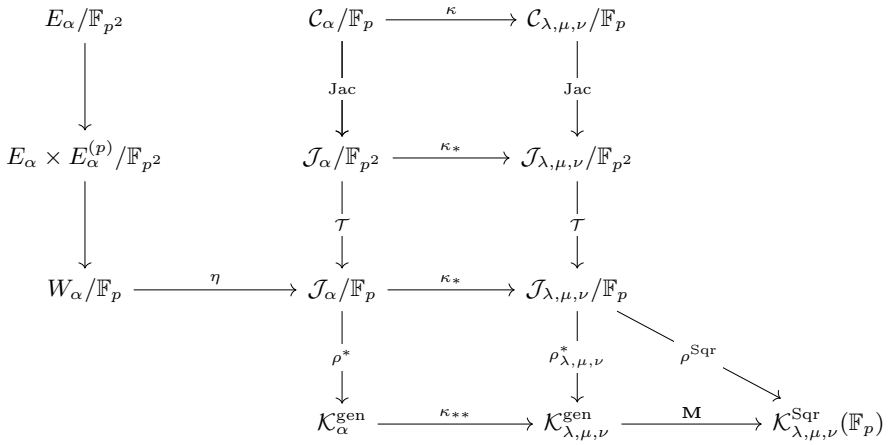


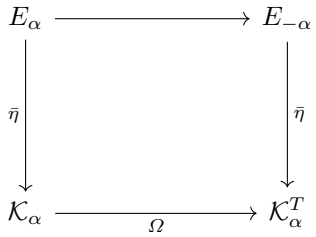
Fig. 1. The maps involved when finding the (squared) Kummer surface defined over $\mathbb{F}_p$ corresponding to an elliptic curve defined over $\mathbb{F}_{p^2}$, including the maps between Kummers described in this section.

We remark that $\mathcal{K}^T$ is always isomorphic to $\mathcal{K}^{\mathrm{Sqr}}$ over $\mathbb{F}_p$ using the isomorphism

$$\Omega : (X : Y : Z : T) \mapsto (-X : -Y : Z : T).$$

The values F, G, H that define $\mathcal{K}^{\mathrm{Sqr}}$ change to $F, -G, H$ for $\mathcal{K}^T$. The name *elliptic twist* is justified by the following lemma.

Lemma 2. *Let $\mathcal{K}_\alpha/\mathbb{F}_p$ be the elliptic Kummer surface associated to $E_\alpha/\mathbb{F}_{p^2}$. Then $\mathcal{K}_\alpha^T/\mathbb{F}_p$ is the elliptic Kummer surface associated to $E_{-\alpha}$, the twist of E_α. In other words, the following square commutes.*



Proof. The twist map $\delta : (x, y) \mapsto (-x, iy)$ maps E_α to $E_{-\alpha}$. We want to show that, for $P_{\pm\alpha} \in E_{\pm\alpha}(\mathbb{F}_{p^2})$,

$$(\rho^{\mathrm{Sqr}} \circ \kappa_* \circ \eta \circ \delta)(P_\alpha) = (\Omega \circ \rho^{\mathrm{Sqr}} \circ \kappa_* \circ \eta)(P_\alpha).$$

Viewing $\eta : E_\alpha(\mathbb{F}_{p^2}) \to \mathcal{J}_\alpha(\mathbb{F}_p)$ as the composition $\mathcal{T} \circ \rho \circ \psi$ (where ψ, ρ, $\mathcal{T}$ are the maps given by Costello [24, pg. 8]), we get that $\eta(P_{-\alpha}) = (\eta \circ \delta)(P_\alpha)$. Therefore, it suffices to show that for divisors $D_{\pm\alpha} \in \mathcal{J}_{\pm\alpha}(\mathbb{F}_p)$ $\Omega \circ \rho^{\mathrm{Sqr}}(D_\alpha) = \rho^{\mathrm{Sqr}}(D_{-\alpha})$.

To show this, we observe that, if $\alpha \mapsto -\alpha$, we have $\beta \mapsto i\beta$ and $\gamma \mapsto i\gamma$. Writing $\beta = \beta_0 + i\beta_1$ and $\gamma = \gamma_0 + i\gamma_1$ we have $\beta_0 \mapsto \beta_1, \beta_1 \mapsto -\beta_0$ (similar for γ_0, γ_1). Looking at the formulæ for the Rosenhain's in terms of α, β, γ in Proposition 1, the Rosenhain invariants λ, μ, ν are unchanged by $\alpha \mapsto -\alpha$.

Therefore, by Equation (9), we get that $\mu_1 \mapsto -\mu_1$ and $\mu_2 \mapsto -\mu_2$. [8] So, $\rho^{\mathrm{Sqr}} : \mathcal{J}_\alpha(\mathbb{F}_p) \to \mathcal{K}_\alpha$ maps a divisor $D_\alpha \in \mathcal{J}_\alpha(\mathbb{F}_p)$ to $(X_1 : X_2 : X_3 : X_4)$ and $\rho^{\mathrm{Sqr}} : \mathcal{J}_{-\alpha}(\mathbb{F}_p) \to \mathcal{K}_{-\alpha}$ maps $D_{-\alpha} \in \mathcal{J}_{-\alpha}(\mathbb{F}_p)$ to $(-X_1 : -X_2 : X_3 : X_4)$. □

Note that the proof for Lemma 2 shows that both $\mathcal{C}_\alpha$ and $\mathcal{C}_{-\alpha}$ map to the same Rosenhain curve $\mathcal{C}_{\lambda,\mu,\nu}$. However, the choice of constants μ_i defining the Kummer surface $\mathcal{K}_\alpha$ not only depends on λ, μ, ν, but also on the concrete values of β_i and γ_i. This choice impacts the design slightly: the map $\kappa : \mathcal{C}_\alpha \to \mathcal{C}_{\lambda,\mu,\nu}$ is given by five values (a, b, c, d, e). The constants a, b, c, d can be computed over $\mathbb{F}_p$, however, e is defined as the square root of $e^2 \in \mathbb{F}_p$, and therefore lies in $\mathbb{F}_p$ only half the time. Whenever $e \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$, the map κ is only defined over $\mathbb{F}_{p^2}$, even though the curves themselves are isomorphic over $\mathbb{F}_p$. As e only determines the y-coordinate of the image, this does not affect the composition map $\eta : E_\alpha \to \mathcal{K}_\alpha$, however it impacts the difficulty of implementation and efficiency of η. Fortunately, the following result shows that we can always avoid this.

Lemma 3. *Denote by $\kappa : \mathcal{C}_\alpha \to \mathcal{C}_{\lambda,\mu,\nu}$ the map given by (a, b, c, d, e) and by $\kappa^T : \mathcal{C}_{-\alpha} \to \mathcal{C}_{\lambda,\mu,\nu}$ the analogous map for the elliptic twist given by $(a^T, b^T, c^T, d^T, e^T)$. Then, either κ or κ^T is defined over $\mathbb{F}_p$. In other words, $e \in \mathbb{F}_p$ if and only if $e^T \notin \mathbb{F}_p$, and vice versa.*

Proof. As both (a, b, c, d) and (a^T, b^T, c^T, d^T) are always defined over $\mathbb{F}_p$ by definition, we only need to show that e^2 and $(e^T)^2$ differ by a non-square in $\mathbb{F}_p$. Direct computation shows that

$$e^2 = -\left(\frac{\gamma_0}{\gamma_1}\right)^6 \cdot {e^T}^2,$$

where $-\left(\frac{\gamma_0}{\gamma_1}\right)^6$ is a non-square. [9] This ensures that precisely one of e^2 or ${e^T}^2$ has a square root over $\mathbb{F}_p$, which proves that statement. □

Lemma 3 allows us to always choose η defined over $\mathbb{F}_p$ by taking it to be $E_\alpha \to \mathcal{K}_\alpha$ or the equivalent map $E_\alpha \to E_{-\alpha} \to \mathcal{K}_\alpha^T \to \mathcal{K}_\alpha$.

2.10 Scholten's construction is gluing

Let $\mathcal{J}_\alpha$ be as in Section 2.5. The constructed $\mathcal{J}_\alpha$ is $(2, 2)$-isogenous *over* $\mathbb{F}_{p^2}$ to the product of elliptic curves $E_\alpha \times E_\alpha^{(p)}$ (where $E_\alpha^{(p)}$ is the Frobenius conjugate), say

$$\varphi_\alpha : \mathcal{J}_\alpha \to E_\alpha \times E_\alpha^{(p)}.$$

[8] For a Magma version of this proof, see `TwistProof.m`.

[9] See `TwistProof.m` for a proof of this in `MAGMA`.

Such an isogeny is often called a *splitting* and its dual is the *gluing* of E_α and $E_\alpha^{(p)}$ over some two-torsion. Note that the projections $E_\alpha \times E_\alpha^{(p)} \to E_\alpha$ and $E_\alpha \times E_\alpha^{(p)} \to E_\alpha^{(p)}$ are not defined over $\mathbb{F}_p$ (otherwise this would have implications on the quantum security).

Lemma 4. *Let $\alpha = \alpha_0 + \alpha_1 i \in \mathbb{F}_{p^2}$ with $\alpha_0, \alpha_1 \neq 0$. Then the Weil restriction W_α is $(2,2)$-isogenous over $\mathbb{F}_p$ to some $\mathcal{J}_\alpha/\mathbb{F}_p$. For the right gluing, $E_\alpha \times E_\alpha^{(p)}$ is $(2,2)$-isogenous to the extension of this $\mathcal{J}_\alpha$ over $\mathbb{F}_{p^2}$.*

We summarise this observation in Figure 2.

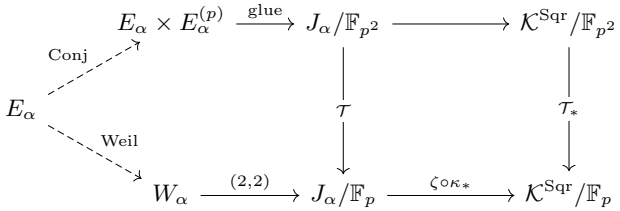


Fig. 2. A clearer picture of the Weil Restriction maps

3 Using pairings on Kummer surfaces

Pairings on abelian varieties have proven to be essential in the construction and cryptanalysis of many cryptographic primitives [1, 36, 39]. Most relevant to this article is their use in isogeny-based cryptography, in particular recent work [22, 25, 42] that shows how the degree 2 Tate pairing can be used to deterministically sample specific 2^n-torsion points on elliptic curves. The aim of this section is to generalise this result to Kummer surfaces in order to enable efficient point compression. To this end, we introduce the general theory to describe the image of isogenies using pairings in Section 3.1, we apply this to Jacobians in genus 2 in Section 3.2 to generalise [22, Thm. 2] to Kummer surfaces. We then show how to concretely compute such pairings of degree 2 in Section 3.4.

We remark that, though our target application is compression of SQIsign signatures, the possible applications of generalised Tate pairings to study the image of isogenies spread much wider than SQIsign, or even Kummer surfaces[10].

[10] Even pre-quantum elliptic curve subgroup membership testing [40] can be rewritten in this language.

3.1 Describing the image of isogenies using pairings

In a step towards generalising the elliptic curve techniques for deterministic point sampling to Kummer surfaces, we describe a general method to decompose $A/\operatorname{Im}\hat{\varphi}$ in terms of $\ker\varphi$ for any isogeny $\varphi : A \to B$ between abelian varieties. A similar interpretation of the Tate pairing was independently given by Robert [54]. In this work, we apply such techniques in concrete cases and decompose $[2]\mathcal{J}_{\mathcal{C}}$, resp. $[2]\mathcal{K}_{\mathcal{C}}$ using similar methods as used before to describe $[2]E$. Such techniques allows us to sample points with improved precision in $\mathcal{J}_C \setminus [2]\mathcal{J}_{\mathcal{C}}$, and in general this technique has proven useful for isogeny-based cryptography.

In dimension 1. Before stating the general theorem, let us recall the following classical result in dimension 1 on the image under doubling $[2]E$.

Theorem 1 (Thm. 4.1, [37]). *Let $E/k : y^2 = (x-\lambda_1)(x-\lambda_2)(x-\lambda_3)$ be an elliptic curve. Then $P \in [2]E$ if and only if*

$$(x-\lambda_1), (x-\lambda_2) \text{ and } (x-\lambda_3) \text{ are all squares.}$$

This can be rephrased and specialised in terms of reduced Tate pairings [22, Thm. 2]: A point $P \in E$ is in $[2]E$ if and only if the reduced Tate pairing with all three 2-torsion points $(\lambda_i, 0)$ is trivial. Furthermore, points $P \in E \setminus [2]E$ lie above $L_i = (\lambda_i, 0)$ if $t_2(L_i, P) = 1$ and $t_2(L_j, P) = -1$ for $j \neq i$.

General Theorem. The above result is a particular instantiation of a much more general result on *generalised Tate pairings*, associated to any isogeny $\varphi : A \to B$ between abelian varieties. We first sketch this general framework, and detail how the dimension 1 example is a specific case, before applying it in the dimension 2 setting.

As shown by Bruin [11], to any separable isogeny $\varphi : A \to B$ between abelian varieties over $\mathbb{F}_q$ (q a prime power) such that the kernel of $\hat{\varphi}$ is annihilated by $[q-1]$, we can associate a perfect pairing

$$\ker\varphi \times \operatorname{coker}\hat{\varphi} \to \mathbb{F}_q^*.$$

We refer to the above pairing as the *generalised φ-Tate pairing*. This pairing has been studied in the context of cryptography [15] for elliptic curves.

As Robert [54] notes, perfectness implies that we can precisely identify $\operatorname{Im}\hat{\varphi}$ using $\ker\varphi$ and this pairing, in the following sense.

Lemma 5 (Cor. 5.2, [54]). *Let t_φ denote the (reduced) φ-Tate pairing. Then*

$$Q \in \operatorname{Im}\hat{\varphi} \quad \Leftrightarrow \quad t_\varphi(P, Q) = 1 \text{ for all } P \in \ker\varphi.$$

Beyond this, we can decompose the cosets of $\operatorname{coker}\hat{\varphi} = A/\operatorname{Im}\hat{\varphi}$ using the *profile* of a point $Q \in A$.

Definition 3. *Let t_φ denote the (reduced) φ-Tate pairing and let $Q \in A$. The* profile *of Q is the array of evaluations of $t_\varphi(P, Q)$ in all points $P \in \ker \varphi$. We denote the profile of Q for the reduced φ-Tate pairing by $t_{\ker \varphi}(Q)$.*

By Lemma 5, the generalised φ-Tate pairing is constant on each coset, so the profile of a point Q determines precisely in which coset of $A/\operatorname{Im} \hat{\varphi}$ it lies.

By bilinearity of the Tate pairing, the profile of $Q + Q'$ is the pointwise multiplication of their profiles. Furthermore, any basis of $\ker \varphi$ is enough to determine the full profile, and we therefore use the smaller array of evaluations $(t_\varphi(P_i, Q))_i$ for P_i a basis of $\ker \varphi$.

When we take $\varphi = [2]$ to be the multiplication-by-2 map on an elliptic curve E, we recover the dimension 1 example given in the previous section. If, instead, we take $\varphi : E \to E/\langle L_i \rangle$ be the isogeny with kernel generated by L_i (as defined in the paragraph following Theorem 1), we have that $\operatorname{coker} \hat{\varphi}$ consists precisely of two cosets: $\mathcal{O} + [2]E$ and $L_i + [2]E$. Then $t_\varphi(L_i, P) = 1$ if and only if P is above L_i, for $P \notin [2]E$. As $t_{\ker \varphi}$ gives information only with respect to the smaller set $\ker \varphi$ about the coarser cosets $A/\operatorname{Im} \hat{\varphi}$, we see that $t_{\ker \varphi}$ gives a subset of information of $t_{\ker[\deg \varphi]}$. However, this information is given with fewer computations, and may in some settings give enough information.

As an example, the technique, used in SIDH/SIKE, CSIDH and SQIsign, to sample points on Montgomery curves whose order is divisible by 2^f by sampling a non-square x-coordinate x_P can be rephrased as sampling points in the coset $E \setminus \operatorname{Im} \hat{\varphi}$, where $\varphi : E \to E/\langle (0,0) \rangle$, given that the reduced Tate pairing $t_\varphi((0,0), P)$ is exactly the Legendre symbol of x_P.

The cokernel as a group. The group structure of $\operatorname{coker} \varphi$ for a separable d-isogeny φ with d prime and kernel of dimension n is the same as that of the kernel: both are isomorphic to μ_d^n. Assuming μ_d is rational, both are isomorphic to $(\mathbb{Z}/d\mathbb{Z})^n$ by some choice of primitive root $\zeta_d \in \mathbb{F}_q$. Using $t_{\ker \varphi}$, this result becomes intuitive.

Lemma 6. *For principally polarised abelian varieties A, B over $\mathbb{F}_q$, let $\varphi : A \to B$ be a separable d-isogeny with d prime and rational kernel generators, such that the generalised Tate pairing t_φ is perfect. Then*

$$\ker \varphi \xrightarrow{\sim} \operatorname{coker} \hat{\varphi}(\mathbb{F}_q) \xrightarrow{\sim} \mu_d^n,$$

where μ_d are the d-th roots of unity in $\mathbb{F}_q$.

Proof. Per definition of φ, its kernel is a $\mathbb{Z}$-module of rank n. Let $K_1, \ldots, K_n \in A(\mathbb{F}_q)$ be a basis of $\ker \varphi$. The map $t_{\ker \varphi} : \operatorname{coker} \hat{\varphi}(\mathbb{F}_q) \to \mu_d^n$ given by

$$t_{\ker \varphi} : P \mapsto (t_\varphi(K_1, P), \ldots, t_\varphi(K_g, P)$$

gives a surjective map $A(\mathbb{F}_q) \to \mu_d^n$, with kernel $\operatorname{Im} \hat{\varphi}(\mathbb{F}_q)$ [54, Cor. 5.2]. Hence, the map induces an isomorphism $A(\mathbb{F}_q)/\operatorname{Im} \hat{\varphi}(\mathbb{F}_q) = \operatorname{coker} \hat{\varphi}(\mathbb{F}_q) \xrightarrow{\sim} \mu_d^n$. □

This result is used in some cases in pairing-based cryptography, where the Tate pairing of level n can sometimes be viewed as a pairing $E[n] \times E[n] \to \mathbb{F}_q^*$, and in general is useful in practical applications of profiles of generalised Tate pairings, as we see in next sections.

3.2 Decomposing $\mathcal{J}_\mathcal{C}$ in dimension 2 using profiles

This section uses the Tate pairing for the multiplication-by-2 map $[2]$ on $\mathcal{J}_\mathcal{C}$ to decompose $\mathcal{J}_\mathcal{C}$ into cosets $\mathcal{J}_\mathcal{C}/[2]\mathcal{J}_\mathcal{C}$, following the general theory developed in Section 3.1. In contrast to elliptic curves, for our dimension 2 setting we require this decomposition to identify the right coset, rather than simply identifying $\mathcal{J}_\mathcal{C} \setminus [2]\mathcal{J}_\mathcal{C}$ (that is, all but the trivial coset). For the rest of this work, we assume Jacobians whose complete 2-torsion is rational, so that Lemma 6 applies over the base field.

We demonstrate how [22, Thm. 2] can be generalised to dimension 2 (or more generally to any dimension). This aligns with a description of $[2]\mathcal{J}_\mathcal{C}$ sketched by Cassels and Flynn [12, Ch. 10]. The core idea is to identify the coset of Q in $\mathcal{J}_\mathcal{C}/[2]\mathcal{J}_\mathcal{C}$ using the *profile* $t_{\ker[2]}$ (see Definition 3) of Q. As we are able to compute these pairing values on $\mathcal{K}_\mathcal{C}$ too, both for the general and squared models, we get an analogous result for Kummer surfaces.

Theorem 2. *Let $P \in \mathcal{K}_\mathcal{C}$ and let $\{L_{i,j}\}_{1\le i<j\le 6}$ denote the fifteen points of order 2. Then $P \in [2]\mathcal{K}_\mathcal{C}$ if and only if $t_{\ker[2]}(P)$ is trivial, e.g.*

$$t_2(L_{i,j}, P) = 1 \quad \textit{for all } 1 \le i < j \le 6.$$

Proof. This follows from Lemma 5, or more directly, the non-degeneracy of the Tate pairing implies $P \in [2]\mathcal{K}_\mathcal{C}$ if and only if $t_2(K, P) = 1$ for any $K \in \mathcal{K}_\mathcal{C}[2]$. □

By bilinearity of the Tate pairing, given a basis $B_1, \ldots, B_4$ for $\mathcal{K}_\mathcal{C}[2]$ and writing $L_{i,j} = a \cdot B_1 + b \cdot B_2 + c \cdot B_3 + d \cdot B_4$ for $a, b, c, d \in \{0, 1\}$, we can compute $t_2(L_{i,j}, P)$ in terms of the four Tate pairings $t_2(B_i, P)$ as

$$t_2(L_{i,j}, P) = t_2(B_1, P)^a \cdot t_2(B_2, P)^b \cdot t_2(B_3, P)^c \cdot t_2(B_4, P)^d.$$

and thus Theorem 2 can more succintly be given as follows.

Corollary 1. *Let $P \in \mathcal{K}_\mathcal{C}$ and let $\{B_1, B_2, B_3, B_4\}$ be a basis for $\mathcal{K}_\mathcal{C}[2]$. Then $P \in [2]\mathcal{K}_\mathcal{C}$ if and only if $t_2(B_i, P) = 1$ for all $1 \le i \le 4$.*

Hence, we can quickly identify points in $\mathcal{K}_\mathcal{C} \setminus [2]\mathcal{K}_\mathcal{C}$, by sampling a random point P until $t_2(B_i, P) = -1$ for some $1 \le i \le 4$.

Remark 2. Let f be such that 2^f is the maximal power-of-two torsion on $\mathcal{K}_\mathcal{C}(\mathbb{F}_p)$. It is tempting to think that $P \in \mathcal{K}_\mathcal{C} \setminus [2]\mathcal{K}_\mathcal{C}$ if and only if the order of P is divisible by 2^f. However, this implication only works in one direction: Any P with order divisible by 2^f cannot be in $[2]\mathcal{K}_\mathcal{C}(\mathbb{F}_p)$ as this contradicts the maximality of f. However, points in $\mathcal{K}_\mathcal{C} \setminus [2]\mathcal{K}_\mathcal{C}$ do not necessarily have an order divisible by 2^f.

3.3 Decomposing Jacobians isogenous to Weil restrictions

Consider the superspecial Jacobian $\mathcal{J}_\alpha$ that is $(2,2)$-isogenous to the Weil restriction of a supersingular E_α, and thus, has Rosenhain invariants λ, μ, ν such that $\lambda \cdot \mu = \nu$. Let $m = \frac{p+1}{2}$ and let 2^f be the largest power of 2 dividing m. Then, as a group, $\mathcal{J}_\alpha$ is isomorphic to

$$\mathbb{Z}_m \times \mathbb{Z}_m \times \mathbb{Z}_2 \times \mathbb{Z}_2.$$

In the following paragraphs, we show how profiles $t_{\ker[2]}(P)$ identify points $P \in \mathbb{Z}_m \times \mathbb{Z}_m$ whose order is divisible by 2^f, e.g. have maximal power-of-two torsion. This is in general useful for isogeny-based cryptography, and we will rely heavily on this fact in later sections.

Decomposition into cosets. Let $\mathcal{B} = (P_1, P_2, P_3, P_4)$ be a basis for $\mathcal{J}_\mathcal{C}$ such that P_1 and P_2 generate the subgroup $\mathbb{Z}_m \times \mathbb{Z}_m$, and P_3 and P_4 generates $\mathbb{Z}_2 \times \mathbb{Z}_2$. Theorem 2 gives us that $P \in [2]\mathcal{J}_\mathcal{C}$ if and only if $t_{\ker[2]}(P)$ is trivial. We find that $\mathcal{J}_\mathcal{C}/[2]\mathcal{J}_\mathcal{C}$ decomposes into 16 cosets

$$aP_1 + bP_2 + cP_3 + dP_4 + [2]\mathcal{J}_\mathcal{C}, \quad a, b, c, d \in \{0, 1\}$$

with the trivial coset $[2]\mathcal{J}_\mathcal{C}$ given by $a = b = c = d = 0$, and so the group structure of $\mathcal{J}_\mathcal{C}/[2]\mathcal{J}_\mathcal{C}$ is isomorphic to $(\mathbb{Z}/2\mathbb{Z})^4$. As the profile of points in a coset is constant given the basis $\mathcal{B}$, we can write $t_{\mathcal{B},[2]}(a, b, c, d)$ for the profile associated to the coset $aP_1 + bP_2 + cP_3 + dP_4 + [2]\mathcal{J}_\mathcal{C}$. We compute $t_{\mathcal{B},[2]}(a, b, c, d)$ simply as $t_{\ker[2]}(Q)$ for $Q = aP_1 + bP_2 + cP_3 + dP_4$, given $\mathcal{B} = (P_1, \ldots, P_4)$. By bilinearity of the Tate pairing, addition of these profiles is well-defined.

The three cosets, $P_3 + [2]\mathcal{J}_\mathcal{C}$, $P_4 + [2]\mathcal{J}_\mathcal{C}$ and $P_3 + P_4 + [2]\mathcal{J}_\mathcal{C}$ contain precisely all points of $\mathcal{J}_\mathcal{C}$ whose order is divisible by 2^{f-1}, besides the trivial coset. Hence, such points have profile $t_{\mathcal{B},[2]}(0, 0, c, d)$ with $c, d \in \{0, 1\}$.

All 12 other cosets therefore contain precisely all points whose order is divisible by 2^f. In particular, the points in $\mathbb{Z}_m \times \mathbb{Z}_m$ whose order is divisble by 2^f are given precisely by the cosets $P_1 + [2]\mathcal{J}_\mathcal{C}$, $P_2 + [2]\mathcal{J}_\mathcal{C}$ and $P_1 + P_2 + [2]\mathcal{J}_\mathcal{C}$. That is, they are identified by profiles $t_{\mathcal{B},[2]}(a, b, 0, 0)$ with $a, b \in \{0, 1\}$. Summarizing, we get the following theorem, improving on Theorem 2.

Theorem 3. *Let $P \in \mathcal{J}_\mathcal{C}$. Let $\mathcal{B} = (P_1, P_2, P_3, P_4)$ be a basis of $\mathcal{J}_\mathcal{C}$ as given above. Let $t_{\mathcal{B},[2]}(a, b, c, d) := t_{\ker[2]}(aP_1 + bP_2 + cP_3 + dP_4)$. Then we get*

$$P \in [2]\mathcal{J}_\mathcal{C} \quad \Leftrightarrow \quad t_{\ker[2]}(P) = t_{\mathcal{B},[2]}(0, 0, 0, 0),$$

and

$$2^f \nmid \operatorname{ord}(P) \quad \Leftrightarrow \quad t_{\ker[2]}(P) = t_{\mathcal{B},[2]}(0, 0, c, d),$$

and

$$2^f \mid \operatorname{ord}(P) \quad \Leftrightarrow \quad t_{\ker[2]}(P) \neq t_{\mathcal{B},[2]}(0, 0, c, d).$$

Furthermore, $P \in \langle P_1, P_2 \rangle$ with order divisible by 2^f if and only if

$$t_{\ker[2]}(P) = t_{\mathcal{B},[2]}(a, b, 0, 0), \quad \text{with not both } a, b = 0.$$

The above theorem can similarly be adapted to the general Tate pairing φ to decompose $\operatorname{coker} \varphi$ using profiles.

3.4 Computing pairings of degree 2 in dimension 2

The general theory to compute pairings on Jacobians of hyperelliptic curves is well-developed and a good overview is given by Galbraith, Hess, and Vercauteren [34]. In this section, we compute pairings of degree 2 on Jacobians and Kummers.

On Jacobians. Let $D_{i,j} = (w_i, 0) + (w_j, 0)$ be an element of $\mathcal{J}_\mathcal{C}$ of order 2 i.e., the divisor on $\mathcal{J}_\mathcal{C}$ with support $\{(w_i, 0), (w_j, 0)\}$ where w_j are the Weierstrass values, and $D_P = (x_1, y_1) + (x_2, y_2)$ any element of $\mathcal{J}_\mathcal{C}(\mathbb{F}_p)$. Whenever $D_{i,j}$ and D_P are coprime (i.e., have disjoint support), the Tate pairing $\langle D_{i,j}, D_P\rangle_2$ is computed as

$$T_2(D_{i,j}, D_P) = \langle D_{i,j}, D_P\rangle_2 = (w_i - x_1)(w_i - x_2)(w_j - x_1)(w_j - x_2). \quad (10)$$

Using resultants, we can express this computation in terms of the Mumford representations of $D_{i,j}$ and D_P, ensuring computations can stay over the base field. The *reduced* Tate pairing $t_2(D_{i,j}, D_P)$ is then defined as

$$t_2(D_{i,j}, D_P) = \langle D_{i,j}, D_P\rangle_2^{\frac{p^k-1}{2}},$$

where k is the embedding degree. Note that, when $k = 1$, this coincides with the Legendre symbol of the Tate pairing. When $D_{i,j}$ and D_P are not coprime, we take any random element $S \in \mathcal{J}_\mathcal{C}$ such that $D_{i,j}$ is coprime with both S and $D_P + S$. which allows us to compute $t_2(D_{i,j}, D_P)$ as

$$t_2(D_{i,j}, D_P) = t_2(D_{i,j}, D_P + S)/t_2(D_{i,j}, S).$$

On general Kummers. As in Section 2.1, let $\mathcal{K}^{\text{gen}}[2] \setminus \{\mathbf{o}\} = \{L_{i,j}\}_{1\le i<j\le 6}$ with pre-images $D_{i,j} = (w_i, 0) + (w_j, 0) \in \mathcal{J}_\mathcal{C}$, where $(w_i, 0) \in \mathcal{C}$ are the six Weierstrass points. Then $L_{i,j} = (1 : l_2^{(i,j)} : l_3^{(i,j)} : l_4^{(i,j)}) \in \mathcal{K}^{\text{gen}}$ with

$$l_2^{(i,j)} = w_i + w_j, \quad l_3^{(i,j)} = w_i \cdot w_j,$$

and where $l_4^{(i,j)}$ can be derived from l_2 and l_3. We can rewrite the Tate pairing $t_2(D_{i,j}, D_P)$ for $i, j \neq 1$ in terms of Kummer coordinates $L_{i,j} = (1 : l_2 : l_3 : l_4)$ and any other coprime Kummer point $P = (1 : k_2 : k_3 : k_4)$ as

$$\langle L_{i,j}, P\rangle_2 = l_3^2 + k_3^2 - (k_2 l_2 l_3 + k_2 k_3 l_2) + k_2^2 l_3 + k_3 l_2^2 - 2 \cdot k_3 l_3. \quad (11)$$

Whenever $i = 1$, the point $L_{1,j}$ has the form $(0 : 1 : w_j : w_j^2)$ and one can compute similar formulas for $t_{i,j}$[11]. Whenever $L_{i,j}$ and P are not co-prime, we add some element $L_{i',j'} \in \mathcal{K}^{\text{gen}}[2]$ to P to obtain the required co-primality.[12]

[11] Alternatively, one can find suitable $L_{i',j'}$ to compute $t_{i,j}(P)$ as $t_2(L_{i,j} + L_{i',j'}, P)/t_2(L_{i',j'}, P)$, where the above formula can be applied as long as both $L_{i,j} + L_{i',j'}$ and $L_{i',j'}$ are of the required form.

[12] As discussed throughout Section 2, the action of a 2-torsion point $L_{i,j}$ on $\mathcal{K}$ is well-defined and given by a matrix $W_{i,j}$.

On Squared Kummers. On squared Kummers, there are three distinct methods to compute the degree-2 Tate pairing: (a) follow a similar method to that used for general Kummer surfaces; (b) follow the monodromy approach due to Robert [52]; and (c) map the Kummer points to partial Jacobian elements and compute the pairing on the Jacobian. We will write $T_{i,j}(P)$ or $t_{i,j}(P)$ to refer to the unreduced, resp. reduced, Tate pairing of degree 2 between $L_{i,j}$ and $P \in \mathcal{K}^{\mathrm{Sqr}}$.

(a) using squared Kummer coordinates. The approach used for general Kummers also works on the squared Kummer surface. Due to the more difficult map from $\mathcal{J}_{\lambda,\mu,\nu}$ to $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, we require the use of additional theta functions, beyond the four coordinate functions X_1, X_2, X_3, X_4, to compute such pairings. Specifically, we require the coordinate functions X_8 and X_{10}, as given by Gaudry [35], to compute all possible Tate pairings of degree 2. However, due to the need for additional theta functions, the above approach is neither efficient nor elegant.

(b) monodromy approach by Robert. An alternative approach to compute such a Tate pairing on Kummer surfaces is sketched by Lubicz and Robert [44]. To compute $t_{i,j}(Q)$, we require the matrix $W_{i,j}$ representing the translation-by-$L_{i,j}$ map. For elliptic Kummers, these are given in Appendix A.2. These computations are explained more generally by Robert [52]; we apply their Algorithm 5.2 to Kummer surfaces of genus-2 hyperelliptic curves for a fixed pairing $T_{i,j}$. See Appendix B for a detailed explanation.

(c) using the efficient maps to recover u_0 and u_1. The last method uses the efficient map to recover u_0 and u_1 for a point P as given by Equation (6). The values u_0 and u_1 allow us to compute the left-side of the Mumford representation $a(x) = x^2 + u_1 x + u_0$. All pairings $t_{i,j}(P)$ can then be computed as the resultant of $a(x)$ with $(x - w_i)(x - w_j)$. This has the additional advantage that many values can be re-used to compute multiple pairings $t_{i,j}(P)$ for the same P, which we heavily rely on in later sections to compute the profile $t_{\ker[2]}(P)$.

4 Algorithms for Kummer-based cryptography

The aim of this section is to introduce several tools required to transport isogeny-based cryptography to Kummer surfaces. Throughout this section, the tools we develop are largely motivated by our showcasing example: compressed SQIsign verification between Kummer surfaces. To develop an efficient point compression algorithm in Section 4.3, we require the following tools.

- For many of our algorithms, we often assume that two points $P, Q \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ both arise either from the Jacobian $\mathcal{J}_{\lambda,\mu,\nu}$ or from its twist. We can check this by ensuring $\mathsf{CheckOrigin}(\mathcal{K}, \mathbf{o}, P) = \mathsf{CheckOrigin}(\mathcal{K}, \mathbf{o}, Q)$.
- To compress the point K defining our $(2^f, 2^f)$-isogeny in Section 5, we need to write K as $P + [s]Q$ for some deterministically sampled $P, Q \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}[2^f]$ and some scalar $s \in \{1, \ldots, 2^f\}$. In Section 4.3, we develop an algorithm to deterministically sample P, Q using *pairings* on Kummer surfaces.

– To decompress P, Q, s and compute the kernel generator K we require a three point ladder – `3ptLadder` – which takes as input $P, Q, P - Q$ and s. The point difference, $P - Q$, will not be known a priori, and so must be computed using PointDifference.

We emphasise that these algorithms are more widely applicable beyond the scope of this work and make a step towards making isogeny-based cryptography using Kummer surfaces practical.

4.1 Identifying twist points

The rational points on the Kummer surface $\mathcal{K}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ consist of the points originally coming from $\mathcal{J}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ as well as from its twist $\mathcal{J}^T_{\lambda,\mu,\nu}(\mathbb{F}_p)$[13]. In other words, the elements

$$D = \langle x^2 + u_1 x + u_0, i \cdot (v_1 x + v_0) \rangle \in \mathcal{J}_{\lambda,\mu,\nu}(\mathbb{F}_{p^2})$$

also map to $\mathbb{F}_p$-rational points $P \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}(\mathbb{F}_p)$.

Recognizing if a random point $P \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ is an image point of the original Jacobian $\mathcal{J}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ or its twist is essential for many steps in later algorithms. For example, the algorithm PointDifference, which computes $P \pm Q$ given $P, Q \in \mathcal{K}$, requires both P and Q to originate from the same Jacobian in order to return rational points $P \pm Q \in \mathcal{K}(\mathbb{F}_p)$.

The map $(\rho^{\text{Sqr}})^{-1} : \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu} \to \mathcal{J}_{\lambda,\mu,\nu}$ described by polynomials F_0, F_1, F_2 over $\mathbb{F}_p$ (see Equations (5) and (6)) gives us the criteria we need to recognise the origin of a points on $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$, as summarised by the following lemma.

Lemma 7. *Let $\mathcal{K}^{Sqr}_{\lambda,\mu,\nu}$ be a squared Kummer surface with $\mathbf{o} = (\mu_1 : \mu_2 : \mu_3 : \mu_4)$, arising from a hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$ with Weierstrass points $(w_i, 0)$. A point $P = (X_1 : X_2 : X_3 : X_4) \in \mathcal{K}^{Sqr}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ originates from an element $D \in \mathcal{J}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ if and only if the following value z is a square in $\mathbb{F}_p$:*

$$z = u_0(w_3 w_4 - u_0)(w_4 + w_6 + u_1) - \omega X_1/\mu_1 \in \mathbb{F}_p$$

with u_0, u_1 as given in Equation (6), and ω in Equation (7). Otherwise, P originates from an element $D \in \mathcal{J}^T_{\lambda,\mu,\nu}(\mathbb{F}_p)$.

Proof. Direct computation shows that the values u_0 and u_1 are such that the pre-image $D_P \in \mathcal{J}(\mathbb{F}_{p^2})$ of P has Mumford representation $\langle x^2 + u_1 x, +u_0, - \rangle$. As D_P maps to $P \in \mathcal{K}(\mathbb{F}_p)$, using Equation (4) we find that u_0, u_1 and v_0^2 must be rational. Hence, either $D_P \in \mathcal{J}(\mathbb{F}_p)$, with $v_0 \in \mathbb{F}_p$, or $D_P \in \mathcal{J}^T(\mathbb{F}_p)$, with $v_0 = i \cdot \alpha$ for some $\alpha \in \mathbb{F}_p$.

As $z = v_0^2$ (see Equation (8)), when $D_P \in \mathcal{J}(\mathbb{F}_p)$, this is therefore a square in $\mathbb{F}_p$. Conversely, if $D_P \in \mathcal{J}^T(\mathbb{F}_p)$, we find $z = (i\alpha)^2 = -\alpha^2$, which is not a square for $p \equiv 3 \mod 4$. □

[13] This is a different twist than the *elliptic twist* defined in Section 2.9. The elliptic twist $\mathcal{K}^T$ originates from twisting the elliptic curve E_α, whereas this twist originates from a twist of the hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$.

We use this lemma to construct the algorithm CheckOrigin. Given a point $P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, it will output `true` if $P \in \rho^{\mathrm{Sqr}}(\mathcal{J}_{\lambda,\mu,\nu}(\mathbb{F}_p))$ or `false` if $P \in \rho^{\mathrm{Sqr}}(\mathcal{J}^T_{\lambda,\mu,\nu}(\mathbb{F}_p))$.

Algorithm 1 CheckOrigin.

Input: A Kummer surface $\mathcal{K} = \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ with zero $\mathbf{o} = (\mu_1 : \mu_2 : \mu_3 : \mu_4)$ and a point $P = (X_1 : X_2 : X_3 : X_4) \in \mathcal{K}(\mathbb{F}_p)$

Output: `true` if $P \in \rho^{\mathrm{Sqr}}(\mathcal{J}_{\lambda,\mu,\nu}(\mathbb{F}_p))$ or `false` if $P \in \rho^{\mathrm{Sqr}}(\mathcal{J}^T_{\lambda,\mu,\nu}(\mathbb{F}_p))$.

1: $\widetilde{X} \leftarrow \mathtt{C}_{\mathtt{Inv}(\mathbf{o})}(P)$
2: Compute $u_0 \leftarrow F_1(\widetilde{X})/F_0(\widetilde{X})$ and $u_1 \leftarrow F_2(\widetilde{X})/F_0(\widetilde{X})$.
3: Compute $\omega \leftarrow G(\widetilde{X})/F_0^2(\widetilde{X})$ and $v_0^2 \leftarrow u_0(w_3w_5 - u_0)(w_4 + w_6 + u_1) - \omega\tilde{X}_1$.
4: **return** `IsSquare` (v_0^2)

Remark 3. CheckOrigin is much simpler than the general maps given by theta functions, described by Gaudry [35], to compute the origin. Previous works [8] using the general maps would avoid computing the origin as the computations are too involved, and would simply work their way around such problems. The above algorithm may therefore also improve, for example, the twist security of hyperelliptic curve Diffie–Hellman-style protocols.

4.2 Difference of points

Though we want to do arithmetic on the Kummer surface $\mathcal{K}$ for efficiency, we only have a pseudo-group law. In particular, to compute $P+Q \in \mathcal{K}$ or $P+[s]Q \in \mathcal{K}$, we require the knowledge of P, Q *and* $P - Q \in \mathcal{K}$. For many applications, however, we will only have access to $P, Q \in \mathcal{K}$ a priori. To overcome this, we develop an algorithm to compute $P \pm Q \in \mathcal{K}$, given $P, Q \in K$, defined only up to sign.

Let $S = P + Q$ and $D = P - Q$, given on the Kummer surface as $S = (S_1 : S_2 : S_3 : S_4)$ and $D = (D_1 : D_2 : D_3 : D_4)$. We follow the approach by Renes and Smith [50, Prop. 4], using the biquadratic forms B_{ij} associated to the Kummer surface with the property

$$S_i \cdot D_j + D_i \cdot S_j = \lambda_{ij} B_{ij}(P, Q),$$

i.e., they are equal up to some projective constant $\lambda_{ij} \in \mathbb{F}_p$. Writing B_{ij} for $B_{ij}(P, Q)$, we get six degree-2 forms $f_{i,j}$ in variables x_1, x_2, x_3, and x_4 for $1 \leq i < j \leq 4$ given by

$$f_{i,j}(x_i, x_j) = B_{jj} \cdot x_i^2 - 2B_{ij}x_ix_j + B_{ii}x_j^2,$$

such that $f_{i,j}(X_i, X_j) = 0$ if and only if $R = (X_1 : X_2 : X_3 : X_4)$ corresponds to either S or D. Without loss of generality, we set $X_1 = 1$ and solve each subsequent equation to determine a solution $R = (X_1 : X_2 : X_3 : X_4)$ to this system of equations corresponding to either S or D. In this way, we can deterministically find $R = P \pm Q$ given P and Q. In practice, we compute the greatest common

divisor of two polynomials of degree-2 to derive the shared root between e.g. $f_{1,3}$ and $f_{2,3}$, which corresponds to the root X_3. In our implementation, we specialise the inversion-free Euclid algorithm from [20] to polynomials of degree-2 to get a precise cost for such a computation. The resulting algorithm PointDifference is optimised to reduce finite field arithmetic.

Algorithm 2 PointDifference

Input: A Kummer surface $\mathcal{K}$ with two points $P, Q \in \mathcal{K}$.
Output: A deterministic point $R = (X_1 : X_2 : X_3 : X_4) \in \mathcal{K}$, with $R = P \pm Q$.

for $i, j \in [1..4]$ with $i \leq j$ **do**
  $A_{i,j} \leftarrow B^{\mathcal{K}}_{i,j}(P, Q)$
for $1 \leq i, j \leq 4$ with $i < j$ **do**
  $f_{i,j} \leftarrow A_{j,j}x_i^2 - 2A_{i,j}x_ix_j + A_{i,i}x_j^2$
Write $f_{1,2}(1, x_2)$ as $(x_2 - \alpha_1)(x_2 - \alpha_2)$ with $\alpha_1 > \alpha_2$.
Write $\gcd(f_{1,3}(1, x_3), f_{2,3}(\alpha_2, x_3))$ as $(x_3 - \alpha_3)$
Write $\gcd(f_{1,4}(1, x_4), f_{2,4}(\alpha_2, x_4))$ as $(x_4 - \alpha_4)$
return $R = (1 : \alpha_2 : \alpha_3 : \alpha_4)$

It is possible to return both S and D (without knowing which one is which) by using the other root α_1 of $f_{1,2}(1, x_2)$. The extra cost for this is only two extra gcd computations. This version of the algorithm is used in later sections, which we show by setting a flag `both` to `true`.

On the Squared Kummer surface. The above approach works well on the canonical Kummer surface $\mathcal{K}^{\text{can}}$, as the bilinear forms are symmetric and efficiently computable for this surface. However, as explained well by Renes and Smith [50], this is not the case on $\mathcal{K}^{\text{Sqr}}$ which has costly equations for B_{ij}. Nevertheless, there exists another Kummer surface model, called the *Intermediate* Kummer surface $\mathcal{K}^{\text{Int}}$, which has elegant bilinear forms and is related to $\mathcal{K}^{\text{Sqr}}$ via a Hadamard map $\mathcal{K}^{\text{Sqr}} \xrightarrow{\mathtt{H}} \mathcal{K}^{\text{Int}}$. Therefore, rather than working with the inefficient biquadratics of $\mathcal{K}^{\text{Sqr}}$ we simply use the efficient isomorphisms

$$\mathcal{K}^{\text{Sqr}} \xrightarrow{\mathtt{H}} \mathcal{K}^{\text{Int}} \xrightarrow{\mathtt{H}} \mathcal{K}^{\text{Sqr}}.$$

That is, we map P and Q to $\mathtt{H}(P)$ and $\mathtt{H}(Q)$ on $\mathcal{K}^{\text{Int}}$, compute the point difference R of $\mathtt{H}(P)$ and $\mathtt{H}(Q)$ using PointDifference on $\mathcal{K}^{\text{Int}}$, and map back $R \mapsto \mathtt{H}(R) \in \mathcal{K}^{\text{Sqr}}$ to find the required $\mathtt{H}(R) = P \pm Q$.

4.3 Point sampling with certain profile

A well-known trick [25] to sample points on Montgomery curves $E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x$ with all available 2^f-torsion, is to sample random non-square $x \in \mathbb{F}_p$ until $x^3 + Ax^2 + x$ is square. This is an application of Theorem 1, as x being non-square is equivalent to $t_2((0, 0), (x, -)) = -1$, cleverly using the well-chosen

Weierstrass point $(0,0)$ on E_A to make the Tate pairing efficient and, more importantly, independent from E_A[14].

In a similar way, the profiling technique from Section 3 can be used to sample random points on Kummer surfaces with specific profiles and can be applied to efficiently compress a point K of order 2^f on elliptic Kummer surfaces, as we can deterministically sample points P, Q such that their pre-images D_P and D_Q span the $\mathbb{Z}/2^f \times \mathbb{Z}/2^f$ subgroup containing the pre-image of K. By applying PointCompression to P, Q and K, we find a scalar s such that $K = P + [s]Q$.

Naive Point Sampling. We first sample P as a deterministic random element of $\mathcal{K}^{\mathrm{Sqr}}$, and compute the profile $t_{\ker[2]}(P)$ until $t_{\ker[2]}(P)$ is as predescribed. We then similarly sample Q deterministically random until $t_{\ker[2]}(Q)$ is as predescribed and, additionally, different from $t_{\ker[2]}(P)$. We then multiply by the right cofactor to ensure both P and Q have order 2^f.

Elegant Point Sampling. As a generic element D_P of the Jacobian $\mathcal{J}_\mathcal{C}$ can be obtained by two *curve* points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ on $\mathcal{C}(\mathbb{F}_p)$, we can sample deterministic quasi-random points on $\mathcal{K}$ by sampling random points P_i on $\mathcal{C}$, and mapping them to $\mathcal{K}$. We fix $P_2 = (w_i, 0)$ to be a Weierstrass point of $\mathcal{C}$, so that we get quasi-random elements $D_P = (P_1) + (P_2) \in \mathcal{J}_\mathcal{C}$ with Mumford representation $D_P = \langle (x - x_1)(x - w_i), - \rangle$. The profile $t_{\ker[2]}(D_P) = (t_2(D_{k,m}, D_P))_{1 \leq k,m \leq 6}$ is completely determined by products of the Legendre symbols of $(x_1 - w_j)$ and $(w_i - w_j)$ for $1 \leq j \leq 6$. By Equation (10),

$$T_2(D_{k,m}, D_P) = (w_i - w_k)(w_i - w_m)(x_1 - w_k)(x_1 - w_m). \tag{12}$$

Adapting the main theorem of Ohashi [47], we derive the quadratic residues of $w_i - w_j$ for $\mathcal{C}_{\lambda,\mu,\nu}$ as in Scholten's construction.

Lemma 8. *Let $\mathcal{C}_{\lambda,\mu,\nu}$ be a superspecial hyperelliptic curve of genus 2 with Rosenhain invariants λ, μ, ν, associated to a supersingular elliptic curve E_α through Scholten's construction. Then,*

$$\lambda,\ 1-\mu,\ 1-\nu,\ \lambda-\mu,\ \mu-\nu \text{ are squares in } \mathbb{F}_p,$$

and

$$\mu,\ \nu,\ 1-\lambda,\ \nu-\lambda \text{ are non-squares in } \mathbb{F}_p.$$

Proof. First, $\lambda = \mu_1/\mu_2$ is a square and, per technical details of Scholten's construction, the quadratic reciprocity of μ is the opposite of λ. This proves both μ and $\nu = \lambda\mu$ are non-squares. The quadratic reciprocities of $1 - \mu$, $1 - \nu$ and $\lambda - \mu$ are identical through their relation to the theta constants (see Section 2.5). By direct computation, $\lambda - \mu$ has the same quadratic reciprocity as $(\beta_0^2 + \beta_1^2) \cdot (\gamma_0^2 + \gamma_1^2) = n(\beta)n(\gamma)$, and γ, β are squares [24, Lemma 1]. For

[14] This allows precomputation of the quadratic residues of a pre-determined set of x-values, which allows particularly efficient sampling of 2^f-torsion points.

$1-\lambda$, observe by [35, §7.5] that the quadratic reciprocity of $\lambda - 1$ is that of $(\nu-1)\cdot(\mu-1)$ and thus $1-\lambda$ is non-square. Then, combining these results, we find that $\mu - \nu = \mu \cdot (1-\lambda)$ is square, and $\nu - \lambda = -\lambda \cdot (1-\mu)$ non-square. □

Efficient point sampling. By fixing $P_2 = (w_i, 0)$ for some index i, we know the quadratic residues of $(w_i - w_j)$ *a priori*. Thus, to sample a point with the right profile, we do the following:

1. Sample a random $x_1 \in \mathbb{F}_p$ until $x_1(x_1-1)(x_1-\lambda)(x_1-\mu)(x_1-\nu)$ is square i.e. $P_1 = (x_1, -) \in \mathcal{C}_{\lambda,\mu,\nu}$.
2. Compute the Legendre symbols of x_1, $x_1 - 1$, $x_1 - \mu$ and $x_1 - \nu$.
3. Use these values to determine the profile of D_P using Equation (12). Start over if the profile is not as predescribed.
4. Map D_P to $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ to obtain a point $P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ of pre-described profile.

We can go one step further: instead of sampling a random $x_1 \in \mathbb{F}_p$, for the fixed system parameter p, we precompute a list $\mathtt{L}_{b_0,b_1}$ of small $\mathbb{F}_p$-values x, where x has a Legendre symbol $b_0 \in \{1,-1\}$ and $x-1$ a Legendre symbol $b_1 \in \{1,-1\}$. Thus, Step 3 only requires the computation of the Legendre symbol of $x-\mu$ and $x-\nu$ to derive the full profile $t_{\ker[2]}(P)$.

The value $t_2(D_P, D_{2,3})$ is the Legendre symbol of $x_1(x_1-1)w_i(w_i-1)$ and thus the specific element $t_{2,3}(P)$ of the profile $t_{\ker[2]}(P)$ is completely precomputable by precomputing the sets $\mathtt{L}_{b_0,b_1}$. Thus, to find a point with a given profile $T = (t'_{i,j})_{1\le i<j\le 6}$, we can find the associated list $\mathtt{L}_{b_0,b_1}$ that matches up to $t'_{2,3} = t_{2,3}(P)$. We then go through the $x \in \mathtt{L}_{b_0,b_1}$ until we find $P_1 = (x,y) \in \mathcal{C}_{\lambda,\mu,\nu}$, compute the Legendre symbols of $x-\mu$ and $x-\nu$, and derive $t_{\ker[2]}(D_P)$. If $t_{\ker[2]}(D_P) = T$, we map $D_P \mapsto P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$. This computes a point on $\mathcal{K}$ using one square root (to get y) and two Legendre symbols, plus the multiplications required to map to $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ using Equation (4). With no loss in performance, we fix $P_2 = (w_4, 0)$. This is summarised in Algorithm 3, where $\mathtt{DeriveProfile}_{\mathtt{L}}$ determines $t_{\ker[2]}(P)$ using Equation (12) given the Legendre symbols of $x-\mu$ and $x-\nu$ for $x \in \mathtt{L}_{b_0,b_1}$.

4.4 Point Compression

We now describe how to perform point compression for points on Kummer surfaces. More precisely, we show how to compress a point K of order 2^f to a scalar $s \in \mathbb{Z}/2^f\mathbb{Z}$, where $P + [s]Q$ and $P, Q \in \mathcal{K}[2^f]$ are deterministically sampled points. This is useful in cryptographic settings where K is sent over a public channel, as we can send s instead of K thus reducing the communication cost. As long as the receiver deterministically arrives at the same points P, Q, they can recompute the same $K = P + [s]Q$ given only s. With our target application in mind, we will restrict to elliptic Kummer surfaces $\mathcal{K}_\alpha$, though the theory is more widely applicable.

Using Section 4.3, we may assume two elements $D_P, D_Q \in \mathcal{J}_{\mathcal{C}}[N]$ that span a subgroup $\mathbb{Z}_N \times \mathbb{Z}_N$ with $N = 2^f$ and $P, Q \in \mathcal{K}_{\mathcal{C}}[N]$ their images. We want to

Algorithm 3 Improved point sampling on squared Kummer surfaces

Input: A squared Kummer surface $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, a target profile T, and a precomputed list of $\mathbb{F}_p$-values $\mathtt{L} = [x_1, x_2, \ldots]$ such that $t_{2,3}(D_P) = T_{2,3}$ if $(x_i, -) \in \mathcal{C}_{\lambda,\mu,\nu}(\mathbb{F}_p)$.
Output: A point $P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ with profile $t_{\ker[2]}(P) = T$.

for $x \in \mathtt{L}$ **do**
  $\mathtt{RHS} \leftarrow x(x-1)(x-\lambda)(x-\mu)(x-\nu)$
  if $\mathtt{IsSquare}(\mathtt{RHS})$ **then**
    $y \leftarrow \mathtt{Sqrt}(\mathtt{RHS})$, $\gamma_1 \leftarrow \mathtt{IsSquare}(x-\mu)$, $\gamma_2 \leftarrow \mathtt{IsSquare}(x-\nu)$
    $t_{\ker[2]}(D_P) \leftarrow \mathtt{DeriveProfile}_{\mathtt{L}}(\gamma_1, \gamma_2)$
    if $t_{\ker[2]}(D_P) = T$ **then**
      $u_0 \leftarrow x \cdot w_4$, $u_1 \leftarrow -(x + w_4)$, $v_0^2 \leftarrow (w_4/(x-w_4))^2 \cdot \mathtt{RHS}$
      $X_1 \leftarrow \mu_1 \cdot \big(u_0(w_3w_5 - u_0)(w_4 + w_6 + u_1) - v_0^2\big)$
      $X_2 \leftarrow \mu_2 \cdot \big(u_0(w_4w_6 - u_0)(w_3 + w_5 + u_1) - v_0^2\big)$
      $X_3 \leftarrow \mu_3 \cdot \big(u_0(w_3w_6 - u_0)(w_4 + w_5 + u_1) - v_0^2\big)$
      $X_4 \leftarrow \mu_4 \cdot \big(u_0(w_4w_5 - u_0)(w_3 + w_6 + u_1) - v_0^2\big)$
      return $P = (X_1 : X_2 : X_3 : X_4)$

compress a given point $K \in \mathcal{K}^{\mathrm{Sqr}}$, whose pre-image on $\mathcal{J}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ is in the subgroup $\langle D_P, D_Q \rangle$. In general, this is feasible as long as we can solve a discrete logarithm in base N, which is simple in our specific case $N = 2^f$. Thus, after finding preimages for the points P, Q and K on $\mathcal{J}_{\mathcal{C}}$, we can solve the discrete logarithm on $\mathcal{J}_{\mathcal{C}}$ by adapting algorithms from [48] to obtain s. We then compute both $D, S = P \pm Q, P \mp Q$ and recompute both $K_D = \mathtt{3ptLadder}(P, Q, D, s)$ and $K_S = \mathtt{3ptLadder}(P, Q, S, s)$. One of K_D and K_S will then equal K. The compression is thus the bit s plus an additional bit to indicate the use of D or S.

Algorithm 4 PointCompression

Input: A Kummer surface $\mathcal{K}$, deterministically generated points P, Q of order N and a point K of order N such that $D_K \in \langle D_P, D_Q \rangle \subset \mathcal{J}_{\mathcal{C}}$.
Output: A value $s \in [1..N]$ and $b \in \{0, 1\}$ such that $K = \mathtt{3ptLadder}(P, Q, D_b, s)$.

$D_P \leftarrow \rho^{-\mathrm{Sqr}}(P)$, $D_Q \leftarrow \rho^{-\mathrm{Sqr}}(Q)$, $D_K \leftarrow \rho^{-\mathrm{Sqr}}(K)$.
$s \leftarrow \mathtt{DiscLog}(K, P, Q)$
$D_0, D_1 \leftarrow$ PointDifference$(P, Q, \mathtt{both} = \mathtt{true})$
$K_0 \leftarrow \mathtt{3ptLadder}(P, Q, D_0, s)$
if $K_0 = K$ **then**
  return s, 0
return s, 1

We remark that to use PointCompression, we need to know that $D_K \in \langle D_P, D_Q \rangle$ before running the algorithm. The results from Section 3 allow us to do this as long as we can compute the pairing t_N. Decompression requires the value s and a single bit to determine whether to use D or S in $\mathtt{3ptLadder}$. We then recompute K by deterministically sampling P, Q, recomputing $P - Q$ as either D or S, and deriving $K = \mathtt{3ptLadder}(P, Q, P - Q, s)$, as shown in algorithm PointDecompression.

Algorithm 5 PointDecompression

Input: A Kummer surface $\mathcal{K}$, deterministically generated points P, Q of order N, a scalar s and bit $b \in \{0, 1\}$.

Output: A point K of order N on $\mathcal{K}$.

1: $D_0, D_1 \leftarrow$ PointDifference$(P, Q, \texttt{both} = \texttt{true})$

2: **return** $\texttt{3ptLadder}(P, Q, D_b, s)$

Remark 4. Using the elegant sampling method, we essentially find elements $D_P, D_Q \in \mathcal{J}_\mathcal{C}$ before we map these down to $P, Q \in \mathcal{K}^{\text{Sqr}}$. To compress $K \in \mathcal{K}^{\text{Sqr}}$, we therefore do not need to find pre-images D_P and D_Q, as we already have these from this approach to sampling points.

5 (2, 2)-isogenies on Kummer surfaces

This work aims to showcase Kummer surfaces as objects that can lead to practical isogeny-based cryptography. Central to this, therefore, is an understanding of isogenies between Kummer surfaces. This section describes the known and new theory of isogenies between squared Kummer surfaces. In particular, let $\mathcal{J}/\mathbb{F}_p$ be the Jacobian of a genus 2 curve in Rosenhain form, with corresponding squared Kummer surface $\mathcal{K}^{\text{Sqr}}$ defined over $\mathbb{F}_p$.

In Section 5.1, we discuss $(2, 2)$-isogenies on (squared) Kummer surfaces, described as pairs of 2-torsion points $L_{i,j}$, $L_{k,\ell}$ on $\mathcal{K}^{\text{Sqr}}$ whose preimages $D_{i,j}$, $D_{k,m}$ generate a $(2, 2)$-subgroup of $\mathcal{J}[2]$. For each of these kernels, in Section 5.2, we give efficient formulae for computing the corresponding $(2, 2)$-isogeny defined over $\overline{\mathbb{F}}_p$. For the construction of efficient cryptographic protocols, we require the isogenies to be defined over the base field $\mathbb{F}_p$. In Section 5.3, we describe how this can be achieved for certain isogenies, which will be sufficient for our application.

Comparison with other works. Computing $(2, 2)$-isogenies between Kummer surfaces has been studied by various other works. Cassels and Flynn [12, Ch. 9] study $(2, 2)$-isogenies between general Kummer surfaces, whilst Dartois, Maino, Pope, and Robert [27] use the theta model to give efficient formulæ for computing $(2, 2)$-isogenies between canonical Kummer surfaces, later improved by [53, §8]. In the latter work, the authors also provide a constant-time implementation of their algorithm in Rust. In constrast, in Section 5.2 we focus instead on deriving isogenies between squared Kummer surfaces.

5.1 From subgroups to (2, 2)-isogenies

Consider an (N, N)-subgroup $G \subseteq \mathcal{J}_1[N]$ (i.e., a group $G = \langle D_R, D_S \rangle$ generated by two N-torsion points $D_R, D_S \in \mathcal{J}_1[N]$ with $e_N(R, S) = 1$, where e_N is the N-Weil pairing). Any (N, N)-*isogeny* between Jacobians of genus 2 curves is a surjective finite morphism $\Phi\colon \mathcal{J}_1 \to \mathcal{J}_2 = \mathcal{J}_1/G$, with kernel a (N, N)-subgroup

$G \subseteq \mathcal{J}_1[N]$, and where $\Phi(\mathcal{O}_1) = \mathcal{O}_2$. Any (N, N)-isogeny Φ descends to a morphism of squared Kummer surfaces $\varphi\colon \mathcal{K}_1^{\mathrm{Sqr}} \to \mathcal{K}_2^{\mathrm{Sqr}}$, such that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{J}_1 & \xrightarrow{\Phi} & \mathcal{J}_2 \\ \downarrow{\scriptstyle \rho^{\mathrm{Sqr}}} & & \downarrow{\scriptstyle \rho^{\mathrm{Sqr}}} \\ \mathcal{K}_1^{\mathrm{Sqr}} & \xrightarrow{\varphi} & \mathcal{K}_2^{\mathrm{Sqr}} \end{array}$$

With our target application in mind, we focus in particular to the case $N = 2$. By abuse of notation, we then call φ a $(2, 2)$-*isogeny of squared Kummer surfaces*, whose kernel is given by the image of G in $\mathcal{K}_1^{\mathrm{Sqr}}$.

To construct such $(2, 2)$-isogenies, we must understand how the map φ is derived from two 2-torsion points in $\mathcal{K}^{\mathrm{Sqr}}[2]$, described in Section 2.4. As long as $\{i, j\} \cap \{k, \ell\} = \emptyset$, two points $L_{i,j}, L_{k,\ell} \in \mathcal{K}[2]$ will generate a $(2, 2)$-subgroup, and thus give us a kernel of a $(2, 2)$-isogeny[15]. These fifteen $(2, 2)$-subgroups are given by $\langle L_{i,j}, L_{k,\ell}\rangle = \{\mathbf{o},\ L_{i,j},\ L_{k,\ell},\ L_{i,j} + L_{k,\ell}\}$, where

$$(i, j, k, \ell) \in \left\{ \begin{array}{l} (1,2,3,4), (1,2,4,6), (1,2,3,6), (2,3,5,6), (1,3,5,6), \\ (1,6,3,4), (2,6,3,4), (2,3,4,5), (1,3,4,5), (1,4,3,6), \\ (2,4,3,6), (2,3,4,6), (1,3,4,6), (1,4,3,5), (2,4,3,5) \end{array} \right\}$$

Note here that $L_{i,j} + L_{k,\ell} = L_{m,n}$ where $\{m, n\} = \{1, 2, 3, 4, 5, 6\} \backslash \{i, j, k, \ell\}$. This gives fifteen corresponding $(2, 2)$-isogenies given by $\varphi_{ijk\ell} : \mathcal{K}_1^{\mathrm{Sqr}} \to \mathcal{K}_2^{\mathrm{Sqr}} = \mathcal{K}_1^{\mathrm{Sqr}}/\langle L_{i,j}, L_{k,\ell}\rangle$.

5.2 Isogenies defined over $\overline{\mathbb{F}}_p$

We first analyse the general case, where the $(2, 2)$-isogenies are defined over $\overline{\mathbb{F}}_p$. For each $(2, 2)$-subgroup G, we associate a morphism $\alpha : \mathcal{K}_1^{\mathrm{Sqr}} \to \mathcal{K}_1^{\mathrm{Sqr}}$ induced by a linear map on $\mathbb{A}^4$ defined by a matrix $\mathbf{A}$ whose entries lie in $\{0, \pm 1, \pm i\}$, where i is the root of $x^2 + 1 \in \mathbb{F}_p[x]$, such that the corresponding $(2, 2)$-isogeny φ is given by

$$\varphi := \mathtt{S} \circ \alpha \circ \mathtt{C}_{\mathtt{Inv}(A:B:C:D)} \circ \mathtt{H},$$

where the maps $\mathtt{S}, \mathtt{C}$ and $\mathtt{H}$ are as defined in Section 2.6. The matrix $\mathbf{A}$ for each $(2, 2)$-subgroup is specified in [21, Appendix A].

Field of definition of the $(2, 2)$-isogeny. For $\mathcal{K}^{\mathrm{Sqr}}$ defined over $\mathbb{F}_p$, the maps $\mathtt{H}$ and $\mathtt{S}$ are defined over $\mathbb{F}_p$. Let $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$, where i is as defined above. The linear map α is defined over $\mathbb{F}_{p^2}$. When $\mathbf{A}$ contains entries in $\{0, \pm 1\}$, however, it is defined over $\mathbb{F}_p$. This is the case for kernels $\langle L_{i,j}, L_{k,\ell}\rangle$, where

$$(i, j, k, \ell) \in \{(1,2,3,4), (1,2,4,6), (2,3,5,6), (1,6,3,4), (2,3,4,5), (1,4,3,6)\}.$$

[15] This is equivalent to a quadratic splitting [57] of the curve equation as used to compute Richelot isogenies.

The scaling map $\mathtt{C}_{\mathtt{Inv}(A:B:C:D)}$, however, requires the computation of $(A : B : C : D)$ from $\mathbf{o} = (\mu_1 : \mu_2 : \mu_3 : \mu_4)$ which in turn requires at most three square roots (and a handful of additions) in $\overline{\mathbb{F}}_p$, using the fact that

$$(A^2 : B^2 : C^2 : D^2) = \mathtt{H}(\mathbf{o}).$$

In some cases, however, $(A : B : C : D)$ can be computed more efficiently without taking square roots.

For the kernel $\langle L_{1,2}, L_{3,4} \rangle$, the map α is the identity, and so the corresponding $(2,2)$-isogeny is given by

$$\varphi = \mathtt{S} \circ \mathtt{C}_{\mathtt{Inv}(A:B:C:D)} \circ \mathtt{H} = \mathtt{C}_{\mathtt{Inv}(A^2:B^2:C^2:D^2)} \circ \mathtt{S} \circ \mathtt{H},$$

where we swap $\mathtt{S}$ and $\mathtt{C}$ to avoid taking square roots, so φ is defined over $\mathbb{F}_p$.

For kernels $\langle L_{i,j}, L_{k,\ell} \rangle$, where

$$(i,j,k,\ell) \in \left\{ \begin{array}{l} (2,3,4,5), (1,3,4,5), (1,4,3,6), (2,4,3,6), \\ (2,3,4,6), (1,3,4,6), (1,4,3,5), (2,4,3,5) \end{array} \right\},$$

the required square roots can be extracted from the 4-torsion points lying above the kernel points. More precisely, let R, S be such that $[2]R = L_{i,j}$ and $[2]S = L_{k,\ell}$, and let $h(P)_i$ be the i-th coordinate of $\mathtt{H}(P)$. For the sake of clarity, we suppose that $L_{i,j}$ and $L_{k,\ell}$ are of the form $(1 : \star : 0 : 0)$ and $(1 : 0 : \star : 0)$, respectively. The other cases follow similarly.

We compute the scaling value $(1/A : 1/B : 1/C : 1/D)$ using the points lying above the kernel by noting that

$$\frac{A}{B} = c_1 \frac{h(R)_1}{h(R)_2}, \ \frac{A}{C} = c_2 \frac{h(S)_1}{h(S)_3}, \text{ and } \frac{C}{D} = c_3 \frac{h(R)_3}{h(R)_4}, \tag{13}$$

for some constants $c_i \in \{1, -i\}$. From this, we get $(1/A : 1/B : 1/C : 1/D)$ by

$$\Big(h(R)_2 h(S)_3 h(S)_3 h(R)_4 : c_1 h(R)_1 h(S)_3 h(S)_3 h(R)_4 : \\ : c_2 h(R)_2 h(S)_1 h(S)_3 h(R)_4 : c_2 c_3 h(R)_2 h(S)_3 h(S)_1 h(R)_3 \Big).$$

We see that $(1/A : 1/B : 1/C : 1/D)$ requires no inversions, and only needs operations in $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$, assuming we have $\mathbb{F}_p$- or $\mathbb{F}_{p^2}$-rational 4-torsion (respectively) and depending on the constants c_i, which are given in Appendix D.

5.3 $(2,2)$-isogenies defined over $\mathbb{F}_p$

To obtain efficient protocols, we are interested in $(2,2)$-isogenies that are defined over $\mathbb{F}_p$. Such isogenies arise from kernels such that α is defined over $\mathbb{F}_p$ and where the scaling point can be computed using $\mathbb{F}_p$-operations, i.e., the constants c_i involved lie in $\mathbb{F}_p$ and we have $\mathbb{F}_p$-rational points R, S lying above $L_{i,j}$, $L_{k,\ell}$, respectively.

Assuming we have rational 4-torsion points lying above, the isogeny φ is defined over $\mathbb{F}_p$ for kernels $\langle L_{i,j}, L_{k,\ell}\rangle$ where

$$(i,j,k,\ell) \in \{(1,2,3,4),(2,3,4,5),(1,4,3,6)\}\,.$$

These kernels coincide precisely with the isogenies derived by Costello [24] which we will require in Section 7 to define SQIsign verification using squared Kummer surfaces. However, when using elliptic Kummer surfaces $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\lambda\mu}$ arising from Scholten's construction, the kernels can be generated by a *single* point rather than a subgroup of $\mathcal{K}^{\mathrm{Sqr}}[2]$. We discuss this in more depth in Section 6.

Cost of computing a $(2,2)$-isogeny over $\mathbb{F}_p$. The maps `H`, `C`, `Inv` and `S` together cost $14\mathbf{M}$ and $8\mathbf{a}$. For kernel $\langle L_{1,2}, L_{3,4}\rangle$ we compute $(A^2 : B^2 : C^2 : D^2)$ as $\mathtt{H}(\mathbf{o})$ using $8\mathbf{a}$, meaning the corresponding $(2,2)$-isogeny requires $14\mathbf{M}$ and $16\mathbf{a}$ to compute. For kernels $\langle L_{2,3}, L_{4,5}\rangle$ and $\langle L_{1,4}, L_{3,6}\rangle$, we compute the scaling point $(1/A : 1/B : 1/C : 1/D)$ using the 4-torsion points, as shown in Equation (13), with (at most) $6\mathbf{M}$ and $16\mathbf{a}$.

6 $(2,2)$-isogenies on elliptic Kummer surfaces

In this section, we specialise to $(2^n, 2^n)$-isogenies between elliptic Kummer surfaces as derived from Scholten's construction. In particular, we consider $(2^n, 2^n)$-isogenies generated by a single point $P \in \mathcal{K}[2^n]$ that arise from 2-isogenies between elliptic curves. We refer to such isogenies, induced by isogenies between elliptic curves, as *elliptic* isogenies.

Consider the elliptic curve $E_\alpha/\mathbb{F}_{p^2} : x(x-\alpha)(x-\frac{1}{\alpha})$. There are three 2-isogenies with kernels generated by 2-torsion points $D_0 = (0,0)$, $D_\alpha = (\alpha, 0)$, $D_{1/\alpha} = (\frac{1}{\alpha}, 0) \in E_\alpha(\mathbb{F}_{p^2})[2]$. We obtain the elliptic Kummer surface $\mathcal{K}_\alpha/\mathbb{F}_p$ associated to E_α using Scholten's construction. We first describe the isogenies $\varphi_i : \mathcal{K}_\alpha \to \mathcal{K}'_\alpha$ that correspond to the elliptic curve isogenies $\phi : E_\alpha \to E'_\alpha = E_\alpha/\langle D\rangle$, thus recovering the isogenies given by Costello [24]. In particular, these elliptic $(2,2)$-isogenies can be computed more efficiently than quoted for general $(2,2)$-isogenies between Kummer surfaces arising from theta coordinates [27]. We then discuss how to use these formulæ to efficiently compute *chains* of elliptic isogenies between these Kummer surfaces. This will be key to our application in Section 7 on SQIsign verification, as there we will need to compute $(2^n, 2^n)$-isogenies.

6.1 Elliptic $(2,2)$-isogenies

Let $D \in E_\alpha[8]$ be a 8-torsion point such that $[4]D \in \{D_0, D_\alpha, D_{1/\alpha}\}$. We assume throughout this section that D is $\mathbb{F}_{p^2}$-rational so that the image of D on $\mathcal{K}_\alpha$ is $\mathbb{F}_p$-rational. We remark that as E_α is supersingular, we can ensure there is an $\mathbb{F}_{p^2}$-rational 8-torsion point by enforcing $8 \mid \#E_\alpha(\mathbb{F}_{p^2}) = (p \pm 1)^2$.

Recall that $\bar{\eta} : E_\alpha \to \mathcal{K}_\alpha$ is itself a $(2,2)$-isogeny, thus, when we map D down to the Kummer surface $\mathcal{K}_\alpha$, we obtain a 4-torsion point $P := \bar{\eta}(D)$. Furthermore, $[2]P$ completely describes the $(2,2)$-isogeny corresponding to the elliptic curve isogeny ϕ. We depict this in Figure 3.

In the lemma that follows, we give explicit equations for the 3 possible elliptic $(2,2)$-isogenies corresponding to the 2-isogenies generated by $D_0, D_\alpha, D_{1/\alpha}$ on E_α In particular, we see that the isogenies defined over $\overline{\mathbb{F}}_p$ given in Section 5.3 collapse to the isogenies given by Costello [24] in this special setting when $\nu = \lambda\mu$.

Lemma 9 ([24]). *Let $D \in E_\alpha[8]$ such that $[4]D \in \{D_0, D_\alpha, D_{1/\alpha}\}$, and define $\mathcal{K}_\alpha$ be the corresponding elliptic Kummer surface with identity $\mathbf{o}$. Let $P' := \bar{\eta}(D) \in \mathcal{K}_\alpha[4]$ and $P := [2]P' \in \mathcal{K}_\alpha[2]$, and denote their images under the Hadamard map by $\mathtt{H}(P) = (h_1 : h_2 : h_3 : h_4)$ and $\mathtt{H}(P') = (h'_1 : h'_2 : h'_3 : h'_4)$. Then:*

1. *If $D = D_0$, then $P \in \{L_{5,6}, L_{3,4}\}$ describes the isogeny given by $\varphi_0 = \mathtt{C}_S \circ \mathtt{S} \circ \mathtt{H}$, where $S = \mathtt{Inv}(\mathtt{H}(\mathbf{o}))$.*
2. *If $D = D_\alpha$, then $P \in \{L_{2,3}, L_{1,6}\}$ describes $\varphi_\alpha = \mathtt{S} \circ \mathtt{H} \circ \mathtt{C}_S \circ \mathtt{H}$, where*
   - *$S = (h_2h'_4 : h_1h'_4 : h_2h'_1 : h_2h'_1)$, if $P = L_{1,6}$, or*
   - *$S = (h_2h'_3 : h_1h'_3 : h_2h'_1 : h_2h'_1)$, if $P = L_{2,3}$.*
3. *If $D = D_{1/\alpha}$, then $P \in \{L_{1,4}, L_{2,5}\}$ describes $\varphi_{1/\alpha} = \mathtt{S} \circ \mathtt{H} \circ \mathtt{C}_S \circ \mathtt{H}$, where*
   - *$S = (h_2h'_4 : h_1h'_4 : h_2h'_1 : h_2h'_1)$, if $P = L_{2,5}$, or*
   - *$S = (h_2h'_3 : h_1h'_3 : h_2h'_1 : h_2h'_1)$, if $P = L_{1,4}$.*

We briefly describe how the isogenies given in Section 5.3 that can be defined over $\mathbb{F}_p$ collapse to the isogenies in Lemma 9 in this specific setting.

The isogeny φ_0 follows directly noting that α is the identity map. For the isogeny φ_α from Item 2, we have $\alpha = \mathtt{H}$ and $S = \mathtt{Inv}(A : B : C : D)$, where S is as in the statement of the lemma. Indeed, observing that for ellpitic Kummer surfaces we have $C = D = 1$, we have that $\mathtt{H}(P) = (A : B : B : A)$ and so $h_1/h_2 = A/B$.

When $P = L_{2,3} = (1 : 0 : \star : 0)$, using Equation (13) we have that $h'_1/h'_3 = A$. Therefore $S = (1 : h_1/h_2 : h'_1/h'_3 : h'_1/h'_3)$. In the other case, $P = (1 : 0 : 0 : \star)$ and $h'_1/h'_4 = A$. Therefore $S = (1 : h_1/h_2 : h'_1/h'_4 : h'_1/h'_4)$.

For the isogeny $\varphi_{1/\alpha}$ from Item 3, we find that $\alpha(X_1 : X_2 : X_3 : X_4) = \mathtt{H}(-X_1 : X_2 : X_3 : X_4)$. We can verify that $S = \mathtt{Inv}(-A : B : C : D)$ in a similar method to above, and so $\alpha \circ \mathtt{C}_{\mathtt{Inv}(A : B : C : D)} = \mathtt{H} \circ \mathtt{C}_S$, as required.

In this way, the isogenies are fully described by a point P, rather than two points as in the previous sections.

Remark 5. If we only have $\mathbb{F}_{p^2}$-rational 4-torsion on E_α, computing the isogeny corresponding to D_α or $D_{1/\alpha}$ requires the computation of 3 square roots (in $\mathbb{F}_{p^2}$) as discussed in Section 5.2.

In what follows, we describe how we compute the codomain of elliptic isogenies, the Rosenhain invariants of the codomain, and the image of points through the isogeny. We give explicit counts for each procedure, which refer to the optimised implementation of each routine found in the accompanying code.

Computing the codomain of an elliptic isogeny. To compute the image of an elliptic $(2,2)$-isogeny φ, it suffices to compute the constants defining the codomain, given by $\varphi(\mathbf{o})$. For isogeny 1, we have that $\varphi(\mathbf{o}) = \mathtt{H}(\mathbf{o})$, which can be computed in $8\mathbf{a}$. For isogenies 2 and 3, we compute the constants by evaluating φ at $\mathbf{o}$ in $11\mathbf{M}$ and $32\mathbf{a}$.

Computing the Rosenhain invariants of the codomain. Key to many of the algorithms in Sections 3 and 4 is the knowledge of the Rosenhain invariants of the curve $\mathcal{C}_\alpha$ associated to the Kummer surface $\mathcal{K}_\alpha$. Gaudry [35, §4.2] shows that for elliptic Kummer surfaces, we have

$$\lambda = \frac{\mu_1}{\mu_2}, \quad \mu = \frac{\tau\mu_2 - 1}{\tau\lambda\mu_2 - 1}, \quad \nu = \lambda\mu. \tag{14}$$

Thus, to compute the Rosenhain invariants of the codomain, it suffices to compute τ, which we can extract from the 2-torsion points on the Kummer surface $\mathcal{K}_\alpha$ in the following sense. Consider the elliptic isogeny $\varphi : \mathcal{K}_\alpha \to \mathcal{K}'_\alpha$ described by $K \in \mathcal{K}_\alpha[2]$. Assuming we have $\mathbb{F}_p$-rational 4-torsion on the domain Kummer surface, let K' be the $\mathbb{F}_p$-rational point such that $K = [2]K'$. Then, $\varphi(K')$ is a point of order 2. If it describes φ_α or $\varphi_{1/\alpha}$, using the description of the 2-torsion on elliptic Kummer surfaces in Section 2.5, we find that the non-zero coordinates of $\varphi(K')$ will be 1 and τ or $1/\tau$ (after normalisation), respectively. We use one bit of information to communicate whether we compute τ or $1/\tau$, which we then use to compute the Rosenhain invariants of the codomain from Equation (14).

Pushing a point through an elliptic isogeny. This amounts to evaluating the maps from Lemma 9 at a point $(X_1 : X_2 : X_3 : X_4)$. We assume we have the point S used in the scaling map $\mathtt{C}_S$ defined in Lemma 9. For isogeny 1, this takes $8\mathbf{M}$ and $8\mathbf{a}$. For isogenies 2 and 3, this takes of $8\mathbf{M}$ and $16\mathbf{a}$.

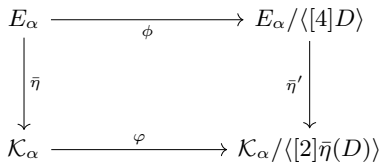


Fig. 3. The elliptic $(2,2)$-isogeny φ induced by a 2-isogeny ϕ.

6.2 Chains of elliptic $(2,2)$-isogenies

Following the blueprint from the previous section, assuming $\mathbb{F}_{p^2}$-rational 2^{n+2}-torsion on E_α we can compute an $\mathbb{F}_p$-rational elliptic $(2^n, 2^n)$-isogeny between elliptic Kummer surfaces, by taking elliptic $(2,2)$-isogeny steps and using the $\mathbb{F}_p$-rational 4-torsion on $\mathcal{K}_\alpha$ at each step to compute the point S needed for the scaling map. We depict this in Figure 4.

For each isogeny $\varphi_0, \varphi_\alpha$, and $\varphi_{1/\alpha}$ which maps $\mathcal{K}_\alpha \to \mathcal{K}'_\alpha$, the dual isogeny is of type 1 in Lemma 9, with kernel $\langle L'_{1,2}, L'_{3,4} \rangle \subset K'_\alpha$, namely $\varphi'_0 : \mathcal{K}'_\alpha \to \mathcal{K}_\alpha$. Therefore, to avoid *backtracking*, after the first step we only take $(2,2)$-isogenies of type 2 and 3, namely φ_α or $\varphi_{1/\alpha}$. We can therefore use the technique described in the previous section to compute the Rosenhain invariants corresponding to the codomain Kummer surface, where we require $\mathbb{F}_{p^2}$-rational 2^{n+2}-torsion on E_α. For most cryptographic applications, we can enforce rational 2^{n+2}-torsion by letting $2^{n+2} \mid p^2 - 1$, as E_α is supersingular.

Strategies. Given a point $K \in \mathcal{K}_\alpha[2^{n+1}]$, we compute the corresponding $(2^n, 2^n)$-isogeny as a chain of elliptic $(2,2)$-isogenies of length n. A naive way of doing this is the following. Set $K_0 := K$ and $\mathcal{K}_0 := \mathcal{K}_\alpha$, and then do the following steps for each $k = 1$ to n:

1. Compute the 4-torsion point $P' = [2^{n-i}]K_i$ and 2-torsion point $P = [2]P'$.
2. From Lemma 9, use P and P' to compute the $(2,2)$-isogeny $\varphi_i : \mathcal{K}_{i-1} \to \mathcal{K}_i$ corresponding to P.
3. Compute $K_{i+1} = \varphi(K_i)$ and compute the constants defining the codomain Kummer surface $\mathcal{K}_i$.

The elliptic $(2^n, 2^n)$-isogeny φ described by K will be $\varphi_n \circ \cdots \circ \varphi_1 : \mathcal{K}_0 \to \mathcal{K}_n$, as depicted in Figure 4.

$$\begin{array}{ccccccccc}
E_\alpha & \xrightarrow{\phi_1} & E_1 & \xrightarrow{\phi_2} & \cdots & \xrightarrow{\phi_{n-1}} & E_{n-1} & \xrightarrow{\phi_n} & E_\alpha/\langle [4]D \rangle \\
\downarrow{\scriptstyle \bar{\eta}} & & \downarrow & & & & \downarrow & & \downarrow{\scriptstyle \bar{\eta}'} \\
\mathcal{K}_\alpha & \xrightarrow[\varphi_1]{} & \mathcal{K}_1 & \xrightarrow[\varphi_2]{} & \cdots & \xrightarrow[\varphi_{n-1}]{} & \mathcal{K}_{n-1} & \xrightarrow[\varphi_n]{} & \mathcal{K}_\alpha/\langle [2]\bar{\eta}(D) \rangle
\end{array}$$

Fig. 4. The elliptic $(2^n, 2^n)$-isogeny φ induced by the 2^n-isogeny ϕ .

A better way to compute chains of isogenies is to use *optimal strategies*. This was first introduced in the context of SIDH/SIKE [28], and has shown to be adaptable to the Kummer surface setting [17, 21, 27]. By using optimal strategies, we reduce the number of doublings performed by instead storing intermediate points obtained during repeated doublings and pushing them through the isogeny. Taking the cost model $\mathbf{M} = 0.8\mathbf{S}$, the cost of doubling is around $1.8\mathbf{x}$ the cost of computing the image of a point under the isogeny. By shifting the cost in this way, and noting that in our setting we only need to push a single Kummer point through the isogeny (rather than two points), we precompute the optimal strategy to take when computing our chain of $(2,2)$-isogenies. Using these strategies in our implementation, we observe roughly a factor two reduction in the cost compared to the naive isogeny evaluation. We thank Michael Meyer for the implementation of this optimization.

7 SQIsign Verification on Kummer Surfaces

We now have the necessary tools in place to turn to our target application: performing SQIsign on Kummer surfaces. We refer to [16, 29] for more details on SQIsign. The essential tool is Scholten's construction, which allows us to construct an elliptic Kummer surface $\mathcal{K}_\alpha = \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\lambda\mu}$ defined over $\mathbb{F}_p$ corresponding to an elliptic curve E_α defined over $\mathbb{F}_{p^2}$. Beyond Scholten's construction, we require the techniques from Sections 3 and 4 to enable efficient compression of isogenies between Kummer surfaces, and we require the theory from Sections 5 and 6 to efficiently compute elliptic isogenies between Kummer surfaces. We emphasise that we initiate this construction for supersingular curves E_α which ensures superspecial elliptic Kummer surfaces $\mathcal{K}_\alpha$.

7.1 Using Scholten's construction

Let $E_\alpha : y = x(x-\alpha)(x-\frac{1}{\alpha})$ be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let f be the largest positive integer such that $2^f \mid p+1$. This ensures we have $\mathbb{F}_{p^2}$-rational 2^f-torsion on E_α. A SQIsign response isogeny φ_{resp} is an isogeny of degree 2^e, with $e \approx 1000$. To compute only $\mathbb{F}_{p^2}$-rational 2-isogenies, SQIsign verification splits up φ_{resp} into $n = \lceil e/f \rceil$ *blocks* $\varphi_i : E_i \to E_{i+1}$, such that $\varphi_{\text{resp}} = \varphi_n \circ \ldots \circ \varphi_1$, with $\ker \varphi_i = \langle K_i \rangle$ for some $K_i \in E_i(\mathbb{F}_{p^2})$ of order 2^f. We use Scholten's construction to compute the superspecial elliptic Kummer $\mathcal{K}_i^{\text{Sqr}}$ associated to the elliptic curve E_i, giving us the following commuting diagram.

$$\begin{array}{ccccccc}
E_0 & \xrightarrow{\phi_1} & E_1 & \xrightarrow{\phi_2} & \ldots & \xrightarrow{\phi_n} & E_n \\
\downarrow{\scriptstyle \bar{\eta}_0} & & \downarrow{\scriptstyle \bar{\eta}_1} & & & & \downarrow{\scriptstyle \bar{\eta}_n} \\
\mathcal{K}_0^{\text{Sqr}} & \xrightarrow{\varphi_1} & \mathcal{K}_1^{\text{Sqr}} & \xrightarrow{\varphi_2} & \ldots & \xrightarrow{\varphi_n} & \mathcal{K}_n^{\text{Sqr}}
\end{array}$$

Thus, we can verify a SQIsign response $\varphi_{\text{resp}} : E_0 \to E_n$ by computing the corresponding chain of elliptic $(2^f, 2^f)$-isogenies $\varphi : \mathcal{K}_0^{\text{Sqr}} \to \mathcal{K}_n^{\text{Sqr}}$, now defined over $\mathbb{F}_p$. By Lemma 4, one can also see the first and last steps of this diagram as gluing and splitting isogenies. Thus, $\varphi_{\text{resp}} : E_0 \to E_n$ can similarly be viewed as a two-dimensional isogeny

$$\Phi_{\text{resp}} : E_0 \times E_0^{(p)} \to \mathcal{K}_0^{\text{Sqr}} \to \ldots \to \mathcal{K}_n^{\text{Sqr}} \to E_n \times E_n^{(p)}.$$

7.2 Computing a single block

We first describe how to transport the computation of a single block $\varphi_i : E_i \to E_{i+1}$ to a computation involving Kummer surfaces.

We start with a point P on $E_i(\mathbb{F}_{p^2})$ of order 2^f. Pushing P down to $\mathcal{K}_i^{\text{Sqr}}$ through the $(2,2)$-isogeny $\bar{\eta}_i$, the point $\bar{\eta}_i(P)$ has order 2^{f-1} on $\mathcal{K}_i^{\text{Sqr}}(\mathbb{F}_p)$, which is the maximal power-of-two torsion on $\mathcal{K}$. As depicted in Section 6, the elliptic

isogenies φ_i we derive from $\bar{\eta}_i(P)$ are those given by Lemma 9. It is most cost effective to use $\bar{\eta}_i(P)$ to perform $f-2$ $(2,2)$-isogenies, so that we always have $\mathbb{F}_p$-rational 4-torsion lying above our kernel generators to compute our isogenies. This elliptic $(2^{f-2}, 2^{f-2})$-isogeny φ_i then corresponds to the elliptic curve isogeny ϕ_i with kernel $\langle [4]P \rangle \subset E_i[2^{f-2}]$.

In this way, by moving to the Kummers we lose 2 bits in the length of the isogeny per block. However, this should not have a large effect on performance as long as we perform SQIsign verification with the same number of blocks, as observed in AprèsSQI [22].

7.3 Uncompressed SQIsign signatures

Uncompressed SQIsign is a variant of SQIsign where we assume that the kernel K_i of the i-th block $\varphi_i : E_i \to E_{i+1}$ is simply given as (the x-coordinate of) a point K_i, not using any compression techniques. SQIsign verification then consists of the recomputation of two isogenies: the above-mentioned *response* $\varphi_{\text{resp}} : E_A \to E_2$ and (the dual of) the *challenge* $\varphi_{\text{chall}} : E_1 \to E_2$.

The challenge isogeny. As we only have considered efficient $(2,2)$-isogenies in this work, we require the challenge isogeny to be of degree 2^λ, where λ is the security parameter. Hence, we require $f \geq \lambda$ to be able to describe the challenge isogeny again using a single Kummer point $K \in \mathcal{K}$. Beyond that, the signer needs to be slightly more careful in constructing the challenge isogeny: they should not use the deterministic basis of E_1 to hash to a random isogeny φ_{chall}, as the verifier will only see the Kummer surface, on which the deterministically sampled basis is different. So, the signer computes the associated Kummer surface $\mathcal{K}_2^{\text{Sqr}}$ and hashes to a challenge isogeny φ_{chall}, then lifts φ_{chall} to the elliptic curve to compute the curve E_2. Only then can the signer craft a response isogeny between E_A and E_2 and push this down to Kummer surfaces.

The response isogeny. With the theory given in Sections 2 and 6, we can use Scholten's construction to push the full isogeny φ_{resp} to give an isogeny between sqaured Kummer surfaces using Scholten's construction. We split up φ_{resp} into $n = \lceil e/(f-2) \rceil$ isogenies of degree 2^{f-2} to ensure that we can perform a single block on the Kummer isogenies when starting with a point of order 2^f on the elliptic curve. The response isogeny is then given as $(x_1, \ldots, x_n)$, where x_i is the x-coordinate of a point K_i generating the kernel of the i-block φ_i.

Remark 6. A minor difference between the isogeny on the elliptic curves and the Kummer surfaces is that we sometimes end up on the *twist* of $\mathcal{K}_{i+1}^{\text{Sqr}}$, which we need to correct for. However, we can simply communicate this in the signature at the cost of 1 bit, and the cost for twist correction is negligible (see Section 2.9). This does not impact security, as one could lexicographically decide on either Kummer surface to normalise this choice, and all information required is public.

7.4 Compressed SQIsign signatures

Compressing SQIsign signatures requires many of the general techniques described in Sections 3 and 4. More generally, we apply the theory we developed to improve the performance of isogeny-based cryptography on Kummer surfaces.

Before describing compressed isogenies between Kummer surfaces, we recall the two core tools used for compression of elliptic curve SQIsign signatures:

1. An efficient and deterministic method to sample a basis P, Q for $E[2^f]$,
2. A recomputation of K_i as $P + [s]Q$ given a scalar $s \in \mathbb{Z}/2^f\mathbb{Z}$.

AprèsSQI [22] gives a detailed analysis of the cost of both steps, with several optimizations which we can generalise to the higher-dimensional case.

Using the sampling method from Section 4.3, we can compress the kernel point K more efficiently than the general method sketched in PointCompression, by using that, in signing, we can compute the divisor D_K on the Jacobian $\mathcal{J}_\mathcal{C}$ corresponding to K as the image of the kernel point in the associated elliptic curve. This allows us to forgo Line 1 of Algorithm 4 (which maps the kernel point K to D_K), and instead start the point compression directly on $\mathcal{J}_\mathcal{C}$, with the caveat that we must ensure the results stay consistent for the verifier.

Our approach is as follows. Given the element $D_K \in \mathrm{Im}(\eta) \subset \mathcal{J}_\mathcal{C}(\mathbb{F}_p)$, the signer samples $D \in \mathcal{J}_\mathcal{C}(\mathbb{F}_p)$ by sampling a deterministic sequence of points $P \in \mathcal{C}_{\lambda,\mu,\nu}$ until $D = (P) + ((w_i, 0)) - D_\infty$ has the same profile as D_K and sets $D_P \leftarrow D$. The signer then samples D_Q similarly until we ensure $D_K \in \langle D_P, D_Q \rangle$ and computes a, b such that $D_K = [a][\frac{p+1}{2^f}]D_p + [b][\frac{p+1}{2^f}]D_Q$. As D_K has the same profile as D_P, we are ensured that a is odd and set $s = b/a \in \mathbb{Z}/2^f$. Thus, the signer updates $D_K \leftarrow [s^{-1}]D_K$.

The signer then pushes D_K, D_P and D_Q to $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ as K, P and Q, and derives $D, S \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ using PointDifference. Finally, the signer recomputes both $K_D \leftarrow \texttt{xMUL}(\texttt{3ptLadder}(P, Q, D, s), 2^f)$ and $K_S \leftarrow \texttt{xMUL}(\texttt{3ptLadder}(P, Q, S, s), 2^f)$ to find which one equals K, and adds this information as a bit b. The compression of K is then given as the pair (s, b). The verifier can then use the same deterministic sampling procedure to derive $P, Q \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, and use efficient Kummer arithmetic, instead of expensive Jacobian arithmetic, to derive $K \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$.

7.5 Performance

Our benchmarks, using the same cost model as AprèsSQI, show that SQIsign verification on Kummer surfaces, in comparison to elliptic curves, takes less than 1.5× the number of $\mathbb{F}_p$-operations for both the uncompressed variant and the compressed variant. As shown in [4], the core Kummer arithmetic (`H`, `S` and $\texttt{C}_P$) can be very efficiently vectorised on larger CPUs, where vector units are typically the most powerful computational units. If such efficient vectorisation scales to the primes used in SQIsign, this may potentially allow for faster SQIsign verification on Kummer surfaces than elliptic curves. Furthermore, many of the proposed algorithms in this work are new, and although we have optimised these to the best of our knowledge, we assume further optimizations are possible.

8 Conclusions

We have described, used, and implemented several techniques and tools to advance the toolbox of isogeny-based cryptography in higher dimensions. Using Scholten's construction, we have shown that SQIsign verification can also be viewed as a very unique $(2^n, 2^n)$-isogeny between products of elliptic curves, with relatively little overhead. With vectorised arithmetic [4], this can potentially outperform SQIsign verification between elliptic curves, and compete with the two-dimensional approaches over $\mathbb{F}_{p^2}$ [2, 32, 46] in terms of verification speed.

Two-dimensional approaches seem to achieve close-to-optimal results for the length of the response isogeny, and therefore potential improvements most likely come from lower-level improvements, such as better isogeny formulas and optimised finite field arithmetic. Our approach, however, can still improve in several aspects. First, the techniques developed in this work are novel, and closer analysis might significantly improve their performance. Second, the overall length of the response isogeny is far off from the theoretical best. Improvements to KLPT, or other approaches to achieve a shorter response therefore potentially allow for drastic improvements to verification.

In a more general sense, our work shows that isogeny-based cryptography in higher dimensions has access to a similar toolbox as isogeny-based elliptic-curve cryptography. In particular, the use of Scholten's construction allows to transport primitives to higher dimensions. Using the tools developed in this work, it is not out of reach to similarly analyze higher-dimensional analogues of other isogeny-based cryptosystems.

References

[1] Paulo SLM Barreto, Hae Y Kim, Ben Lynn, and Michael Scott. "Efficient algorithms for pairing-based cryptosystems". In: *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*. Springer. 2002, pp. 354–369 (cit. on p. 20).

[2] Andrea Basso, Luca De Feo, Pierrick Dartois, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. *SQIsign2D-West: The Fast, the Small, and the Safer*. Cryptology ePrint Archive, Paper 2024/760. https://eprint.iacr.org/2024/760. 2024. URL: https://eprint.iacr.org/2024/760 (cit. on pp. 2, 5, 43).

[3] Daniel J. Bernstein. "Elliptic vs. hyperelliptic, part 1". In: (2006). URL: http://cr.yp.to/talks.html#2006.09.20 (cit. on pp. 9, 10).

[4] Daniel J Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Peter Schwabe. "Kummer strikes back: new DH speed records". In: *Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014. Proceedings, Part I 20*. Springer. 2014, pp. 317–337 (cit. on pp. 3, 4, 10, 42, 43).

[5] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. "The SPHINCS+ Signature Framework". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, 2129–2146. ISBN: 9781450367479. DOI: 10.1145/3319535.3363229 (cit. on p. 2).

[6] Daniel J Bernstein and Tanja Lange. "Hyper-and-elliptic-curve cryptography". In: *LMS Journal of computation and Mathematics* 17.A (2014), pp. 181–202 (cit. on p. 17).

[7] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2018, pp. 353–367 (cit. on p. 2).

[8] Joppe W Bos, Craig Costello, Huseyin Hisil, and Kristin Lauter. "Fast cryptography in genus 2". In: *Journal of Cryptology* 29 (2016), pp. 28–60 (cit. on pp. 4, 10, 11, 28).

[9] W. Bosma, J. Cannon, and C. Playoust. "The Magma algebra system. I. The user language". In: *J. Symbolic Comput.* 24.3-4 (1997). Computational algebra and number theory, pp. 235–265 (cit. on p. 4).

[10] Bradley Wayne Brock. *Superspecial curves of genera two and three*. Princeton University, 1993 (cit. on pp. 6, 17).

[11] Peter Bruin. "The Tate pairing for abelian varieties over finite fields". In: *Journal de theorie des nombres de Bordeaux* 23.2 (2011), pp. 323–328 (cit. on p. 21).

[12] J. W. S. Cassels and E. V. Flynn. *Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1996 (cit. on pp. 7, 8, 23, 33, 50).

[13] W. Castryck, T. Decru, and B. Smith. "Hash functions from superspecial genus-2 curves using Richelot isogenies". In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 268–292 (cit. on pp. 6, 17).

[14] Wouter Castryck and Thomas Decru. "An Efficient Key Recovery Attack on SIDH". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447. DOI: 10.1007/978-3-031-30589-4_15. URL: https://doi.org/10.1007/978-3-031-30589-4_15 (cit. on p. 2).

[15] Wouter Castryck, Marc Houben, Simon-Philipp Merz, Marzio Mula, Sam van Buuren, and Frederik Vercauteren. "Weak instances of class group action based cryptography via self-pairings". In: *Annual International Cryptology Conference*. Springer. 2023, pp. 762–792 (cit. on p. 21).

[16] Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa,

Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. *SQIsign: Algorithm specifications and supporting documentation*. National Institute of Standards and Technology. 2023. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/sqisign-spec-web.pdf (cit. on pp. 2, 3, 40).

[17] Jesús-Javier Chi-Domínguez, Amalia Pizarro-Madariaga, and Edgardo Riquelme. *Computing Isogenies of Power-Smooth Degrees Between PPAVs*. Cryptology ePrint Archive, Paper 2023/508. 2023. URL: https://eprint.iacr.org/2023/508 (cit. on p. 39).

[18] David V Chudnovsky and Gregory V Chudnovsky. "Sequences of numbers generated by addition in formal groups and new primality and factorization tests". In: *Advances in Applied Mathematics* 7.4 (1986), pp. 385–434 (cit. on pp. 4, 9).

[19] Ping Ngai Chung, Craig Costello, and Benjamin Smith. "Fast, uniform scalar multiplication for genus 2 Jacobians with fast Kummers". In: *Selected Areas in Cryptography–SAC 2016: 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers 23*. Springer. 2017, pp. 465–481 (cit. on pp. 11, 14, 15).

[20] Maria Corte-Real Santos, Craig Costello, and Jia Shi. "Accelerating the Delfs-Galbraith Algorithm with Fast Subfield Root Detection". In: *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13509. Lecture Notes in Computer Science. Springer, 2022, pp. 285–314. DOI: 10.1007/978-3-031-15982-4_10 (cit. on p. 29).

[21] Maria Corte-Real Santos, Craig Costello, and Benjamin Smith. "Efficient $(3,3)$-isogenies on fast Kummer surfaces". In: *arXiv preprint arXiv:2402.01223* (2024) (cit. on pp. 34, 39).

[22] Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders. "AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing". In: *Advances in Cryptology - EUROCRYPT 2024 - 43nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 2024 (cit. on pp. 2, 3, 20, 21, 23, 41, 42).

[23] Romain Cosset. "Applications of theta functions for hyperelliptic curve cryptography". PhD thesis. Ph. D Thesis, Université Henri Poincaré-Nancy I, 2011 (cit. on p. 11).

[24] Craig Costello. "Computing Supersingular Isogenies on Kummer Surfaces". In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*. Ed. by Thomas Peyrin and Steven D. Galbraith. Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 428–456. DOI: 10.

1007/978-3-030-03332-3_16. URL: https://doi.org/10.1007/978-3-030-03332-3_16 (cit. on pp. 2–4, 14, 17, 18, 30, 36, 37).

[25] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. "Efficient Compression of SIDH Public Keys". In: *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. Lecture Notes in Computer Science. 2017, pp. 679–706. DOI: 10.1007/978-3-319-56620-7_24 (cit. on pp. 20, 29).

[26] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. "SQISignHD: new dimensions in cryptography". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2024, pp. 3–32 (cit. on p. 2).

[27] Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert. *An Algorithmic Approach to* $(2, 2)$*-isogenies in the Theta Model and Applications to Isogeny-based Cryptography*. Cryptology ePrint Archive, Paper 2023/1747. 2023. URL: https://eprint.iacr.org/2023/1747 (cit. on pp. 2–5, 33, 36, 39).

[28] Luca De Feo, David Jao, and Jérôme Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *J. Math. Cryptol.* 8.3 (2014), pp. 209–247. DOI: 10.1515/jmc-2012-0015. URL: https://doi.org/10.1515/jmc-2012-0015 (cit. on p. 39).

[29] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. "SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93. DOI: 10.1007/978-3-030-64837-4_3. URL: https://doi.org/10.1007/978-3-030-64837-4_3 (cit. on pp. 2, 3, 40).

[30] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. "New Algorithms for the Deuring Correspondence - Towards Practical and Secure SQISign Signatures". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 659–690. DOI: 10.1007/978-3-031-30589-4_23. URL: https://doi.org/10.1007/978-3-031-30589-4_23 (cit. on pp. 2, 3).

[31] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "Crystals-dilithium: A lattice-based digital signature scheme". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2018), pp. 238–268 (cit. on p. 2).

[32] Max Duparc and Tako Boris Fouotsa. *SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies*. Cryptology ePrint Archive, Paper 2024/773. https://eprint.iacr.org/2024/773. 2024. URL: https://eprint.iacr.org/2024/773 (cit. on pp. 2, 5, 43).

[33] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. "Falcon: Fast-Fourier lattice-based compact signatures over NTRU". In: *Submission to the NIST's post-quantum cryptography standardization process* 36.5 (2018), pp. 1–75 (cit. on p. 2).

[34] Steven D Galbraith, Florian Hess, and Frederik Vercauteren. "Hyperelliptic pairings". In: *International Conference on Pairing-Based Cryptography*. Springer. 2007, pp. 108–131 (cit. on p. 25).

[35] Pierrick Gaudry. "Fast genus 2 arithmetic based on Theta functions". In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 243–265 (cit. on pp. 4, 9, 10, 12, 15, 26, 28, 31, 38, 50).

[36] Ryuichi Harasawa, Junji Shikata, Joe Suzuki, and Hideki Imai. "Comparing the MOV and FR reductions in elliptic curve cryptography". In: *Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18*. Springer. 1999, pp. 190–205 (cit. on p. 20).

[37] Dale Husemöller. *Elliptic Curves, 2nd edition*. Springer, 2004 (cit. on p. 21).

[38] Jun-ichi Igusa. "Arithmetic variety of moduli for genus two". In: *Annals of Mathematics* 72.3 (1960), pp. 612–649 (cit. on p. 13).

[39] Antoine Joux. "A one round protocol for tripartite Diffie–Hellman". In: *Journal of cryptology* 17.4 (2004), pp. 263–276 (cit. on p. 20).

[40] Dmitrii Koshelev. "Subgroup membership testing on elliptic curves via the Tate pairing". In: *Journal of Cryptographic Engineering* 13.1 (2023), pp. 125–128 (cit. on p. 20).

[41] Ke-Zheng Li and Frans Oort. *Moduli of supersingular abelian varieties*. Vol. 1680. Springer Science & Business Media, 1998 (cit. on pp. 6, 17).

[42] Kaizhan Lin, Weize Wang, Zheng Xu, and Chang-An Zhao. "A faster software implementation of SQISign". In: *IEEE Transactions on Information Theory* (2024) (cit. on p. 20).

[43] David Lubicz and Damien Robert. "Arithmetic on abelian and Kummer varieties". In: *Finite Fields Appl.* 39 (2016), pp. 130–158. ISSN: 1071-5797,1090-2465. DOI: 10.1016/j.ffa.2016.01.009. URL: https://doi.org/10.1016/j.ffa.2016.01.009 (cit. on p. 4).

[44] David Lubicz and Damien Robert. "Efficient pairing computation with theta functions". In: *International Algorithmic Number Theory Symposium*. Springer. 2010, pp. 251–269 (cit. on pp. 4, 26).

[45] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. "A Direct Key Recovery Attack on SIDH". In: *Advances*

in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 448–471. DOI: 10.1007/978-3-031-30589-4_16. URL: https://doi.org/10.1007/978-3-031-30589-4_16 (cit. on p. 2).

[46] Kohei Nakagawa and Hiroshi Onuki. *SQIsign2D-East: A New Signature Scheme Using 2-dimensional Isogenies*. Cryptology ePrint Archive, Paper 2024/771. https://eprint.iacr.org/2024/771. 2024. URL: https://eprint.iacr.org/2024/771 (cit. on pp. 2, 5, 43).

[47] Ryo Ohashi. "On the Rosenhain forms of superspecial curves of genus two". In: *Finite Fields and Their Applications* 97 (2024), p. 102445 (cit. on p. 30).

[48] Stephen Pohlig and Martin Hellman. "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.)" In: *IEEE Transactions on information Theory* 24.1 (1978), pp. 106–110 (cit. on p. 32).

[49] Joost Renes, Peter Schwabe, Benjamin Smith, and Lejla Batina. "μKummer: efficient hyperelliptic signatures and key exchange on microcontrollers". In: *International Conference on Cryptographic Hardware and Embedded Systems*. Springer. 2016, pp. 301–320 (cit. on pp. 4, 10).

[50] Joost Renes and Benjamin Smith. "qDSA: small and secure digital signatures with curve-based Diffie–Hellman key pairs". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2017, pp. 273–302 (cit. on pp. 4, 28, 29).

[51] Damien Robert. "Breaking SIDH in Polynomial Time". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 472–503. DOI: 10.1007/978-3-031-30589-4_17. URL: https://doi.org/10.1007/978-3-031-30589-4_17 (cit. on p. 2).

[52] Damien Robert. *Fast pairings via biextensions and cubical arithmetic*. Cryptology ePrint Archive, Paper 2024/517. 2024. URL: https://eprint.iacr.org/2024/517 (cit. on pp. 4, 26, 52).

[53] Damien Robert. "Some notes on algorithms for abelian varieties". In: *IACR Cryptol. ePrint Arch.* (2024), p. 406. URL: https://eprint.iacr.org/2024/406 (cit. on p. 33).

[54] Damien Robert. *The geometric interpretation of the Tate pairing and its applications*. Cryptology ePrint Archive, Paper 2023/177. 2023. URL: https://eprint.iacr.org/2023/177 (cit. on pp. 21, 22).

[55] Georg Rosenhain. *Abhandlung über die Functionen zweier Variabler mit vier Perioden: welche die Inversen sind der ultra-elliptischen Integrale erster Klasse*. 65. W. Engelmann, 1895 (cit. on p. 14).

[56] Jasper Scholten. "Weil restriction of an elliptic curve over a quadratic extension". In: *Preprint* (2003) (cit. on pp. 2, 3, 16, 17).

[57] Benjamin Andrew Smith. "Explicit endomorphisms and correspondences". PhD thesis. 2005 (cit. on pp. 13, 34).

[58] National Institute of Standards and Technology (NIST). *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process*. 2022. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf (cit. on p. 2).

[59] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.2)*. https://www.sagemath.org. 2021 (cit. on p. 4).

A Addition matrices for Kummer surfaces

The addition by points of order 2 is well-defined on Kummer surfaces, and can be described by a 4×4-matrix. This appendix describes these matrices for the general Kummer surface, as described by Cassels and Flynn [12] as well as for the squared Kummer surface, which is original work.

A.1 The general Kummer surface

Let $L_{i,j} \in \mathcal{K}[2]$, whose x-part of the Mumford representation on $\mathcal{J}$ is given by $a(x) = x^2 - (w_i + w_j)x + w_i \cdot w_j$. Then a divides the defining polynomial f of $\mathcal{C}$. Write $f = a \cdot h$ with $h = \sum h_i x^i$, that is, $h = \prod_{k \neq i,j}(x - w_k)$. Then the matrix $W_{i,j}$ is given by the composition of the 4×3 matrix

$$\begin{pmatrix} g_2^2 h_0 + g_0 g_2 h_2 - g_0^2 h_4 & g_0 g_2 h_3 - g_0 g_1 h_4 & g_1 g_2 h_3 - g_1^2 h_4 + 2 g_0 g_2 h_4 \\ -g_0 g_2 h_1 - g_0 g_1 h_2 + g_0^2 h_3 & g_2^2 h_0 - g_0 g_2 h_2 + g_0^2 h_4 & g_2^2 h_1 - g_1 g_2 h_2 - g_0 g_2 h_3 \\ -g_1^2 h_0 + 2 g_0 g_2 h_0 + g_0 g_1 h_1 & -g_1 g_2 h_0 + g_0 g_2 h_1 & -g_2^2 h_0 + g_0 g_2 h_2 + g_0^2 h_4 \\ w_{41} & w_{42} & w_{43} \end{pmatrix},$$

adjoined on the right by the column $(g_2, -g_1, g_0, w_{44})^T$, with

$$\begin{aligned} w_{41} =& - g_1 g_2^2 h_0 h_1 + g_1^2 g_2 h_0 h_2 + g_0 g_2^2 h_1^2 - 4 g_0 g_2^2 h_0 h_2 \\ & - g_0 g_1 g_2 h_1 h_2 + g_0 g_1 g_2 h_0 h_3 - g_0^2 g_2 h_1 h_3, \\ w_{42} =& g_1^2 g_2 h_0 h_3 - g_1^3 h_0 h_4 - 2 g_0 g_2^2 h_0 h_3 - g_0 g_1 g_2 h_1 h_3 \\ & + 4 g_0 g_1 g_2 h_0 h_4 + g_0 g_1^2 h_1 h_4 - 2 g_0^2 g_2 h_1 h_4, \\ w_{43} =& - g_0 g_2^2 h_1 h_3 - g_0 g_1 g_2 h_2 h_3 + g_0 g_1 g_2 h_1 h_4 + g_0 g_1^2 h_2 h_4 \\ & + g_0^2 g_2 h_3^2 - 4 g_0^2 g_2 h_2 h_4 - g_0^2 g_1 h_3 h_4, \\ w_{44} =& - g_2^2 h_0 - g_0 g_2 h_2 - g_0^2 h_4. \end{aligned}$$

A.2 Addition matrices for canonical and squared Kummer surfaces

The canonical Kummer surface. For the canonical Kummer surface, addition by a point of order 2 is very simple. As described by Gaudry [35], each point of order 2 is some permutation of $\mathbf{o}$, followed by multiplication by -1 for either 0 or 2 of the coordinates. Similarly, addition of a point of order 2 to $P = (T_1 : T_2 : T_3 : T_4)$ is computed by applying the same permutation to the T_i, and multiplying the same coordinates by -1. For example, adding a point $P = (T_1 : T_2 : T_3 : T_4)$ by $(c : -d : -a : b)$ we get $(T_3 : -T_4 : -T_1 : T_2)$, and the addition matrix for $(c : -d : -a : b)$ is given by

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The squared Kummer surface. For squared Kummer surfaces, the addition matrices $W_{i,j}$ that represent $P \mapsto P + L_{i,j}$ can be computed using the algebraic method sketched in Section 2.4. This gives 4×4-matrices in terms of the Rosenhain values λ, μ, ν and the theta constants μ_i. As these matrices look rather daunting in general form yet are easily derivable from the given values, we do not put them in full form here. Their derivation and their description are given in Magma code in the file `wij_squared_kummer.m`.

The elliptic Kummer surface. For elliptic Kummer surfaces we can specialise the derived $W_{i,j}$ for the squared Kummer surface to the case $\mu_3 = \mu_4 = 1$ and $\nu = \lambda \cdot \mu$, which greatly improves their visual appearance. As we use these throughout the work, we give their full versions here.

Let τ and $\tilde{\tau}$ be the roots of $x^2 - Gx + 1$. In particular, $\tilde{\tau} = 1/\tau$, and $\tau + \tilde{\tau} = \mu_1 + \mu_2$. The terms $\frac{\mu_1 - \tau}{\mu_2 - \tau}$, and their $\tilde{\tau}$ variants, appear often in these matrices. For brevity and clarity, we denote them by

$$\gamma := \frac{\mu_1 - \tau}{\mu_2 - \tau}, \quad \tilde{\gamma} := \frac{\mu_1 - \tilde{\tau}}{\mu_2 - \tilde{\tau}}.$$

Then the addition matrices are given by

$$W_{1,2} := \begin{pmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix} \; W_{3,4} := \begin{pmatrix} 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \\ 1&0&0&0 \end{pmatrix} \; W_{5,6} := \begin{pmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \end{pmatrix}$$

$$W_{1,3} := \begin{pmatrix} 1 & -\gamma & \tau\cdot\gamma & -\tau \\ \gamma & -1 & \tau & -\tau\cdot\gamma \\ \tau\cdot\gamma & -\tau & 1 & -\gamma \\ \tau & -\tau\cdot\gamma & \gamma & -1 \end{pmatrix} \; W_{1,4} := \begin{pmatrix} 1 & -\gamma & \tilde{\tau}\cdot\gamma & -\tilde{\tau} \\ \gamma & -1 & \tilde{\tau} & -\tilde{\tau}\cdot\gamma \\ \tilde{\tau}\cdot\gamma & -\tilde{\tau} & 1 & -\gamma \\ \tilde{\tau} & -\tilde{\tau}\cdot\gamma & \gamma & -1 \end{pmatrix}$$

$$W_{1,5} := \begin{pmatrix} 1 & -\tilde{\gamma} & -\tilde{\tau} & \tilde{\tau}\cdot\tilde{\gamma} \\ \tilde{\gamma} & -1 & -\tilde{\tau}\cdot\tilde{\gamma} & \tilde{\tau} \\ \tilde{\tau} & -\tilde{\tau}\cdot\tilde{\gamma} & -1 & \tilde{\gamma} \\ \tilde{\tau}\cdot\tilde{\gamma} & -\tilde{\tau} & -\tilde{\gamma} & 1 \end{pmatrix} \; W_{1,6} := \begin{pmatrix} 1 & -\tilde{\gamma} & -\tau & \tau\cdot\tilde{\gamma} \\ \tilde{\gamma} & -1 & -\tau\cdot\tilde{\gamma} & \tau \\ \tau & -\tau\cdot\tilde{\gamma} & -1 & \tilde{\gamma} \\ \tau\cdot\tilde{\gamma} & -\tau & -\tilde{\gamma} & 1 \end{pmatrix}$$

$$W_{2,3} := \begin{pmatrix} 1 & -\tilde{\gamma} & \tau\cdot\tilde{\gamma} & -\tau \\ \tilde{\gamma} & -1 & \tau & -\tau\cdot\tilde{\gamma} \\ \tau\cdot\tilde{\gamma} & -\tau & 1 & -\tilde{\gamma} \\ \tau & -\tau\cdot\tilde{\gamma} & \tilde{\gamma} & -1 \end{pmatrix} \; W_{2,4} := \begin{pmatrix} 1 & -\tilde{\gamma} & \tilde{\tau}\cdot\tilde{\gamma} & -\tilde{\tau} \\ \tilde{\gamma} & -1 & \tilde{\tau} & -\tilde{\tau}\cdot\tilde{\gamma} \\ \tilde{\tau}\cdot\tilde{\gamma} & -\tilde{\tau} & 1 & -\tilde{\gamma} \\ \tilde{\tau} & -\tilde{\tau}\cdot\tilde{\gamma} & \tilde{\gamma} & -1 \end{pmatrix}$$

$$W_{2,5} := \begin{pmatrix} 1 & -\gamma & -\tilde{\tau} & \tilde{\tau}\cdot\gamma \\ \gamma & -1 & -\tilde{\tau}\cdot\gamma & \tilde{\tau} \\ \tilde{\tau} & -\tilde{\tau}\cdot\gamma & -1 & \gamma \\ \tilde{\tau}\cdot\gamma & -\tilde{\tau} & -\gamma & 1 \end{pmatrix} \; W_{2,6} := \begin{pmatrix} 1 & -\gamma & -\tau & \tau\cdot\gamma \\ \gamma & -1 & -\tau\cdot\gamma & \tau \\ \tau & -\tau\cdot\gamma & -1 & \gamma \\ \tau\cdot\gamma & -\tau & -\gamma & 1 \end{pmatrix}$$

$$W_{3,5} := \begin{pmatrix} 1 & 1 & -\tau & -\tilde{\tau} \\ 1 & 1 & -\tilde{\tau} & -\tau \\ \tau & \tilde{\tau} & -1 & -1 \\ \tilde{\tau} & \tau & -1 & -1 \end{pmatrix} \quad W_{3,6} := \begin{pmatrix} 1 & \tilde{\tau}^2 & -\tilde{\tau} & -\tilde{\tau} \\ \tilde{\tau}^2 & 1 & -\tilde{\tau} & -\tilde{\tau} \\ \tilde{\tau} & \tilde{\tau} & -1 & -\tilde{\tau}^2 \\ \tilde{\tau} & \tilde{\tau} & -\tilde{\tau}^2 & -1 \end{pmatrix}$$

$$W_{4,5} := \begin{pmatrix} 1 & \tau^2 & -\tau & -\tau \\ \tau^2 & 1 & -\tau & -\tau \\ \tau & \tau & -1 & -\tau^2 \\ \tau & \tau & -\tau^2 & -1 \end{pmatrix} \quad W_{4,6} := \begin{pmatrix} 1 & 1 & -\tilde{\tau} & -\tau \\ 1 & 1 & -\tau & -\tilde{\tau} \\ \tilde{\tau} & \tau & -1 & -1 \\ \tau & \tilde{\tau} & -1 & -1 \end{pmatrix}$$

B Kummer pairings à la Robert

This section details a concrete instantiation to computing pairings of degree 2 on (squared) Kummer surfaces using Algorithm 5.2 by Robert [52]. We assume we want to compute the Tate pairing $t_2(L_{i,j}, Q)$ with $L_{i,j} \in \mathcal{K}[2]$ and $Q \in \mathcal{K}$, which we also denote $t_{i,j}(Q)$. By Appendix A.2, we have $W_{i,j}$, the addition matrix with respect to $L_{i,j}$.

We first normalise the point $L := L_{i,j}$ by its first non-zero coefficient, whose index we denote $n_{i,j}$. We then apply $W_{i,j}$ to get $\widetilde{L}$, and set $\lambda_{i,j} = \widetilde{L}_{n_{i,j}}/\mu_{n_{i,j}}$. The pairing value $t_{i,j}(Q)$ for some $Q \in \mathcal{K}$ can then be computed by computing $D = L_{i,j} \pm Q$ as $W_{i,j} \cdot Q$ $\lambda_Q = D_{n_{i,j}}/Q_{n_{i,j}}$. This gives us Algorithm 6.

Algorithm 6 Monodromy pairing computation.

Input: A Kummer surface $\mathcal{K}$, an index (i, j) with $1 \leq i < j \leq 6$, a point $Q \in \mathcal{K}$ normalised to $n_{i,j}$ and the precomputed $\lambda_{i,j}$.

Output: The reduced 2-Tate pairing $t_{i,j}(Q) = t_2(L_{i,j}, Q)$.

1: $D \leftarrow W_{i,j} \cdot Q$
2: $\lambda_Q \leftarrow D_{n_{i,j}}/Q_{n_{i,j}}$
3: **return** `IsSquare`$(\frac{\lambda_Q}{\lambda_{i,j}})$

This approach is elegant once $W_{i,j}$ and $\lambda_{i,j}$ are computed, and only requires a matrix multiplication and a Legendre symbol. The divisions can be replaced by multiplications for improved performance, as this does not affect the final Legendre symbol.

All in all, to perform a monodromy pairing computation of degree 2 on a given Kummer surface, we require only the translation-by-$L_{i,j}$ maps $W_{i,j}$. For both the general Kummer surface as well as the squared Kummer surfaces, these are given in this work, and more generally they can be computed given the biquadratic forms of a Kummer surface.

C Algebraic Derivations

C.1 Derivation of lift $\mathcal{K}^{\mathbf{Sqr}}_{\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}} \to \mathcal{J}_{\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}}$

In this section, we briefly describe the derivation of the polynomials F_0, F_1, F_2 and G that give a more efficient map to recover u_0, u_1 and v_0^2 on $\mathcal{J}_{\lambda,\mu,\nu}$ given a point $P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, as described in Section 2.7.

Any such point $P = (X_1 : X_2 : X_3 : X_4)$ in the image of $\rho^{\mathrm{Sqr}} : \mathcal{J}_{\lambda,\mu,\nu} \to \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ is given by a scalar multiple $\omega \in \mathbb{F}_p$ of the image of some $D \in \mathcal{J}_{\lambda,\mu,\nu}$ as in Equation (4).

Hence, we have a system of equations

$$\begin{aligned}
X_1 &= \omega\mu_1 \cdot \left(u_0(w_3w_5 - u_0)(w_4 + w_6 + u_1) - v_0^2\right), \\
X_2 &= \omega\mu_2 \cdot \left(u_0(w_4w_6 - u_0)(w_3 + w_5 + u_1) - v_0^2\right), \\
X_3 &= \omega\mu_3 \cdot \left(u_0(w_3w_6 - u_0)(w_4 + w_5 + u_1) - v_0^2\right), \\
X_4 &= \omega\mu_4 \cdot \left(u_0(w_4w_5 - u_0)(w_3 + w_6 + u_1) - v_0^2\right).
\end{aligned}$$

with known values X_i, Kummer coefficients μ_i and Rosenhain invariants w_i, and unknowns u_0, u_1, v_0 and ω. Let $\tilde{X}_i = X_i/\mu_i$, then we rewrite this system as

$$\begin{aligned}
f_1 &: u_0(w_3w_5 - u_0)(w_4 + w_6 + u_1) - \tilde{X}_1/\omega = v_0^2, \\
f_2 &: u_0(w_4w_6 - u_0)(w_3 + w_5 + u_1) - \tilde{X}_2/\omega = v_0^2, \\
f_3 &: u_0(w_3w_6 - u_0)(w_4 + w_5 + u_1) - \tilde{X}_3/\omega = v_0^2, \\
f_4 &: u_0(w_4w_5 - u_0)(w_3 + w_6 + u_1) - \tilde{X}_4/\omega = v_0^2.
\end{aligned}$$

and so the differences $f_{i-j} = f_i - f_j = 0$ for all $1 \le i < j \le 4$ give us six equations. By assuming u_0^2, u_0u_1, u_0 and ω as independent linear variables, this gives us a matrix F such that $F(u_0^2, u_0u_1, u_0, \omega)^T = (f_{i-j})_{1\le i<j\le 4}$. After row-echelon reduction of F, we find simple equations

$$\begin{aligned}
u_0^2 &= h_1(\tilde{X}_i, w_i) \cdot 1/\omega, \\
u_0u_1 &= h_2(\tilde{X}_i, w_i) \cdot 1/\omega, \\
u_0 &= h_3(\tilde{X}_i, w_i) \cdot 1/\omega
\end{aligned}$$

for some polynomials h_i in the coefficients $\tilde{X}_i$ and w_i, and as we must have $(u_0^2) = (u_0)^2$, the first as the squared variable – the second as the variable squared, we get $\omega = h_3^2/h_1$. Thus, we derive equations for u_0, u_1 and ω from h_1, h_2 and h_3. Finally, by any of the f_i, we then similarly recover an equation for v_0^2. By gathering common factors, we find the polynomials F_0, F_1, F_2 and G defined in the coefficients μ_i, X_i, w_i.

The derivation in Magma code can be found in the file `v2_derive.m`.

D Constants for scaling map in $(2,2)$-isogeny compuation

Let $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$ where i is a root of $x^2+1 \in \mathbb{F}_p[x]$. For kernels $\langle L_{i,j}, L_{k,\ell}\rangle$ where

$$(i,j,k,\ell) \in \left\{ \begin{array}{l} (2,3,4,5),(1,3,4,5),(1,4,3,6),(2,4,3,6), \\ (2,3,4,6),(1,3,4,6),(1,4,3,5),(2,4,3,5) \end{array} \right\},$$

the required square roots can be extracted from the 4-torsion points lying above the kernel points. More precisely, let R, S be such that

$$[2]R = L_{i,j},\ [2]S = L_{k,\ell},$$

and let $h(P)_i$ be the i-th coordinate of $\mathtt{H}(P)$. For the sake of clarity, we suppose that $L_{i,j}$ and $L_{k,\ell}$ are of the form $(1\colon \star\colon 0\colon 0)$ and $(1\colon 0\colon \star\colon 0)$, respectively. The other cases follow similarly.

We compute the scaling value $(1/A : 1/B : 1/C : 1/D)$ using the points lying above the kernel by noting that

$$\frac{A}{B} = c_1 \frac{h(R)_1}{h(R)_2},\ \frac{A}{C} = c_2 \frac{h(S)_1}{h(S)_3},\ \text{and}\ \frac{C}{D} = c_3 \frac{h(R)_3}{h(R)_4},$$

for some constants $c_i \in \{1, -i\}$. From this we can derive $\frac{A}{B}$, $\frac{A}{C}$, and $\frac{A}{D}$ to get

$$\left(\frac{1}{A} : \frac{1}{B} : \frac{1}{C} : \frac{1}{D}\right) = \Big(h(R)_2 h(R)_4 h(S)_3 \colon c_1 h(R)_1 h(R)_4 h(S)_3 \colon \\ : c_2 h(R)_2 h(R)_4 h(S)_1 \colon c_2 c_3 h(R)_2 h(R)_3 h(S)_1\Big).$$

The constants for each kernel are then given as follows:

- If $(i,j,k,\ell) = (2,3,4,5)$ or $(1,4,3,6)$, then $(c_1, c_2, c_2c_3) = (1,1,1)$.
- If $(i,j,k,\ell) = (1,3,4,5)$ or $(2,4,3,6)$, then $(c_1, c_2, c_2c_3) = (1,-i,-i)$.
- If $(i,j,k,\ell) = (2,3,4,6)$ or $(1,4,3,5)$, then $(c_1, c_2, c_2c_3) = (-i,1,-i)$.
- If $(i,j,k,\ell) = (1,3,4,6)$ or $(2,4,3,5)$, then $(c_1, c_2, c_2c_3) = (-i,-i,1)$.