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

Maria Corte-Real Santos[1,2] and Krijn Reijnders[3,4]

[1] University College London, United Kingdom
[2] École Normale Supérieure de Lyon, France
[3] Radboud University Nijmegen, Netherlands
[4] COSIC, KU Leuven, Belgium

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 hyper-elliptic 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, now defined over $\mathbb{F}_p$ rather than $\mathbb{F}_{p^2}$.

Keywords: post-quantum cryptography · isogenies · Kummer surface · 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. 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 [BDK+18] and the digital signature schemes Dilithium [DKL+18], Falcon [FHK+18], and SPHICNS+ [BHK+19]. 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 [NIS22].

Isogeny-based cryptography offers an answer to these problems. SQIsign [DKL+20; DLLW23; AAA+25] 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.

E-mail: maria.corte_real_santos@ens-lyon.fr (Maria Corte-Real Santos), crypto.krijn@gmail.com (Krijn Reijnders)

Author list in alphabetical order; see https://www.ams.org/profession/leaders/CultureStatement04.pdf. The first author has been supported in part by UK EPSRC grant EP/S022503/1 and by the European Research Council (ERC) under grant No. 101116169 (AGATHA CRYPTY). The second author was supported in part by the ERC under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT - No. 101020788), by the Research Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VOEWICS02.

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2025-12-18.

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 [DLRW24], a new scheme that offers much more competitive signing times and improved security reductions, at the cost of verification speed. Second, AprèsSQI [CEMR24], a variant of SQIsign optimised for verification speed, with additional trade-offs between verification time and signature size.

Recent works [BDF+24; NOC+24; DF24] 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 [CD23; MMP+23; Rob23a] and in particular using fast isogeny formulas derived from theta structures [DMPR24].

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 such pairings in the context of isogeny-based cryptography by developing a new tool, which we call profiles. Using pairings and profiles, we derive several algorithms including CheckOrigin, PointDifference and PointCompression. We further describe Scholten's construction [Sch03] which allows us to associate 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, and, similarly, a specific $(2,2)$-isogeny between such Kummer surfaces to any 2-isogeny between the corresponding elliptic curves, due to Costello [Cos18]. These special $(2,2)$-isogenies are a natural subset of regular $(2,2)$-isogenies, which we describe using the framework of theta structures [DMPR24].

With these algorithms and techniques, isogeny-based primitives using supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ can be modified to work with *elliptic* Kummer surfaces over $\mathbb{F}_p$ instead. Additionally, we can potentially exploit vectorisation for the core Kummer arithmetic, which is very efficient on modern architectures [BCLS14]. As a showcase of these tools and techniques, we show that the original SQIsign verification [DKL+20; DLLW23; AAA+25], or its AprèsSQI variant [CEMR24], can be performed completely over Kummer surfaces defined over $\mathbb{F}_p$. Using Scholten's construction [Sch03], this turns the one-dimensional response isogeny into a two-dimensional isogeny between products of elliptic curves. The paper is structured as follows.

- 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 and defines the Tate profile, which allows us to generalise previous results [CEMR24, Thm. 2] to the dimension 2 (and higher) case in Theorem 1.
- 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 [Cos18]. We show that the codomain, and its Rosenhain invariants,

cost[1] 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 [DMPR24] 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 in Python and `MAGMA` containing:

- An optimised Python implementation of compressed and uncompressed SQIsign verification on Kummer surfaces, used for benchmarking. We use SageMath [The21] for the Jacobian arithmetic required in PointCompression, an algorithm performed only in signing.
- All the algorithms and maps in this article are implemented in `MAGMA` [BCP97], 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.

Our code is available under the MIT license in the following repository:

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

Related work. Kummer surfaces of genus-2 Jacobians were first introduced to cryptography by Chudnovsky and Chudnovsky [CC86], who gave a variant of Lenstra's ECM factoring algorithm. Gaudry [Gau07] then proposed 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 [BCLS14; BCHL16; RSSB16] and signature schemes [RSSB16; RS17] give significant improvements over elliptic curves in many contexts, in particular with new techniques for efficient vectorisation of Kummer surface computations, leading to new speed records for high-security constant-time (hyper)elliptic curve Diffie–Hellman [BCLS14]. In parallel to this, Lubicz and Robert [LR16] developed algorithms for efficient arithmetic on Kummer surfaces using the theory of theta functions of level 2, and efficient algorithms for pairing computations [LR10], recently improved by Robert [Rob24a]

Orthogonally, in a work by Costello [Cos18], Kummer surfaces were introduced to isogeny-based cryptography in the context of SIDH. Costello extends Scholten's construction to associate a $(2,2)$-isogeny over $\mathbb{F}_p$ to any 2-isogeny between elliptic curves, for which we provide a new context in this work. Such $(2,2)$-isogenies are rather special, and we therefore refer to these as *elliptic* isogenies. Kummer surfaces are implicitly used in algorithms to compute $(2,2)$-isogenies using theta structures of level 2 [DMPR24], which arise in recent variants of SQIsign [NOC$^+$24; BDF$^+$24; DF24], and elliptic isogenies can be understood in this framework of theta structures. However, elliptic isogenies can be computed more efficiently than general (2,2)-isogenies.

In recent years, new variants of SQIsign have emerged [NOC$^+$24; BDF$^+$24; DF24] 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

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

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. We thank Sabrina Kunzweiler for giving us a better method to recover the full divisor in Section 4.1.

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 Jacobian $\mathcal{J}_C$ 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 $\mathcal{C}_{\lambda,\mu,\nu}$ in Rosenhain form then it also admits a *canonical* and *squared* Kummer surface. These two other Kummer surfaces are closely related; we describe them in Sections 2.3 and 2.4, and give maps $\mathcal{K}_\mathcal{C}^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}}$ and $\mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{Sqr}}$. 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.[2]

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

[2] A reader overwhelmed by all these Kummer surfaces, maps, and constructions may keep Figure 1 at their side, which details the main construction and maps required for this work.

- the general Kummer surface by $\mathcal{K}_\alpha^{\text{gen}}$,
- 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. Whenever we use pairs w_i, w_j, we always assume $i < j$. In particular, only i can be 1.

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 [LO98; Bro93; CDS20]. We refer to a hyperelliptic curve as superspecial when its Jacobian is superspecial. Superspecial Jacobians can be defined over $\mathbb{F}_{p^2}$ and have $\mathbb{F}_{p^2}$-rational 2-torsion. This makes them efficient, as their Kummer surfaces are then also defined over $\mathbb{F}_{p^2}$. In later sections, we may require $\mathbb{F}_p$-rational 2-torsion to ensure an $\mathbb{F}_p$-rational Kummer surface.

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 [CF96, §3] to compute the corresponding general Kummer surface. Let

$$\mathcal{C} : y^2 : c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5 + c_6 x^6,\ c_i \in \Bbbk,$$

where c_6 can equal 0 if $\mathcal{C}$ is in the degree-5 model.[3] 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

$$\mathcal{K}^{\text{gen}} : K_2 X_4^2 + K_1 X_4 + K_0 = 0,$$

where

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

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

[3] Sometimes called the odd degree model.

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

$$X_1 := 1, \quad X_2 := x_1 + x_2, \quad X_3 := x_1 x_2, \quad X_4 := \frac{F(x_1, x_2) - 2y_1 y_2}{(x_1 - x_2)^2},$$

with

$$\begin{aligned} F(x_1, x_2) = 2c_0 + c_1(x_1 + x_2) + 2c_2 x_1 x_2 + c_3(x_1 + x_2)x_1 x_2 + 2c_4(x_1 x_2)^2 \\ + c_5(x_1 + x_2)(x_1 x_2)^2 + 2c_6(x_1 x_2)^3. \end{aligned}$$

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[4] $\langle x^2 + a_1 x + a_0, b_1 x + b_0 \rangle \in \mathcal{J}_{\mathcal{C}}$, we can 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.

$$\tilde{\rho} : \langle x^2 + a_1 x + a_0, b_1 x + b_0 \rangle \mapsto (X_1 \colon X_2 \colon X_3 \colon X_4)$$

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 $w_i, w_j \neq \infty$. When ∞ 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 fifteen points $L_{i,j}$ of order 2 on $\mathcal{K}$ given by

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

Addition by 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 this translation-by-$L_{i,j}$, i.e., the map $P \mapsto P + L_{i,j}$, by 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 [CF96], and we provide a compact presentation in Appendix A.

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

2.2 Rosenhain form of a hyperelliptic curve

To construct the canonical and squared Kummer surfaces, we require 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

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

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 of 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 [CC86] in 1986. These more efficient Kummer surfaces come in two forms: the *canonical* Kummer surface, as described by Gaudry [Gau07]; and the closely related *squared* Kummer surface, described by Bernstein [Ber06].

2.3 Canonical Kummer surface

Following Gaudry [Gau07], 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.

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

where

$$\begin{aligned} E &:= \frac{(16ABCD)^2\cdot abcd}{(a^2d^2-b^2c^2)(a^2c^2-b^2d^2)(a^2b^2-c^2d^2)}, \text{ and}\\ F &:= \frac{a^4-b^4-c^4+d^4}{a^2d^2-b^2c^2}, \quad G := \frac{a^4-b^4+c^4-d^4}{a^2c^2-b^2d^2}, \quad H := \frac{a^4+b^4-c^4-d^4}{a^2b^2-c^2d^2}. \end{aligned}$$

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 arithmetic on these surfaces to Gaudry [Gau07].

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

2.4 Squared Kummer surface

The squared Kummer surface has been the subject of interest in hyperelliptic curve cryptography [Ber06; BCLS14; RSSB16; BCHL16] as it boasts the fastest arithmetic when one can stay on a single surface, as described by Bernstein [Ber06].

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

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

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 $\mathbf{o} \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ is $\mathbf{o} = (\mu_1 : \mu_2 : \mu_3 : \mu_4)$. As given by Bos, Costello, Hisil, and Lauter [BCHL16], we can derive the constants $\mu_1, \mu_2, \mu_3, \mu_4$ from the Rosenhain invariants $\lambda, \mu, \nu \in \Bbbk$ as

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

Map from the Jacobian to the Squared Kummer Surface. The map $\rho^{\mathrm{Sqr}} : \mathcal{J}_{\lambda,\mu,\nu} \to \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ is given by [Cos11; CCS17]:

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

where

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

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 $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$. 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 2-torsion on $\mathcal{J}_{\lambda,\mu,\nu}$. 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 $\lambda, \mu, \nu \in \Bbbk$.[5] 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

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

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 [Gau07] and [Gau07, §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 $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$. 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. 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

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

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 [Smi05, § 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 [Igu60, p. 620] is a 'nice' subspace of the general moduli space cut out by the equation $\nu = \lambda\mu$.

Construction of the elliptic Kummer surface $\mathcal{K}^{\mathbf{Sqr}}_{\lambda,\mu,\lambda\mu}$. 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 $\mathcal{K}^{\mathbf{Sqr}}_{\lambda,\mu,\lambda\mu}$. 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:

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

Note the additional symmetry between the left and right column, which positively impacts the efficiency of arithmetic on the elliptic Kummer surface.

Addition by points of order 2 on $\mathcal{K}^{\mathbf{Sqr}}_{\lambda,\mu,\lambda\mu}$. 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.

2.6 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 well-suited 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 `S`, the Hadamard involution `H`, the scaling map $\mathtt{C}_P$, and the inversion map `Inv`, defined as follows:

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

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

Curve arithmetic. The four basic morphisms `S`, `H`, `C` and `Inv` are enough to define the curve operations doubling `xDBL` : $P \mapsto 2P$, differential addition `xADD` : $P, Q, P-Q \mapsto P+Q$, scalar multiplication `xMUL` : $P \mapsto [n]P$ as described in [CCS17, Appendix A]. We can also derive a three-point ladder $P, Q, P - Q, s \mapsto P + sQ$.

2.7 Maps between Kummer surfaces

We can derive a map between $\mathcal{K}_{\mathcal{C}}^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}}$ from an isomorphism between the curve $\mathcal{C}$ and its Rosenhain form $\mathcal{C}_{\lambda,\mu,\nu}$. 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 [Ros95]. It is given by five values $(a, b, c, d, e) \in \Bbbk$, namely

$$\kappa : \mathcal{C} \to \mathcal{C}_{\lambda,\mu,\nu}, \quad (x, y) \mapsto \left(\frac{ax + b}{cx + d}, \frac{ey}{(cx + d)^3} \right)$$

as described by Costello [Cos18, §2]. This map κ induces a map $\kappa_* : \mathcal{J} \to \mathcal{J}_{\lambda,\mu,\nu}$ between their Jacobians [Cos18, §3]. From κ_*, we construct the induced map $\kappa_{**} : \mathcal{K}^{\text{gen}} \to \mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}}$ 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 by the first coordinate. We have:

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

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 of the domain and K_1', K_2', K_3' of 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 $\mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}}$ and $\mathcal{K}_{\lambda,\mu,\nu}^{\text{Sqr}}$ be the general and squared Kummer surface associated to the same hyperelliptic curve $\mathcal{C}_{\lambda,\mu,\nu}$. The isomorphism between $\mathcal{K}_{\lambda,\mu,\nu}^{\text{gen}}$ and $\mathcal{K}_{\lambda,\mu,\nu}^{\text{Sqr}}$ is given by Chung, Costello, and Smith [CCS17] as a linear map $\mathbf{M}$, by interpolating image points under both $\rho_{\lambda,\mu,\nu}^*$ and ρ^{Sqr} of divisors of $D \in \mathcal{J}_{\lambda,\mu,\nu}$.

2.8 Scholten's construction

In 2003, Scholten [Sch03] 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 [Sch03] 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 [Cos18, Prop. 1]. To the Jacobian $\mathcal{J}_\alpha$ of $\mathcal{C}_\alpha$, we can associate a general Kummer surface $\mathcal{K}^{\mathrm{gen}}_\alpha$. As we prefer to work with the squared Kummer model, the next proposition follows Costello [Cos18, §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 [LO98; Bro93; CDS20].

Proposition 1 (§5, [Cos18])**.** *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 $\beta^2 = (\alpha^2 - 1)/\alpha$. Define four values $\zeta_i \in \mathbb{F}_p$ by*

$$\begin{aligned} \zeta_1 &= \beta_0\gamma_1 + \beta_1\gamma_0, \quad & \zeta_2 &= \beta_0\gamma_0 + \beta_1\gamma_1, \\ \zeta_3 &= \beta_0\gamma_1 - \beta_1\gamma_0, \quad & \zeta_4 &= \beta_0\gamma_0 - \beta_1\gamma_1. \end{aligned}$$

Then $\mathcal{C}_\alpha$ is isomorphic over $\mathbb{F}_p$ to $\mathcal{C}_{\lambda,\mu,\nu}$ where

$$\lambda = -\frac{\zeta_1 \cdot \zeta_2}{\zeta_3 \cdot \zeta_4}, \quad \mu = \frac{\zeta_2 \cdot \zeta_4}{\zeta_1 \cdot \zeta_3}, \quad \nu = -\left(\frac{\zeta_2}{\zeta_3}\right)^2.$$

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 $\kappa_{**} : \mathcal{K}^{\mathrm{gen}}_\alpha \to \mathcal{K}^{\mathrm{gen}}_{\lambda,\mu,\nu}$. Composing this with $\mathbf{M} : \mathcal{K}^{\mathrm{gen}}_{\lambda,\mu,\nu} \to \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, we obtain a map $\mathcal{K}^{\mathrm{gen}}_\alpha \to \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$. The theta constants can be computed using Equation (3) combined with γ_0 and γ_1 as

$$\mu_1 = \sqrt{\lambda} \cdot \left(\frac{\gamma_0^2 - \gamma_1^2}{\gamma_0^2 + \gamma_1^2}\right), \quad \mu_2 = \mu_1/\lambda, \quad \mu_3 = \mu_4 = 1. \tag{5}$$

We refer to this elliptic Kummer surface $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\lambda\mu}$ as the "Kummer surface associated to E_α" through Scholten's construction, and denote $\mathcal{K}_\alpha := \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\lambda\mu}$ to make the association explicit.

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 [BL14]. We use the formulation by Costello [Cos18, §3], which 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 $\rho^{\mathrm{Sqr}} \circ \kappa_\star$ from Section 2.7 to get $\bar{\eta} := \rho^{\mathrm{Sqr}} \circ \kappa_\star \circ \eta$, we can map points $P \in E_\alpha(\mathbb{F}_{p^2})$ to $\bar{\eta}(P) \in \mathcal{K}_\alpha(\mathbb{F}_p)$. The *trace map* $\mathcal{T}$ allows us to map points from $\mathcal{J}(\mathbb{F}_{p^2}) \to \mathcal{J}(\mathbb{F}_p)$.

In Figure 1, we provide a diagram summarising the maps introduced in Sections 2.7 and 2.8.

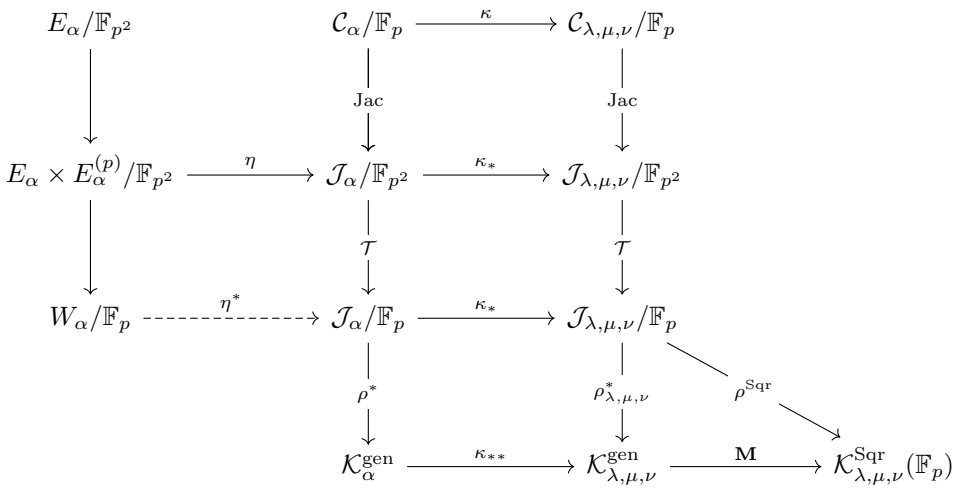


Figure 1: The maps involved when finding the (squared) Kummer surface defined over $\mathbb{F}_p$ corresponding to an elliptic curve defined over $\mathbb{F}_{p^2}$, including the maps between Kummers described in this section.

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.

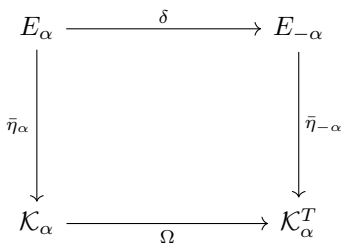
Definition 2. The *elliptic twist* of a squared Kummer surface $\mathcal{K}^{\mathrm{Sqr}}$, denoted $\mathcal{K}^T$ is defined as the surface with theta coordinates $(-\mu_1, -\mu_2, \mu_3, \mu_4)$, where μ_i are the theta coordinates of $\mathcal{K}^{\mathrm{Sqr}}$.

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 [Cos18, 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 (5), we get that $\mu_1 \mapsto -\mu_1$ and $\mu_2 \mapsto -\mu_2$.[6] 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 $(X_1 \colon X_2 \colon X_3 \colon X_4)$ 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 $(-X_1 \colon -X_2 \colon X_3 \colon X_4)$. □

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.[7] 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, i.e., there exists a (2,2)-isogeny

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

Such an isogeny is often called a *splitting* and its dual $\hat{\varphi}_\alpha$ is the *gluing* of E_α and $E_\alpha^{(p)}$ along 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 $\mathbb{F}_p$, which would have implications on the quantum security. We can frame Scholten's construction in the language of gluing and splitting, visualised in Figure 2, where $\mathcal{T}$ is the trace map from $\mathcal{J}_\alpha(\mathbb{F}_{p^2})$ to $\mathcal{J}_\alpha(\mathbb{F}_p)$ and $\mathcal{T}_*$ the induced map on their Kummer surfaces.

[6] For a Magma version of this proof, see `TwistProof.m`.
[7] See `TwistProof.m` for a proof of this in `MAGMA`.

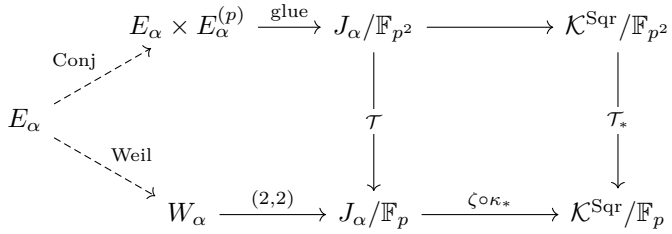


Figure 2: A clearer picture of the Weil Restriction maps

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, $E_\alpha \times E_\alpha^{(p)}$ equipped with the product polarisation is $(2,2)$-isogenous to the extension of this $\mathcal{J}_\alpha$ over $\mathbb{F}_{p^2}$.*

Proof. Weil [Wei61, §1.3] shows that the Weil restriction is isomorphic to $E_\alpha \times E_\alpha^{(p)}$ over $\mathbb{F}_{p^2}$. By applying the gluing formulas from [CD23, Prop. 1], we obtain a hyperelliptic curve $\mathcal{C}$. We find that $\mathcal{C}$ and $\mathcal{C}_\alpha$ are isomorphic by comparing the Igusa-Clebsch invariants using a computer algebra system. This shows that the (2,2)-isogeny $E_\alpha \times E_\alpha^{(p)} \to \mathcal{J}_\alpha$ is the gluing of E_α and $E_\alpha^{(p)}$ over the two-torsion. □

3 Pairings on Kummer surfaces

Pairings on abelian varieties have proven to be essential in the construction and cryptanalysis of many cryptographic primitives [HSSI99; Jou04; BKLS02]. Most relevant to this article is their use in isogeny-based cryptography, in particular recent work [CJL+17; LWXZ24; CEMR24] 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. We use pairings to decompose Jacobians in genus 2 in Section 3.1 which generalises [CEMR24, Thm. 2] to Kummer surfaces, and specialise to elliptic Kummer surfaces in Section 3.2. We furthermore discuss the concrete computation of degree-2 pairings in Section 3.3.[8]

Key to these results are *generalised Tate pairings*: As shown by Bruin [Bru11], 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

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

known as the *generalised φ-Tate pairing*. This pairing has been studied in the context of cryptography [CHM+23] for elliptic curves and for abelian varieties [Rob23b]. As Robert [Rob23b] notes, perfectness implies that we can precisely identify $(\operatorname{Im} \hat{\varphi})(\mathbb{F}_q)$ using $\ker \varphi$ and this pairing, in the following sense.

Lemma 5 (Cor. 5.2, [Rob23b])**.** *Let t_φ denote the (reduced) φ-Tate pairing. Then*

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

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

[8] Although our target application is compression of SQIsign signatures, the possible applications of generalised Tate pairings to study the image of isogenies are much more general than SQIsign, or even Kummer surfaces. One example is pre-quantum elliptic curve subgroup membership testing [Kos23].

Definition 3. Let t_φ denote the (reduced) φ-Tate pairing and let $Q \in A(\mathbb{F}_q)$. The *profile* of Q is the array of evaluations of $t_\varphi(P, Q)$ in all points $P \in (\ker \varphi)(\mathbb{F}_q)$. 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 $(A/\operatorname{Im}\hat{\varphi})(\mathbb{F}_q)$ it lies. By bilinearity of the Tate pairing, the profile of $Q + Q'$ is the pointwise multiplication of their profiles. Furthermore, any basis of $(\ker \varphi)(\mathbb{F}_q)$ 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 $(\ker \varphi)(\mathbb{F}_q)$.

Example 1. One may use the above decomposition to rephrase the technique to sample points on Montgomery curves whose order is divisible by 2^f by sampling a non-square x-coordinate x_P. Namely, in terms of generalised Tate pairings, this requires us to sample points in the coset $(E \setminus \operatorname{Im}\hat{\varphi})(\mathbb{F}_q)$, 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. This technique is used in many isogeny-based schemes, such as SIDH/SIKE, CSIDH and SQIsign.

The group structure of the rational cokernel. 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 profiles, 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*

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

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 $(\operatorname{Im}\hat{\varphi})(\mathbb{F}_q)$ [Rob23b, 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. Intuitively, this lemma shows that profiles of points are coordinates of Q in $(\operatorname{coker}\hat{\varphi})(\mathbb{F}_q)$, with respect to a given basis of $(\ker \varphi)(\mathbb{F}_q)$.

3.1 Decomposing Jacobians 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 $\mathcal{J}_\mathcal{C}/[2]\mathcal{J}_\mathcal{C}$. 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 (rather than an extension field).

Theorem 1 gives a generalisation of [Hus04] 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 [CF96, Ch. 10].

Theorem 1. *Let $P \in \mathcal{K}_\mathcal{C}$ and let $\{L_{i,j}\}_{1\leq i<j\leq 6}$ 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, namely*

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

Proof. This follows from Lemma 5. 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

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

Thus, Theorem 1 can more succinctly 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 $1 \leq i \leq 4$.*

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 $1 \leq i \leq 4$.

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.

Remark 3. The results above show that the higher-dimensional case is more nuanced than the dimension 1 case. In dimension 1, one often only needs a single quadratic reciprocity check on some value related to x_P for $P \in E(\mathbb{F}_q)$ to determine the behaviour of P with respect to the 2^f-torsion $E[2^f]$. See, for example, [ZJP+19, Thm. 1], [CEMR24, Thm. 1], and [DEF+25, App. D]. From the viewpoint of profiles, we translate that as follows: a single quadratic reciprocity check on some value related to x_P corresponds to a degree-2 generalized Tate pairing $e_\varphi(Q, P)$, where $Q \in E[2]$ and $\ker\varphi = \langle Q \rangle$. Hence, we can position P with respect to the 2^f-torsion *above* Q. For example, when $Q = (0, 0)$, the non-triviality of $e_\varphi(Q, P)$ equals x_P being non-square, and implies that the order of P is divisible by 2^f and not above $(0, 0)$.

In dimension g, the kernel of an $(n, \ldots, n)$-isogeny has rank g in the n-torsion group, and we will therefore need at least g pairings to position a point P, in general. This deviates from the dimension 1 scenario in two important ways. First, we may still use a single pairing to determine results on the order of P, as shown in Corollary 1, however, a single pairing is no longer equivalent to a simple check on quadratic reciprocity. Second, accurately positioning a point P with respect to an $(n, \ldots, n)$-isogeny requires g pairings instead of 1 pairing.

3.2 Decomposing Jacobians isogenous to Weil restrictions

Consider the superspecial Jacobian $\mathcal{J}_\alpha$ that is $(2, 2)$-isogenous to the Weil restriction $W(E_\alpha)$ 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. As $W(E_\alpha)(\mathbb{F}_p) \cong \mathbb{Z}_{p+1} \times \mathbb{Z}_{p+1}$, and $\mathcal{J}_\alpha$ is $(2, 2)$-isogenous and has rational 2 torsion, the resulting group structure is

$$\mathcal{J}_\alpha(\mathbb{F}_p) \cong \mathbb{Z}_m \times \mathbb{Z}_m \times \mathbb{Z}_2 \times \mathbb{Z}_2.$$

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, i.e., have maximal power-of-two torsion. This is in general

useful for isogeny-based cryptography, and we rely heavily on this fact in later sections. The core idea is to identify the coset of Q in $\mathcal{J}_\mathcal{C}/[2]\mathcal{J}_\mathcal{C}$ using the *profile* $t_{\ker[2]}$ of Q, as defined in Definition 3. We present this theory for Jacobians isogenous to the Weil restriction of an elliptic curves, as the group structure of such Jacobians gives us a good example of the theory. We emphasize, however, that this can be more generally applied to any Jacobian, abelian variety, or Kummer variety.

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 1 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 $aP_1 + bP_2 + cP_3 + dP_4 + [2]\mathcal{J}_\mathcal{C}$. Given $\mathcal{B} = (P_1, \ldots, P_4)$, we compute $t_{\mathcal{B},[2]}(a, b, c, d)$ simply as $t_{\ker[2]}(Q)$ for $Q = aP_1 + bP_2 + cP_3 + dP_4$. 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 1 and generalizing [CEMR24, Thm. 2].

Theorem 2. *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 $(\operatorname{coker}\varphi)(\mathbb{F}_q)$ using profiles.

3.3 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 [GHV07]. 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

$$\langle D_{i,j}, D_P\rangle_2 = (w_i - x_1)(w_i - x_2)(w_j - x_1)(w_j - x_2). \tag{6}$$

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

$$\langle L_{i,j}, P\rangle_2 = l_3^2 + k_3^2 - (k_2 l_2 l_3 + k_2 k_3 l_2) + k_2^2 l_3 + k_3 l_2^2 - 2 \cdot k_3 l_3. \tag{7}$$

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 $t_2(L_{i,j}, -)$.[9] Whenever $L_{i,j}$ and P are not coprime, we add some element $L_{i',j'} \in \mathcal{K}^{\text{gen}}[2]$ to P to obtain the required coprimality.[10]

On Squared Kummers. On squared Kummers, there are three distinct methods to compute the degree-2 Tate pairing.

- **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 $\mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$, 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 [Gau07], to compute all possible Tate pairings of degree 2. Due to the need for additional theta functions, the above approach is neither efficient nor elegant.
- **Monodromy approach by Robert.** An alternative approach to compute such a Tate pairing on Kummer surfaces is sketched by Lubicz and Robert [LR10]. To compute $t_2(L_{i,j}, Q)$, we require the matrix $W_{i,j}$ representing the translation-by-$L_{i,j}$

[9] Alternatively, one can find suitable $L_{i',j'}$ to compute $t_2(L_{i,j}, P)$ 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.

[10] 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 $W_{i,j}$.

map. For elliptic Kummers, these are given in Appendix A. These computations are explained more generally by Robert [Rob24a]; we apply their Algorithm 5.2 to Kummer surfaces of genus-2 hyperelliptic curves for a fixed pairing $\langle L_{i,j}, - \rangle_2$. See Appendix B for a detailed explanation.

- **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 (9). 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 $t_2(L_{i,j}, P)$ 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 $t_2(L_{i,j}, P)$ 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 4.5, we require the following tools.

- A more efficient partial inverse map from the Kummer to the Jacobian, which enables many of the subsequent algorithms.
- An algorithm CheckOrigin, to derive if a point $P \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ originates from the Jacobian $\mathcal{J}_{\lambda,\mu,\nu}$ or its twist $\mathcal{J}^T_{\lambda,\mu,\nu}$ over $\mathbb{F}_p$. We use this to verify that two points $P, Q \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ both arise from the same Jacobian by $\mathsf{CheckOrigin}(\mathcal{K}, \mathbf{o}, P) = \mathsf{CheckOrigin}(\mathcal{K}, \mathbf{o}, Q)$, which we require in later algorithms.
- An algorithm PointDifference, which takes as input $P, Q \in \mathcal{K}$ and derives $P \pm Q \in \mathcal{K}$. This enables the computation of $P + sQ$ in later sections.
- An algorithm to efficiently sample points with certain profiles, which ensures efficient deterministic basis sampling on $\mathcal{K}$, specifically for $\mathcal{K}[2^f]$.
- An algorithm PointCompression, which enables us to compress a point $K \in \mathcal{K}[2^f]$ to f bits, by writing K as $P + sQ$ for a deterministic basis P, Q and $s \in [1..2^f]$.[11] For the $(2^f, 2^f)$-isogenies defined in Section 5, we can thus compress these to f bits.

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 feasible and practical.

4.1 Map from Kummer to Jacobian

For several applications in this work, 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 [Gau07], 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.

[11] To compute K given P, Q, s, 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.

Using algebraic tools, detailed in Appendix C, we find a more elegant and efficient way to map a point $(X_1 : X_2 : X_3 : X_4) \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$ to a divisor with Mumford coordinates $\langle u(x), v(x)\rangle = \langle x^2 + u_1 x + u_0, v_1 x + v_0\rangle \in \mathcal{J}_{\lambda,\mu,\nu}$, using three polynomials $F_0, F_1, F_2 \in \Bbbk[X]$ given by

$$\begin{aligned} F_0 =& (w_4 - w_6)\tilde{X}_1 + (w_3 - w_5)\tilde{X}_2 - (w_4 - w_5)\tilde{X}_3 - (w_3 - w_6)\tilde{X}_4, \\ F_1 =& (w_3 - w_5)w_4 w_6 \tilde{X}_1 + (w_4 - w_6)w_3 w_5 \tilde{X}_2 \\ & - (w_3 - w_6)w_4 w_5 \tilde{X}_3 - (w_4 - w_5)w_3 w_6 \tilde{X}_4, \\ F_2 =& -(\tilde{X}_1 + \tilde{X}_2 - \tilde{X}_3 - \tilde{X}_4)(w_3 w_4 - w_5 w_6). \end{aligned} \tag{8}$$

We then recover u_0 and u_1 as

$$u_0 = F_1(\tilde{X})/F_0(\tilde{X}), \quad u_1 = F_2(\tilde{X})/F_0(\tilde{X}), \tag{9}$$

where $\tilde{X}_i = X_i/\mu_i$. This recovery works regardless of the projective representation of P. Explicitly, 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 allows us to recover ω.

$$\begin{aligned} G &= -(w_3 - w_4)(w_5 - w_6)(w_3 w_4 - w_5 w_6)F_1(\tilde{X}), \\ \omega &= G(\tilde{X})/F_0^2(\tilde{X}). \end{aligned} \tag{10}$$

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

$$v_0^2 = u_0(w_3 w_5 - u_0)(w_4 + w_6 + u_1) - \omega \tilde{X}_1. \tag{11}$$

For a number of applications, having u_0, u_1, v_0^2 is already sufficient. However, to fully recover the divisor, we need to compute the corresponding v_0, v_1. Let $\mathcal{C}_{\lambda,\mu,\nu} \colon y^2 = f(x)$ be the curve corresponding to $\mathcal{J}_{\lambda,\mu,\nu}$. Let $q(x)$ and $r(x)$ be the quotient and remainder of $\big(f(x) - v(x)^2\big)/u(x)$, respectively. Then, $\deg(r) \leq 1$; write $r(x) = r_1 x + r_0$. If $\langle u(x), v(x)\rangle$ is a Mumford divisor, we have $r_1 = r_0 = 0$. This means that by computing the remainder $r(x)$ symbolically, we get the following relations between the coefficients of $v(x)$ and $u(x)$:

$$\begin{aligned} v_1^2 &= (v_0^2 - g_1)/u_0 \\ v_0 &= (u_1 v_1^2 + g_2)/(2v_1) \end{aligned}$$

where $g_1, g_2 \in \Bbbk[\lambda, \mu, \nu, u_0, u_1]$ and can thus be efficiently computed. We then compute v_1 as a (choice of) square root of $(v_0^2 - g_1)/u_0$, and compute v_0 using the second equation. We choose the square root so that v_0^2 matches the value that we have already computed.

4.2 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 $\mathcal{J}^T_{\lambda,\mu,\nu}(\mathbb{F}_p)$.[12] 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 $P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}(\mathbb{F}_p)$.

Recognizing if a random point $P \in \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}(\mathbb{F}_p)$ 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

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

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 $(\rho^{\mathrm{Sqr}})^{-1} : \mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu} \to \mathcal{J}_{\lambda,\mu,\nu}$ described by polynomials F_0, F_1, F_2 over $\mathbb{F}_p$ (see Equations (8) and (9)) gives us the criteria we need to recognise the origin of a points on $\mathcal{K}^{\mathrm{Sqr}}_{\lambda,\mu,\nu}$, 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 $\mathbb{F}_p$:*

$$z = u_0(w_3w_4 - u_0)(w_4 + w_6 + u_1) - \omega X_1/\mu_1 \in \mathbb{F}_p$$

with u_0, u_1 as given in Equation (9), and ω in Equation (10). 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 (11)), 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 $p \equiv 3 \mod 4$. □

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: Set $\widetilde{X}_i \leftarrow X_i/\mu_i$ for $1 \le i \le 4$.
2: Set $u_0 \leftarrow F_1(\widetilde{X})/F_0(\widetilde{X})$, $u_1 \leftarrow F_2(\widetilde{X})/F_0(\widetilde{X})$, and $\omega \leftarrow G(\widetilde{X})/F_0^2(\widetilde{X})$.
3: Compute $z \leftarrow v_0^2 \leftarrow u_0(w_3w_5 - u_0)(w_4 + w_6 + u_1) - \omega\tilde{X}_1$.
4: **return** `IsSquare`(z)

Remark 4. CheckOrigin is much simpler than the general maps given by theta functions, described by Gaudry [Gau07], to compute the origin. Previous works [BCHL16] 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 [BCHL16].

4.3 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 [RS17, Prop.

4], using the biquadratic forms B_{ij} associated to the Kummer surface with the property

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

Namely, 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

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

where $f_{i,j}(X_i, X_j) = 0$ 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, for example, $f_{1,3}$ and $f_{2,3}$, which corresponds to the root X_3. In our implementation, we specialise the inversion-free Euclid algorithm from [CCS22] to polynomials of degree-2 to get a precise cost of 11**M** and 3**a** for this 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**
  $A_{i,j} \leftarrow B_{i,j}^{\mathcal{K}}(P, Q)$
for $1 \leq i, j \leq 4$ with $i < j$ **do**
  $f_{i,j} \leftarrow A_{j,j} x_i^2 - 2A_{i,j} x_i x_j + A_{i,i} x_j^2$
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 [RS17], 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 $\mathtt{H}(R) = P \pm Q$.

4.4 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 Theorem 2, we may assume we have 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 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 [PH78] to obtain s. We then compute both $D, S = P \pm Q, P \mp Q$ and recompute both $K_D = \texttt{3ptLadder}(P, Q, D, s)$ and $K_S = \texttt{3ptLadder}(P, Q, S, s)$. 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 3 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 $K = \texttt{3ptLadder}(P, Q, D_b, s)$.

$D_P \leftarrow \rho^{-\mathrm{Sqr}}(P)$, $D_Q \leftarrow \rho^{-\mathrm{Sqr}}(Q)$, $D_K \leftarrow \rho^{-\mathrm{Sqr}}(K)$.
$s \leftarrow \texttt{DiscLog}(K, P, Q)$
$D_0, D_1 \leftarrow$ PointDifference$(P, Q, \texttt{both} = \texttt{true})$
$K_0 \leftarrow \texttt{3ptLadder}(P, Q, D_0, s)$
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 3ptLadder. We then recompute K by deterministically sampling P, Q, recomputing $P - Q$ as either D or S, and deriving $K = \texttt{3ptLadder}(P, Q, P - Q, s)$, as shown in algorithm PointDecompression.

Algorithm 4 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 $\mathcal{K}$.

$D_0, D_1 \leftarrow$ PointDifference$(P, Q, \texttt{both} = \texttt{true})$
return $\texttt{3ptLadder}(P, Q, D_b, s)$

4.5 Point sampling with certain profile

To use PointCompression, we require deterministically sampled points P and 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, which can be guaranteed by Theorem 2 as long as P and Q have the right profile. We describe two methods to deterministically sample points with the right profile.

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 $t_{\ker[2]}(D_P) = (t_2(D_{k,m}, D_P))_{1 \le k,m \le 6}$ is completely determined by products of the Legendre symbols of $(x_1 - w_j)$ and $(w_i - w_j)$ for $1 \le j \le 6$. By Equation (6),

$$\langle D_{k,m}, D_P \rangle_2 = (w_i - w_k)(w_i - w_m)(x_1 - w_k)(x_1 - w_m). \tag{12}$$

Adapting the main theorem of Ohashi [Oha24], 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 [Cos18, Lemma 1]. For $1 - \lambda$, observe by [Gau07, §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 5, where $\mathtt{DeriveProfile}_{\mathtt{L}}$ determines $t_{\ker[2]}(P)$ using Equation (12) given the Legendre symbols of $x - \mu$ and $x - \nu$ for $x \in \mathtt{L}_{b_0,b_1}$.

Algorithm 5 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 $P = (X_1 : X_2 : X_3 : X_4)$

5 $(2,2)$-isogenies on Kummer surfaces

This work aims to showcase the practicality of Kummer surfaces in 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 canonical and 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 $\mathcal{K}^{\mathrm{Sqr}}$ defined over $\mathbb{F}_p$.

In Section 5.1, we first describe the general connection between subgroups $G \subset \mathcal{J}$ and kernels of isogenies $\mathcal{J} \to \mathcal{J}/G$, and more specifically $(2,2)$-isogenies between Kummer surfaces and subgroups $G = \langle L_{i,j}, L_{k,\ell} \rangle$ of the 2-torsion. We then go over the known theory for such isogenies between canonical Kummer surfaces in Section 5.2, which shows that we can construct such $(2,2)$-isogenies as a sequence of easy-to-describe linear maps. This description allows us to adapt such formulae also to $(2,2)$-isogenies between squared Kummer surfaces in Section 5.3. For the construction of efficient cryptographic protocols, we require the isogenies to be defined over the base field $\mathbb{F}_p$, however, in general, such isogenies are only defined over $\overline{\mathbb{F}}_p$. We take special care to study when these isogenies are defined over $\mathbb{F}_p$, which depends on the subgroup $G = \langle L_{i,j}, L_{k,\ell} \rangle$.

Comparison with other works. Computing $(2,2)$-isogenies between Kummer surfaces has been studied by various other works. Cassels and Flynn [CF96, Ch. 9] study $(2,2)$-isogenies between general Kummer surfaces, whilst Dartois, Maino, Pope, and Robert [DMPR24] use the theta model to give efficient formulæ for computing $(2,2)$-isogenies between canonical Kummer surfaces, later improved by [Rob24b, §8]. In the latter work, the authors also provide a constant-time implementation of their algorithm in Rust. In

contrast, in Section 5.3 we focus instead on deriving isogenies between squared Kummer surfaces following the method for canonical surfaces in [CCS25].

5.1 From subgroups to isogenies

We first define what we mean by a $(2,2)$-isogeny between Kummer surfaces (in any model). Consider an (N,N)-subgroup $G \subseteq \mathcal{J}_1[N]$, namely, 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 $e_N(D_R, D_S) = 1$, where e_N is the N-Weil pairing. An (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 $\Phi(0_1) = 0_2$, where 0_1 and 0_2 are the identity points on $\mathcal{J}_1$ and $\mathcal{J}_2$, respectively. Any (N,N)-isogeny Φ descends to a morphism of the associated Kummer surfaces $\varphi\colon \mathcal{K}_1 \to \mathcal{K}_2$, such that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{J}_1 & \xrightarrow{\Phi} & \mathcal{J}_2 \\ \downarrow & & \downarrow \\ \mathcal{K}_1 & \xrightarrow{\varphi} & \mathcal{K}_2 \end{array}$$

By abuse of notation, we call φ an (N,N)-*isogeny of Kummer surfaces*, with kernel $\ker\varphi = \rho(G)$, where $G \subseteq \mathcal{J}_1[N]$ and $\rho : \mathcal{J}_1 \to \mathcal{K}_1$ the projection to the Kummer surface.

Kernels of $(2,2)$-isogenies. For the remainder of the work, we specialize to $N = 2$, i.e., $(2,2)$-isogenies whose kernels are derived from subgroups $G \subseteq \mathcal{J}[2]$. If we denote $G = \langle D_{i,j}, D_{k,l} \rangle$, then G is a $(2,2)$-subgroup when $\{i,j\} \cap \{k,\ell\} = \emptyset$. Hence, on the Kummer surface, the associed 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.[13] 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 $\varphi_{ijk\ell} : \mathcal{K}_1 \to \mathcal{K}_2 := \mathcal{K}_1/\langle L_{i,j}, L_{k,\ell} \rangle$.

5.2 Isogenies between canonical Kummer surfaces

We briefly recall previous work on computing $(2,2)$-isogenies between canonical Kummer surfaces, as it will help inform us on how to adapt the theory to squared Kummer surfaces. Let $\mathcal{K}_1^{\text{can}}$ be a canonical Kummer surface described by $\mathbf{o} = (a : b : c : d)$ the identity point. To construct a $(2,2)$-isogeny between canonical Kummer surfaces, recall that we can describe the doubling map $[2] : \mathcal{K}_1^{\text{can}} \to \mathcal{K}_1^{\text{can}}$ as the sequence

$$[2] = \mathtt{C}_{\bar{T}} \circ \mathtt{H} \circ \mathtt{S} \circ \mathtt{C}_T \circ \mathtt{H} \circ \mathtt{S}$$

where $T = \mathtt{Inv}(\sqrt{A^2} : \sqrt{B^2} : \sqrt{C^2} : \sqrt{D^2})$ and $\bar{T} = \mathtt{Inv}(a : b : c : d)$.[14] This sequence factors through[15] the $(2,2)$-isogeny

$$\mathtt{C}_T \circ \mathtt{H} \circ \mathtt{S} : \mathcal{K}_1^{\text{can}} \to \mathcal{K}_2^{\text{can}}, \tag{13}$$

[13] This is equivalent to a quadratic splitting [Smi05] of the curve equation as used to compute Richelot isogenies.

[14] We recall that, a priori, only the constants A^2, B^2, C^2, D^2 are available to us (as they can be efficiently computed from the fundamental theta constants), and we must take square roots to compute A, B, C, D.

[15] See also the visualisation in in [RS17, Figure 1].

The kernel of this $(2,2)$-isogeny is the subgroup

$$\{(a:b:c:d),(a:b:-c:-d),(a:-b:c:-d),(a:-b:-c:d)\}, \tag{14}$$

which we call the *distinguished kernel*. This corresponds to $\langle L_{1,2}, L_{3,4}\rangle$ on the squared Kummer surface. A natural question arises: how do we describe the 14 other isogenies?

This question is answered in [CCS25]: Given a kernel $G \subseteq \mathcal{K}_1^{\text{can}}[2]$, we first apply an automorphism $\alpha_G^{\text{can}} : \mathcal{K}_1^{\text{can}} \to \mathcal{K}_1^{\text{can}}$ depending on the kernel G, such that $\alpha_G^{\text{can}}(G) \subseteq \mathcal{K}_1^{\text{can}}[2]$ is the distinguished kernel. Then, we apply the isogeny formula given in (13). This results in the following $(2,2)$-isogeny formula for the kernel G:

$$\mathtt{C}_T \circ \mathtt{H} \circ \mathtt{S} \circ \alpha_G^{\text{can}} : \mathcal{K}_1^{\text{can}} \to \mathcal{K}_1^{\text{can}}/G, \tag{15}$$

where now $\mathtt{Inv}(T) = \mathtt{S}^{-1} \circ \mathtt{H} \circ \mathtt{S} \circ \alpha_G^{\text{can}}(\mathbf{o})$. The morphism α_G is induced by a linear map on $\mathbb{A}^4$ defined by a matrix $\mathbf{A}_G$ whose entries lie in $\{0, \pm 1, \pm\beta\}$, where β is the root of $x^2+1 \in \mathbb{F}_p[x]$. The matrix $\mathbf{A}_G$ for each $(2,2)$-subgroup G is specified in [CCS25, Appendix A].

Remark 5. Using the theory of theta functions, [DMPR24] obtains the following formula for $(2,2)$-isogenies between canonical Kummer surfaces:

$$\mathtt{H} \circ \mathtt{C}_T \circ \mathtt{H} \circ \mathtt{S} : \mathcal{K}_1^{\text{can}} \to \mathcal{K}_2^{\text{can}},$$

where $T = \mathtt{Inv}(\sqrt{A^2} : \sqrt{B^2} : \sqrt{C^2} : \sqrt{D^2})$. At first glance, this may seem peculiar as there is no dependence on the kernel of the isogeny; there are fifteen possible kernels. Indeed, rather than considering isogenies corresponding to the different kernels, [DMPR24] explains how to set-up the underlying theta structure of the domain surface so that the kernel is always of a fixed form. By applying an automorphism to the theta structure, we can access other kernels [Rob24b, §8]. This is equivalent to the map α_G^{can} described above to move the required kernel to the distinguished kernel, as given in (15).

5.3 Isogenies between squared Kummer surfaces

From [RS17, Figure 1], we are similarly able to derive isogenies between squared Kummer surfaces, using the usual squaring map $\mathtt{S} : \mathcal{K}^{\text{can}} \to \mathcal{K}^{\text{Sqr}}$. Namely, we shift the sequence of operators and deduce that the $(2,2)$-isogeny $\varphi_{ijk\ell} : \mathcal{K}_1^{\text{Sqr}} \to \mathcal{K}_1^{\text{Sqr}}/\langle L_{i,j}, L_{k,\ell}\rangle$ with kernel $\langle L_{i,j}, L_{k,\ell}\rangle$ is given by

$$\varphi_{ijk\ell} := \mathtt{S} \circ \alpha_{ijk\ell}^{\text{Sqr}} \circ \mathtt{C}_T \circ \mathtt{H}, \tag{16}$$

where $T = \mathtt{Inv}(\sqrt{A^2} : \sqrt{B^2} : \sqrt{C^2} : \sqrt{D^2})$ with

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

for $\mathbf{o}$ the identity on the domain surface. We slightly need to adjust $\alpha_{ijk\ell}^{\text{Sqr}}$ in comparison to Section 5.2: let $G = \mathtt{C}_T \circ \mathtt{H}(\langle L_{i,j}, L_{k,\ell}\rangle)$, then $\alpha_{ijk\ell}^{\text{Sqr}} := \alpha_G^{\text{can}}$.

Computation of the scaling map. To compute the scaling map $\mathtt{C}_T$, we require the values $T = \mathtt{Inv}(\sqrt{A^2} : \sqrt{B^2} : \sqrt{C^2} : \sqrt{D^2})$, which we can derive from the identity $\mathbf{o} = (\mu_1 : \mu_2 : \mu_3 : \mu_4)$ of the domain surface. Working projectively, we can compute T with at most three square roots and a handful of additions in $\overline{\mathbb{F}}_p$. In some cases, however, T can be computed more efficiently, as we now detail. From this point onwards, we will assume that the squared Kummer surface $\mathcal{K}^{\text{Sqr}}$ has been constructed from a Jacobian via the maps detailed in Section 2.4.

For the kernel $\langle L_{1,2}, L_{3,4}\rangle$, the map α_{1234} is the identity, and so the corresponding $(2,2)$-isogeny is given by

$$\varphi_{1234} = \mathtt{C}_{\mathtt{Inv}(A^2:B^2:C^2:D^2)} \circ \mathtt{S} \circ \mathtt{H} : \mathcal{K}_1^{\mathrm{Sqr}} \to \mathcal{K}_2^{\mathrm{Sqr}},$$

where we have swapped $\mathtt{S}$ and $\mathtt{C}$ in comparison to (16) to avoid taking square roots. In this case, the isogeny is defined over the field of definition of $\mathcal{K}_1^{\mathrm{Sqr}}$.

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 $(1\colon \star\colon 0\colon 0)$ 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

$$\frac{A}{B} = c_1 \frac{h_{R,1}}{h_{R,2}}, \quad \frac{A}{C} = c_2 \frac{h_{S,1}}{h_{S,3}}, \quad \text{and} \quad \frac{C}{D} = c_3 \frac{h_{R,3}}{h_{R,4}}, \tag{17}$$

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

$$\left(\begin{array}{ccc} h_{R,2}h_{S,3}h_{S,3}h_{R,4} & : & c_1 \cdot h_{R,1}h_{S,3}h_{S,3}h_{R,4} \\ c_2 \cdot h_{R,2}h_{S,1}h_{S,3}h_{R,4} & : & c_2 \cdot c_3 \cdot h_{R,2}h_{S,3}h_{S,1}h_{R,3} \end{array} : \right).$$

In this case, the isogeny is defined over the field of definition of $\mathcal{K}_1^{\mathrm{Sqr}}$, its 4-torsion points and the constants c_i, which are given in Appendix D.

Remark 6. One can view this method of computing the scaling map $\mathtt{C}_T$ as making a choice of square root dictated by the torsion points lying above the kernel. Furthermore, it is, in some sense, the 'correct' choice. Indeed, suppose on the domain surface we have an isotropic subgroup $\langle R, S\rangle \in \mathcal{K}_1^{\mathrm{Sqr}}[4]$, such that $\langle [2]R, [2]S\rangle$ is the kernel of the $(2,2)$-isogeny we want to compute. If we compute the scaling map $\mathtt{C}_T$ with these torsion points, the corresponding choice of square root will be compatible with the underlying theta structure, in the sense that the kernel of the next $(2,2)$-isogeny step will be of a fixed form, that is, the automorphism α will be $\mathtt{H}$, which corresponds to $\langle L_{2,3}, L_{4,5}\rangle$. This phenomenon is similarly described in [KMM+25, §3.5], where the choice of these square roots is equivalent to the choice of the kernel for the next isogeny in the chain.

Isogenies defined over $\mathbb{F}_p$. Looking forward to our application in Section 7, we are interested in squared Kummer surfaces defined over $\mathbb{F}_p$. Furthermore, for efficiency, we want our isogenies to also be defined over $\mathbb{F}_p$. Assuming we have $\mathbb{F}_p$-rational 4-torsion points lying above the kernel generators $L_{i,j}$ and $L_{k,\ell}$, the corresponding isogeny $\varphi_{ijk\ell}$ is defined over $\mathbb{F}_p$ when

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

These isogenies coincide precisely with those derived by Costello [Cos18], which we use in Section 7 to perform 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.

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 [Cos18]. 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 [DMPR24]. 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.

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 collapse to the isogenies given by Costello [Cos18] in this special setting when $\nu = \lambda\mu$. We observe that the isogenies are fully described by a point P, rather than two points as in the previous sections.

Lemma 9 ([Cos18]). *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 Case 2, we have $\alpha = \mathtt{H}$ and $S = \mathtt{Inv}(A : B : C : D)$, where S is as in the statement of

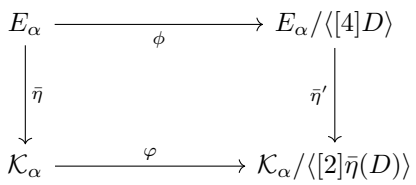


Figure 3: The elliptic $(2,2)$-isogeny φ induced by a 2-isogeny ϕ.

the lemma. Indeed, observing that for ellpitic Kummer surfaces we have $C = D = 1$, we have that $\mathtt{H}(P) = (A\colon B\colon B\colon A)$ and so $h_1/h_2 = A/B$.

When $P = L_{2,3} = (1\colon 0\colon \star\colon 0)$, using Equation (17) we have that $h_1'/h_3' = A$. Therefore $S = (1\colon h_1/h_2 : h_1'/h_3' : h_1'/h_3')$. In the other case, $P = (1\colon 0\colon 0\colon \star)$ and $h_1'/h_4' = A$. Therefore $S = (1\colon h_1/h_2 : h_1'/h_4' : h_1'/h_4')$.

For the isogeny $\varphi_{1/\alpha}$ from Case 3, we find that $\alpha(X_1\colon X_2\colon X_3\colon X_4) = \mathtt{H}(-X_1\colon X_2\colon X_3\colon X_4)$. We can verify that $S = \mathtt{Inv}(-A\colon B\colon C\colon D)$ in a similar method to above, and so $\alpha \circ \mathtt{C}_{\mathtt{Inv}(A\colon B\colon C\colon D)} = \mathtt{H} \circ \mathtt{C}_S$, as required. We note that the fact that α can be set to be $\mathtt{H}$ in both Case 2 and 3 is a consequence of Remark 6.

6.2 Images of $(2,2)$-isogenies

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 [Gau07, §4.2] shows that for elliptic Kummer surfaces, we have

$$\lambda = \frac{\mu_1}{\mu_2}, \quad \mu = \frac{\tau\mu_2 - 1}{\tau\lambda\mu_2 - 1}, \quad \nu = \lambda\mu. \tag{18}$$

with τ as defined in Section 2.5. 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 (18).

Evaluation points under elliptic isogenies. This amounts to evaluating the maps from Lemma 9 at a point $(X_1\colon X_2\colon X_3\colon X_4)$. 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 $16\mathbf{a}$.

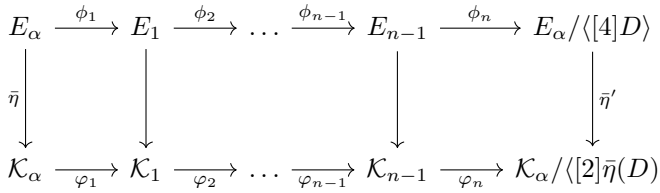


Figure 4: The elliptic $(2^n, 2^n)$-isogeny φ induced by the 2^n-isogeny ϕ .

6.3 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.

A better way to compute chains of isogenies is to use *optimal strategies*. This was first introduced in the context of SIDH/SIKE [DJP14], and has shown to be adaptable to the Kummer surface setting [DMPR24; CCS25; CPR23]. 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 1.8x 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 example application: performing SQIsign on Kummer surfaces. We refer to [DKL+20; AAA+25] 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 $e \approx 1000$, split up 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 $K_i \in E_i[2^f]$. 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 [CEMR24].

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 7. 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 $s \in \mathbb{Z}/2^f\mathbb{Z}$.

In Section 4, we gave a detailed analysis of the cost of both steps in the higher-dimensional case. Using the sampling method from Section 4.5, 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 3 (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 $D_K \in \text{Im}(\eta) \subset \mathcal{J}_\mathcal{C}(\mathbb{F}_p)$, 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 $\mathcal{K}_{\lambda,\mu,\nu}^{\text{Sqr}}$ as K, P and Q, and derives

$D, S \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$ using PointDifference. Finally, the signer recomputes both

$$K_D \leftarrow \texttt{xMUL}(\texttt{3ptLadder}(P, Q, D, s), 2^f) \text{ and } K_S \leftarrow \texttt{xMUL}(\texttt{3ptLadder}(P, Q, S, s), 2^f),$$

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 $P, Q \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$, and use efficient Kummer arithmetic, instead of expensive Jacobian arithmetic, to derive $K \in \mathcal{K}^{\text{Sqr}}_{\lambda,\mu,\nu}$.

7.5 Performance of SQIsign on Kummer Surfaces

Our benchmarks, using the same cost model as AprèsSQI, show that SQIsign verification on Kummer surfaces, in comparison to elliptic curves, takes less than $1.5\times$ the number of $\mathbb{F}_p$-operations for both the uncompressed variant and the compressed variant. As shown in [BCLS14], 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 [BCLS14], this can potentially outperform SQIsign verification between elliptic curves, and compete with the two-dimensional approaches over $\mathbb{F}_{p^2}$ [BDF+24; NOC+24; DF24] 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.

Lastly, we expect to see highly-specialised and optimised implementations of two-dimensional isogenies in specific software or hardware in the (near) future. In such situations, whenever a protocol requires mostly two-dimensional isogenies and only the occasional one-dimensional isogeny, such as verification in some isogeny-based signature schemes, the tools and ideas in this work allow us to compute this one-dimensional isogeny using the specialised two-dimensional implementation. This decreases the code complexity, and may save space in a hardware implementation.

References

[AAA+25] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2025. URL: https://sqisign.org.

[BCHL16] Joppe W Bos, Craig Costello, Huseyin Hisil, and Kristin Lauter. Fast cryptography in genus 2. *Journal of Cryptology*, 29:28–60, 2016. DOI: 10.1007/S00145-014-9188-7.

[BCLS14] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Peter Schwabe. Kummer strikes back: new DH speed records. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 317–337. Springer, 2014. DOI: 10.1007/978-3-662-45611-8_17.

[BCP97] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. DOI: 10.1006/jsco.1996.0125. Computational algebra and number theory.

[BDF+24] Andrea Basso, Pierrick Dartois, Luca De Feo, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D-West - the fast, the small, and the safer. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part III*, volume 15486 of *Lecture Notes in Computer Science*, pages 339–370. Springer, 2024. DOI: 10.1007/978-981-96-0891-1_11.

[BDK+18] 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)*, pages 353–367. IEEE, 2018. DOI: 10.1109/EUROSP.2018.00032.

[Ber06] Daniel J. Bernstein. Elliptic vs. hyperelliptic, part 1, 2006. URL: http://cr.yp.to/talks.html#2006.09.20.

[BHK+19] 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, pages 2129–2146, New York, NY, USA. Association for Computing Machinery, 2019. ISBN: 9781450367479. DOI: 10.1145/3319535.3363229.

[BKLS02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002. DOI: 10.1007/3-540-45708-9_23.

[BL14] Daniel J. Bernstein and Tanja Lange. Hyper-and-elliptic-curve cryptography. *LMS J. Comput. Math.*, 17(Theory):181–202, 2014. DOI: 10.1112/S1461157014000394. URL: https://doi.org/10.1112/s1461157014000394.

[Bro93] Bradley Wayne Brock. *Superspecial curves of genera two and three*. Princeton University, 1993.

[Bru11] Peter Bruin. The Tate pairing for abelian varieties over finite fields. *Journal de theorie des nombres de Bordeaux*, 23(2):323–328, 2011. DOI: 10.5802/jtnb.764.

[CC86] David V Chudnovsky and Gregory V Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *Advances in Applied Mathematics*, 7(4):385–434, 1986. DOI: 10.1016/0196-8858(86)90023-0.

[CCS17] 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*, pages 465–481. Springer, 2017. DOI: 10.1007/978-3-319-69453-5_25.

[CCS22] Maria Corte-Real Santos, Craig Costello, and Jia Shi. Accelerating the Delfs–Galbraith algorithm with fast subfield root detection. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 285–314. Springer, 2022. DOI: 10.1007/978-3-031-15982-4_10.

[CCS25] Maria Corte-Real Santos, Craig Costello, and Benjamin Smith. Efficient $(3, 3)$-isogenies on fast Kummer surfaces. *Research in Number Theory*, 11(1):1–27, 2025. DOI: 10.1007/s40993-024-00600-y.

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *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*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447. Springer, 2023. DOI: 10.1007/978-3-031-30589-4_15.

[CDS20] W. Castryck, T. Decru, and B. Smith. Hash functions from superspecial genus-2 curves using Richelot isogenies. *Journal of Mathematical Cryptology*, 14(1):268–292, 2020. DOI: 10.1515/JMC-2019-0021.

[CEMR24] 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. DOI: 10.1007/978-3-031-58716-0_3.

[CF96] J. W. S. Cassels and E. Victor Flynn. *Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1996.

[CHM+23] 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 Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 762–792. Springer, 2023. DOI: 10.1007/978-3-031-38548-3_25.

[CJL+17] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of SIDH public keys. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *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*, volume 10210 of *Lecture Notes in Computer Science*, pages 679–706, 2017. DOI: 10.1007/978-3-319-56620-7_24.

[Cos11] Romain Cosset. *Applications of theta functions for hyperelliptic curve cryptography*. PhD thesis, Université Henri Poincaré-Nancy I, 2011.

[Cos18] Craig Costello. Computing supersingular isogenies on Kummer surfaces. In Thomas Peyrin and Steven D. Galbraith, editors, *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*, volume 11274 of *Lecture Notes in Computer Science*, pages 428–456. Springer, 2018. DOI: 10.1007/978-3-030-03332-3_16.

[CPR23] 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.

[DEF+25] Pierrick Dartois, Jonathan Komada Eriksen, Tako Boris Fouotsa, Arthur Herlédan Le Merdy, Riccardo Invernizzi, Damien Robert, Ryan Rueger, Frederik Vercauteren, and Benjamin Wesolowski. PEGASIS: practical effective class group action using 4-dimensional isogenies. In *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part I*, pages 67–99, 2025. DOI: 10.1007/978-3-032-01855-7_3.

[DF24] Max Duparc and Tako Boris Fouotsa. SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part III*, volume 15486 of *Lecture Notes in Computer Science*, pages 396–429. Springer, 2024. DOI: 10.1007/978-981-96-0891-1_13.

[DJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014. DOI: 10.1515/jmc-2012-0015.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: a lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*:238–268, 2018. DOI: 10.13154/TCHES.V2018.I1.238-268.

[DKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *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*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020. DOI: 10.1007/978-3-030-64837-4_3.

[DLLW23] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the Deuring correspondence - towards practical and secure SQIsign signatures. In Carmit Hazay and Martijn Stam, editors, *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*, volume 14008 of *Lecture Notes in Computer Science*, pages 659–690. Springer, 2023. DOI: 10.1007/978-3-031-30589-4_23.

[DLRW24] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQIsignHD: new dimensions in cryptography. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2024. DOI: 10.1007/978-3-031-58716-0_1.

[DMPR24] 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. Kai-Min Chung and Yu Sasaki, editors, 2024. DOI: 10.1007/978-981-96-0891-1_10.

[FHK+18] 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. *Submission to the NIST's post-quantum cryptography standardization process*, 36(5):1–75, 2018.

[Gau07] Pierrick Gaudry. Fast genus 2 arithmetic based on theta functions. *Journal of Mathematical Cryptology*, 1(3):243–265, 2007. DOI: 10.1515/JMC.2007.012.

[GHV07] Steven D. Galbraith, Florian Hess, and Frederik Vercauteren. Hyperelliptic pairings. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Pairing-Based Cryptography - Pairing 2007, First International Conference, Tokyo, Japan, July 2-4, 2007, Proceedings*, volume 4575 of *Lecture Notes in Computer Science*, pages 108–131. Springer, 2007. ISBN: 978-3-540-73488-8. DOI: 10.1007/978-3-540-73489-5_7. URL: https://doi.org/10.1007/978-3-540-73489-5%5C_7.

[HSSI99] 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*, pages 190–205. Springer, 1999. DOI: 10.1007/3-540-48910-X_14.

[Hus04] Dale Husemöller. *Elliptic Curves, 2nd edition*. Springer, 2004.

[Igu60] Jun-ichi Igusa. Arithmetic variety of moduli for genus two. *Annals of Mathematics*, 72(3):612–649, 1960. DOI: 10.2307/1970233.

[Jou04] Antoine Joux. A one round protocol for tripartite Diffie–Hellman. *Journal of cryptology*, 17(4):263–276, 2004. DOI: 10.1007/S00145-004-0312-Y.

[KMM+25] Sabrina Kunzweiler, Luciano Maino, Tomoki Moriya, Christophe Petit, Giacomo Pope, Damien Robert, Miha Stopar, and Yan Bo Ti. Radical 2-isogenies and cryptographic hash functions in dimensions 1, 2 and 3. In *Public-key cryptography—PKC 2025. Part III*. Volume 15676, Lecture Notes in Computer Science, pages 265–299. Springer, Cham, 2025. ISBN: 978-3-031-91825-4; 978-3-031-91826-1. DOI: 10.1007/978-3-031-91826-1_9.

[Kos23] Dmitrii I. Koshelev. Subgroup membership testing on elliptic curves via the tate pairing. *J. Cryptogr. Eng.*, 13(1):125–128, 2023. DOI: 10.1007/S13389-022-00296-9. URL: https://doi.org/10.1007/s13389-022-00296-9.

[LO98] Ke-Zheng Li and Frans Oort. *Moduli of supersingular abelian varieties*, volume 1680. Springer Science & Business Media, 1998.

[LR10] David Lubicz and Damien Robert. Efficient pairing computation with theta functions. In Guillaume Hanrot, François Morain, and Emmanuel Thomé, editors, *Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings*, volume 6197 of *Lecture Notes in Computer Science*, pages 251–269. Springer, 2010. ISBN: 978-3-642-14517-9. DOI: 10.1007/978-3-642-14518-6_21. URL: https://doi.org/10.1007/978-3-642-14518-6%5C_21.

[LR16] David Lubicz and Damien Robert. Arithmetic on abelian and Kummer varieties. *Finite Fields Appl.*, 39:130–158, 2016. ISSN: 1071-5797,1090-2465. DOI: 10.1016/j.ffa.2016.01.009.

[LWXZ24] Kaizhan Lin, Weize Wang, Zheng Xu, and Chang-An Zhao. A faster software implementation of SQISign. *IEEE Transactions on Information Theory*, 2024. DOI: 10.1109/TIT.2024.3423675.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *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*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023. DOI: 10.1007/978-3-031-30589-4_16.

[NIS22] 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.

[NOC+24] Kohei Nakagawa, Hiroshi Onuki, Wouter Castryck, Mingjie Chen, Riccardo Invernizzi, Gioella Lorenzon, and Frederik Vercauteren. SQIsign2D-East: A new signature scheme using 2-dimensional isogenies. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part III*, volume 15486 of *Lecture Notes in Computer Science*, pages 272–303. Springer, 2024. DOI: 10.1007/978-981-96-0891-1_9.

[Oha24] Ryo Ohashi. On the rosenhain forms of superspecial curves of genus two. *Finite Fields Their Appl.*, 97:102445, 2024. DOI: 10.1016/J.FFA.2024.102445. URL: https://doi.org/10.1016/j.ffa.2024.102445.

[PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.) *IEEE Trans. Inf. Theory*, 24(1):106–110, 1978. DOI: 10.1109/TIT.1978.1055817. URL: https://doi.org/10.1109/TIT.1978.1055817.

[Rob23a] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *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*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023. DOI: 10.1007/978-3-031-30589-4_17.

[Rob23b] 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.

[Rob24a] Damien Robert. Fast pairings via biextensions and cubical arithmetic. Cryptology ePrint Archive, Paper 2024/517, 2024. URL: https://eprint.iacr.org/2024/517.

[Rob24b] Damien Robert. Some notes on algorithms for abelian varieties. Cryptology ePrint Archive, Paper 2024/406, 2024. URL: https://eprint.iacr.org/2024/406.

[Ros95] Georg Rosenhain. *Abhandlung über die Functionen zweier Variabler mit vier Perioden: welche die Inversen sind der ultra-elliptischen Integrale erster Klasse*, number 65. W. Engelmann, 1895.

[RS17] Joost Renes and Benjamin Smith. qDSA: small and secure digital signatures with curve-based Diffie-Hellman key pairs. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 273–302. Springer, 2017. DOI: 10.1007/978-3-319-70697-9_10.

[RSSB16] 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*, pages 301–320. Springer, 2016. DOI: 10.1007/978-3-662-53140-2_15.

[Sch03] Jasper Scholten. Weil restriction of an elliptic curve over a quadratic extension, 2003.

[Smi05] Benjamin Andrew Smith. *Explicit endomorphisms and correspondences*. PhD thesis, University of Sydney, 2005.

[The21] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.2)*. https://www.sagemath.org. 2021.

[Wei61] André Weil. Adeles and algebraic groups: lectures, 1961.

[ZJP+19] Gustavo H. M. Zanon, Marcos A. Simplício Jr., Geovandro C. C. F. Pereira, Javad Doliskani, and Paulo S. L. M. Barreto. Faster key compression for isogeny-based cryptosystems. *IEEE Trans. Computers*, 68(5):688–701, 2019. DOI: 10.1109/TC.2018.2878829.

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 [CF96] as well as for the squared Kummer surface, which is original work. We then specialse to elliptic Kummer surfaces.

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 following 4×3 matrix

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

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

$$\begin{aligned} w_{41} =& - g_1g_2^2h_0h_1 + g_1^2g_2h_0h_2 + g_0g_2^2h_1^2 - 4g_0g_2^2h_0h_2 \\ & - g_0g_1g_2h_1h_2 + g_0g_1g_2h_0h_3 - g_0^2g_2h_1h_3, \\ w_{42} =& g_1^2g_2h_0h_3 - g_1^3h_0h_4 - 2g_0g_2^2h_0h_3 - g_0g_1g_2h_1h_3 \\ & + 4g_0g_1g_2h_0h_4 + g_0g_1^2h_1h_4 - 2g_0^2g_2h_1h_4, \\ w_{43} =& - g_0g_2^2h_1h_3 - g_0g_1g_2h_2h_3 + g_0g_1g_2h_1h_4 + g_0g_1^2h_2h_4 \\ & + g_0^2g_2h_3^2 - 4g_0^2g_2h_2h_4 - g_0^2g_1h_3h_4, \\ w_{44} =& - g_2^2h_0 - g_0g_2h_2 - g_0^2h_4. \end{aligned}$$

The canonical Kummer surface. For the canonical Kummer surface, addition by a point of order 2 is very simple. As described by Gaudry [Gau07], 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

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

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

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

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

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

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

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

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 [Rob24a]. We assume we want to compute the Tate pairing $t_2(L_{i,j}, Q)$ with $L_{i,j} \in \mathcal{K}[2]$ and $Q \in \mathcal{K}$. By Appendix A, 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 $t_2(L_{i,j}, Q)$ 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.

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.

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 $t_2(L_{i,j}, Q)$.
1: $D \leftarrow W_{i,j} \cdot Q$
2: $\lambda_Q \leftarrow D_{n_{i,j}}/Q_{n_{i,j}}$
3: **return** `IsSquare`$(\frac{\lambda_Q}{\lambda_{i,j}})$

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

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 $1 \leq i < j \leq 4$ 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 $F(u_0^2, u_0u_1, u_0, \omega)^T = (f_{i-j})_{1\leq i<j\leq 4}$. 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 $\mathbb{F}_{p^2} = \mathbb{F}_p(\beta)$ where β 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 $c_i \in \{1, -\beta\}$. From this we can derive $\frac{A}{B}$, $\frac{A}{C}$, and $\frac{A}{D}$ to get

$$\left(\frac{1}{A} : \frac{1}{B} : \frac{1}{C} : \frac{1}{D}\right) = \Big(h(R)_2h(R)_4h(S)_3\colon c_1h(R)_1h(R)_4h(S)_3\colon \\ \colon c_2h(R)_2h(R)_4h(S)_1\colon c_2c_3h(R)_2h(R)_3h(S)_1\Big).$$

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