Richelot Isogenies, Pairings on Squared Kummer Surfaces and Applications *

Chao Chen[1,2] and Fangguo Zhang[1,2]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
[2] Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China
isszhfg@mail.sysu.edu.cn

Abstract. Isogeny-based cryptosystem from elliptic curves has been well studied for several years, but there are fewer works about isogenies on hyperelliptic curves to this date. In this work, we make the first step to explore isogenies and pairings on generic squared Kummer surfaces, which is believed to be a better type of Kummer surfaces. The core of our work is the Richelot isogeny having two kernels together with each dual onto the squared Kummer surfaces, then a chain of Richelot isogenies is constructed simply. Besides, with the coordinate system on the Kummer surface, we modify the squared pairings, so as to propose a self-contained pairing named squared symmetric pairing, which can be evaluated with arithmetic on the same squared Kummer surface. In the end, as applications, we present a Verifiable Delay Function and a Delay Encryption on squared Kummer surfaces.

Keywords: Hyperelliptic Curves · Squared Kummer Surfaces · Richelot Isogenies · Squared Pairings · Verifiable Delay Function · Delay Encryption.

1 Introduction

In the last ten years, isogeny-based cryptography has attracted much attention as it is believed resistant to quantum attacks, yielding several basic schemes such as hash functions [8], key exchange protocols [7,14], and SIKE [27] is one of the alternate candidates in NIST project for the post-quantum cryptosystems. In these schemes, isogenies between elliptic curves can be implemented efficiently via the Vélu formula [39].

In general, elliptic curves can be seen as abelian varieties of dimension one, which makes it natural to consider the curves of high genus. At the same security level, the parameter size of classical Discrete Logarithm Problem (DLP) can be greatly reduced using hyperelliptic curves. Whereas, isogenies between

* This work is supported by Guangdong Major Project of Basic and Applied Basic Research (2019B030302008) and the National Natural Science Foundation of China (No. 61972429).

Jacobians of hyperelliptic curves are much harder than those of elliptic curves. For Richelot isogenies (i.e., $(2,2)$-isogenies), Smith [38, Chapter 8] introduced a formula of genus two, and more discussions can be found in [6, 13, 19, 20, 22]. Nevertheless, the image of points under the above maps is less known. In return, for (ℓ,ℓ)-isogenies where ℓ is an odd integer (in particular, $\ell = 3, 5$), similar formulae were created in [4, 21]. The first practical method to solve this problem was raised by Lubicz and Robert [30], where they employed theta functions to propose an algorithm for computing isogenies between abelian varieties. After that, an explicit (ℓ,ℓ)-isogeny algorithm, together with one algorithm for computing the image of a point under isogenies, was presented [11]. Unfortunately, these algorithms were inefficient for implementation.

From the point of computation efficiency, the Kummer surfaces, proposed by Chudnovskys [9] and Gandry [25], are a better choice. As the quotient of genus-2 Jacobians by ± 1, there are formulae for fast scalar multiplication. Moreover, the arithmetic on Kummer surfaces can be performed via theta functions [31]. There is much progress in this realm during recent years. One is the study of squared Kummer surfaces $\mathcal{K}^{Sqr}$ (which have better connections with Jacobians), and the fast arithmetic was summarized in [3]. The signature scheme qDSA [37] was implemented on squared Kummer surfaces, and Cosset [10] used these surfaces to realize fast factorization. Besides, for isogenies between squared Kummer surfaces, one important discovery was the supersingular isogenies [12]. This work inspire us to study the Richelot isogenies between squared Kummer surfaces and their appealing properties.

As another crucial concept, pairings are powerful tools in cryptography, which have been investigated for several years in hyperelliptic curves, and more information about hyperelliptic pairings is referred to [23]. The computation of pairings is a critical problem, and most approaches take of Miller's algorithm [17]. To change this situation, Lubicz and Robert [29] presented a new approach with fundamental theta functions. In their algorithm, computing pairings only relies on several values of theta functions, corresponding to the coordinates on Kummer surfaces. However, the pairings on squared Kummer surfaces are hard to define. The first attempt was done by [29] which defines the symmetric pairings on Kummer varieties, improved from the work of computing pairings only with x-coordinates in [24]. We proceed with these attempts and propose more efficient pairings on squared Kummer surfaces in this work.

Nowadays, some useful protocols are presented with isogenies, i.e., the Verifiable Delay Function (VDF) and Delay Encryption were established with supersingular isogenies. The VDF [1] for slow computation and fast verification is presented by Dan Boneh et al. in 2018, and there are numerous schemes [16,18,36,41] after its proposal. The first VDF based on isogenies and pairings was raised in [15]. In their scheme, algorithm Eval computes a cyclic isogeny $\phi : E_1 \to E_2$ and its dual isogeny $\hat{\phi}$ of degree ℓ^T in ℓ-isogeny graph. Currently, Burdges and De Feo introduced a new cryptographic primitive named Delay Encryption [5], which can be seen as a hybrid of Identity-Based Encryption (IBE) [2] and key encapsulation. In addition, they implemented it with isogenies and pairings first.

We propose a new VDF and Delay Encryption via Richelot isogenies and squared pairing on Kummer surfaces.

Our Contributions. In this work, we adopt squared Kummer surfaces from the Jacobians of genus two hyperelliptic curves, on which Richelot isogenies and pairings are defined and an associated VDF scheme plus a Delay Encryption scheme is proposed. Our contributions are summarized as follows:

- First of all, we define isogenies between squared Kummer surfaces via the isogenous map between their corresponding Jacobian. With four fundamental theta functions and several additional operations, we construct the Richelot isogenies and their dual isogenies between generic squared Kummer surfaces, and further establish the chains of these isogenies.
- We propose a squared pairing on Jacobians, which eases the computation using the coordinates of points on squared Kummer surfaces only. Besides, we also introduce the self-contained squared symmetric pairings on these surfaces, implying that these pairings can be evaluated with arithmetic on the same surfaces.
- Based on the above designs, we present a new VDF and Delay Encryption, each of which is proved secure against known attacks on the isogeny shortcut problems and is executed for the concrete implementations.

Organization. Section 2 provides some necessary backgrounds. The Richelot isogenies and its dual isogeny between squared Kummer surfaces are studied in Section 3. Section 4 introduces squared symmetric pairings as self-contained pairings. Two applications (VDF and Delay Encryption) along with analysis will be proposed in Section 5. Finally, Section 6 concludes our work.

2 Preliminaries

In this section, we recall the necessary mathematical backgrounds for this paper.

2.1 Hyperelliptic Curves

Let $\overline{\mathbb{F}}_q$ be the algebraic closure of the field $\mathbb{F}_q$. A hyperelliptic curve C of genus g over $\mathbb{F}_q$ with $g \geq 1$ is given by the following equation:

$$C : y^2 + h(x)y = f(x),$$

where $f(x)$ is a monic polynomial of degree $2g+1$, $h(x)$ is a polynomial of degree at most g, and there are no solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ simultaneously satisfying the equation $y^2+h(x)y = f(x)$ and the partial derivative equations $2y+h(x) = 0$ and $h'(x)y - f'(x) = 0$. Hence, C is a nonsingular hyperelliptic curve and has only one point P_∞ at infinity. For any algebraic extension $\mathbb{F}_{q^k}$ of $\mathbb{F}_q$, we consider the set $C(\mathbb{F}_{q^k}) := \{(x, y) \in \mathbb{F}_{q^k} \times \mathbb{F}_{q^k} \mid y^2 + h(x)y = f(x)\} \cup \{P_\infty\}$, called the set of $\mathbb{F}_{q^k}$-rational points on C.

When $g \geq 2$, the set $C(\mathbb{F}_{q^k})$ does not form a group, but we can embed C into an abelian variety of dimension g, which is called the Jacobian of C and denoted by J_C. The Jacobian J_C is isomorphic to the divisor class group of degree zero Pic_C^0. Let $\mathcal{O}$ be the identity of J_C.

Every divisor in Jacobian J_C over field K can be expressed in Mumford representation as a pair $(u(x), v(x))$ of polynomials in $K[x]$, such that $u(x)$ is monic, and $u(x)$ divides $f(x)-h(x)v(x)-v(x)^2$, with $\deg(v(x)) < \deg(u(x)) \leq g$. Let $P_1 = (x_1, y_1), \cdots, P_g = (x_g, y_g)$ be g points on the hyperelliptic curve C, then the Mumford representation $(u(x), v(x))$ associated with g points satisfies $u(x_i) = 0$ and $v(x_i) = y_i$, for all $i = 1, \cdots, g$. For $r \in \mathbb{N}$, we define

$$J_C[r] := \{D \in J_C \mid rD = \mathcal{O}\}$$

as the r-torsion subgroup of J_C.

2.2 Hyperelliptic Pairings

Pairings are useful tools in hyperelliptic curves. The definitions of two familiar pairings on hyperelliptic curves are summarized as follows.

Let C be a hyperelliptic curve of genus g over $\mathbb{F}_q$. Let r be a divisor of $\#J_C$, and coprime to q. The embedding degree is defined to be the smallest positive integer k such that $r \mid (q^k - 1)$. The group of r-th roots of unity in $\mathbb{F}_{q^k}^*$ is denoted by $\mu_r = \{z \in \mathbb{F}_{q^k}^* \mid z^r = 1\}$.

The Weil pairing is a non-degenerate bilinear map

$$J_C(\mathbb{F}_{q^k})[r] \times J_C(\mathbb{F}_{q^k})[r] \longrightarrow \mu_r,$$

which is denoted as $e_r(D_1, D_2)$.

The Tate-Lichtenbaum pairing is a non-degenerate bilinear map

$$J_C(\mathbb{F}_{q^k})[r] \times J_C(\mathbb{F}_{q^k})/rJ_C(\mathbb{F}_{q^k}) \longrightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r,$$

which is denoted as $\langle D_1, D_2 \rangle_r$. To achieve cryptographic applications, we consider the reduced (or modified) pairing

$$t_r(D_1, D_2) = \langle D_1, D_2 \rangle_r^{\frac{q^k-1}{r}}.$$

Similar to pairings of elliptic curves, Miller's algorithm [17, 32] is used to compute hyperelliptic pairings. For more detailed discussions, it refers to [23].

2.3 Richelot Isogenies

It is well-known that we can compute the isogenous elliptic curve through Vélu formula [39], which is the foundation of isogeny-based cryptography. Nevertheless, with the growth of genus, there is no efficient algorithm to compute isogenies between Jacobians.

Since the Jacobian J_C of a curve C is a principally polarized abelian variety, we could consider *isogeny* of princupally polarized abelian varieties, which is a finite dominant homomorphism of abelian varieties, and the kernel of isogeny is a finite isotropic group. The Richelot isogeny is the isogeny whose kernel is contained in the two-torsion subgroup $J_C[2]$. Smith [38] summarized the Richelot isogenies on Jacobians of genus two, whose kernel is a 2-torsion isotropic subgroup of $J_C[2]$.

Proposition 1 ([38]). *Let R be a proper, nontrivial subgroup of $J_C[2]$. If R is the kernel of an isogeny between principally polarised abelian surfaces, then R is a maximal 2-Weil isotropic subgroup of $J_C[2]$ (that is, the 2-Weil pairing restricts trivially to R, and R is not properly contained in any other such subgroup).*

The formula of Richelot isogenies was established in [38, Chapter 8]. In addition, Flynn and Ti [22] analyzed the structure of maximal ℓ-isotropic subgroups of $A[\ell^n]$ and obtained the number of ℓ^n-isogenous abelian varieties, then they also presented a key exchange protocol named the genus two SIDH. Unfortunately, it is difficult to determine the image of a point under the isogenous map.

When we generalize the Jacobian J_C of hyperelliptic curve C to principally polarised abelian varieties A of higher dimension, there is more progress. For instance, for an abelian variety of dimension two, there are fifteen $(2,2)$-isogenous abelian varieties. Florit and Smith [19] illustrated the local neighborhoods of vertices and edges in the $(2,2)$-isogeny graph, but some vertices fail to be the Jacobian of some hyperelliptic curves.

2.4 Theta Functions

We briefly introduce the theta functions in the following. An abelian variety A of dimension g over $\mathbb{C}$ is analytically isomorphic to a torus $\mathbb{C}^g/(\Omega\mathbb{Z}^g + \mathbb{Z}^g)$ with Ω in the Siegel half-space $\mathcal{H}_g$.

The Riemann theta function [34,35] associated with Ω is an analytic function defined as

$$\theta(z,\Omega) = \sum_{m\in\mathbb{Z}^g} \exp(\pi i^t m\Omega m + 2\pi i^t mz),$$

where ${}^t m$ represents the transpose of column vector m. For $a, b \in \mathbb{Q}^g$, the theta function of characteristics a, b is

$$\theta\begin{bmatrix} a \\ b \end{bmatrix}(z,\Omega) = \theta(z+\Omega a + b, \Omega)\exp(\pi i^t a\Omega a + 2\pi i^t a(b+z)).$$

The basis of theta functions of level n given by [35] is

$$\left(\theta\begin{bmatrix} 0 \\ b \end{bmatrix}(z, \frac{\Omega}{n})\right)_{b\in\frac{1}{n}\mathbb{Z}^g/\mathbb{Z}^g}.$$

If $n = k^2$, we have another basis of level n theta functions

$$\left(\theta\begin{bmatrix} a \\ b \end{bmatrix}(kz,\Omega)\right)_{a,b\in\frac{1}{k}\mathbb{Z}^g/\mathbb{Z}^g}.$$

The values of the theta functions with characteristics evaluated at $z = (0, \cdots, 0)$ are called the theta constants, which are the coordinates of theta null point.

When $g = 2$, the notations of four fundamental theta functions are fixed:

$$\begin{aligned} \theta_1(z) &= \theta\left[\begin{smallmatrix}(0,0)\\(0,0)\end{smallmatrix}\right](z,\Omega); \qquad \theta_2(z) = \theta\left[\begin{smallmatrix}(0,0)\\(\frac{1}{2},\frac{1}{2})\end{smallmatrix}\right](z,\Omega); \\ \theta_3(z) &= \theta\left[\begin{smallmatrix}(0,0)\\(\frac{1}{2},0)\end{smallmatrix}\right](z,\Omega); \qquad \theta_4(z) = \theta\left[\begin{smallmatrix}(0,0)\\(0,\frac{1}{2})\end{smallmatrix}\right](z,\Omega). \end{aligned} \tag{1}$$

3 Isogenies on Squared Kummer Surfaces

This section develops the Richelot isogenies on squared Kummer surfaces, extending the isogenies between supersingular Kummer surfaces. Meanwhile, we describe the dual $(2,2)$-isogeny between squared Kummer surfaces. We compare our scheme with the method, presented by Cosset [11], to compute (ℓ,ℓ)-isogenies between generic Jacobians in the end.

3.1 Squared Kummer Surfaces

The Kummer surface is a surface in projective space $\mathbb{P}^3$ as the quotient of some genus-2 Jacobian J by ± 1. From the view of cryptography, although points on the Kummer surface do not form a group, it is a pseudo-group having arithmetic as x-coordinates used in the Montgomery curve.

Improved from the initial work of Chudnovsky [9], Gaudry [25] proposed Kummer surface $\mathcal{K}$:

$$\begin{aligned} \mathcal{K} : 2E_0X_1X_2X_3X_4 = X_1^4 + X_2^4 + X_3^4 + X_4^4 - F_0(X_1^2X_4^2 + X_2^2X_3^2) - \\ G_0(X_1^2X_3^2 + X_2^2X_4^2) - H_0(X_1^2X_2^2 + X_3^2X_4^2). \end{aligned} \tag{2}$$

We write (X_1, X_2, X_3, X_4) as the projective coordinates of a point on $\mathcal{K}$, that is

$$X_1 = \lambda\theta_1(z),\ X_2 = \lambda\theta_2(z),\ X_3 = \lambda\theta_3(z),\ X_4 = \lambda\theta_4(z)$$

for some z in $\mathbb{C}^2$ and $\lambda \in \mathbb{C}^*$. Besides, E_0, F_0, G_0, H_0 are constants parametrized by $a = \theta_1(0), b = \theta_2(0), c = \theta_3(0), d = \theta_4(0)$ as

$$\begin{aligned} E_0 &= -256abcdA^2B^2C^2D^2/(a^2d^2 - b^2c^2)(a^2c^2 - b^2d^2)(a^2b^2 - c^2d^2); \\ F_0 &= (a^4 - b^4 - c^4 + d^4)/(a^2d^2 - b^2c^2); \\ G_0 &= (a^4 - b^4 + c^4 - d^4)/(a^2c^2 - b^2d^2); \\ H_0 &= (a^4 + b^4 - c^4 - d^4)/(a^2b^2 - c^2d^2) \end{aligned}$$

with

$$\begin{aligned} 4A^2 &= a^2 + b^2 + c^2 + d^2; \\ 4B^2 &= a^2 + b^2 - c^2 - d^2; \\ 4C^2 &= a^2 - b^2 + c^2 - d^2; \\ 4D^2 &= a^2 - b^2 - c^2 + d^2. \end{aligned}$$

Following the aforementioned works, fast Kummer surface arithmetic has been exploited. Since all arithmetic can be computed entirely with their squares $\theta_1^2(z), \theta_2^2(z), \theta_3^2(z), \theta_4^2(z)$ in [25], the squared Kummer surface proposed in [9] has been suggested for high-speed implementation.

Definition of Squared Kummer Surfaces. The notations of squared Kummer surfaces are different in the literature [37, Table 1], and we use the notations in [37]. Four fixed constants $\mu_1, \mu_2, \mu_3, \mu_4$ can be computed from the Rosenhain form $C_{\lambda,\mu,\upsilon}$ of an associated genus-2 curve [3]

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

The squared Kummer surface $\mathcal{K}^{Sqr}$ is defined as

$$\mathcal{K}^{Sqr} : EX_1X_2X_3X_4 = \begin{pmatrix} X_1^2 + X_2^2 + X_3^2 + X_4^2 - F(X_1X_4 + X_2X_3) \\ -G(X_1X_3 + X_2X_4) - H(X_1X_2 + X_3X_4) \end{pmatrix}^2 \tag{4}$$

where

$$E = 4\frac{\mu_1\mu_2\mu_3\mu_4(2\hat{\mu}_1)^2(2\hat{\mu}_2)^2(2\hat{\mu}_3)^2(2\hat{\mu}_4)^2}{(\mu_1\mu_4 - \mu_2\mu_3)^2(\mu_1\mu_3 - \mu_2\mu_4)^2(\mu_1\mu_2 - \mu_3\mu_4)^2},$$

$$F = 2\frac{\mu_1\mu_4 + \mu_2\mu_3}{\mu_1\mu_4 - \mu_2\mu_3}, \quad G = 2\frac{\mu_1\mu_3 + \mu_2\mu_4}{\mu_1\mu_3 - \mu_2\mu_4}, \quad H = 2\frac{\mu_1\mu_2 + \mu_3\mu_4}{\mu_1\mu_2 - \mu_3\mu_4}.$$

with

$$\begin{aligned} 2\hat{\mu}_1 &= \mu_1 + \mu_2 + \mu_3 + \mu_4, \\ 2\hat{\mu}_2 &= \mu_1 + \mu_2 - \mu_3 - \mu_4, \\ 2\hat{\mu}_3 &= \mu_1 - \mu_2 + \mu_3 - \mu_4, \\ 2\hat{\mu}_4 &= \mu_1 - \mu_2 - \mu_3 + \mu_4. \end{aligned}$$

Elements on $\mathcal{K}^{Sqr}$ are projective points $(X_1, X_2, X_3, X_4) \in \mathbb{P}^3$ satisfying equation (4), and the identity point is $\mathcal{O}_{\mathcal{K}} = (\mu_1, \mu_2, \mu_3, \mu_4)$, where squared Kummer surface can be recovered. The doubling algorithm, differential algorithm, as well as scalar multiplication algorithm on squared Kummer surfaces are summarized in [3, Section 5.2]. In addition, one can obtain three Rosenhain invariants λ, μ, υ via

$$\lambda = \frac{\mu_1\mu_3}{\mu_2\mu_4}, \quad \mu = \frac{\mu_3\left(1 + \sqrt{\frac{\hat{\mu}_3\hat{\mu}_4}{\hat{\mu}_1\hat{\mu}_2}}\right)}{\mu_4\left(1 - \sqrt{\frac{\hat{\mu}_3\hat{\mu}_4}{\hat{\mu}_1\hat{\mu}_2}}\right)}, \quad \upsilon = \frac{\mu_1\left(1 + \sqrt{\frac{\hat{\mu}_3\hat{\mu}_4}{\hat{\mu}_1\hat{\mu}_2}}\right)}{\mu_2\left(1 - \sqrt{\frac{\hat{\mu}_3\hat{\mu}_4}{\hat{\mu}_1\hat{\mu}_2}}\right)}.$$

The map from $\mathcal{K}$ to J_C is originally given by the theta functions in [25] and is applied for squared Kummer surface in [3,10]. We can solve a system of equations to acquire the coordinates of a divisor on the Jacobian for the inverse direction.

The dual squared Kummer surface $\hat{\mathcal{K}}^{Sqr}$ of a squared Kummer surface $\mathcal{K}^{Sqr}$, introduced in [37], is a squared Kummer surface with identity point $(\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3, \hat{\mu}_4)$. This map constitutes three sub-operations. The Hadamard transform $\mathcal{H} : \mathbb{P}^3 \to \mathbb{P}^3$ is defined as

$$\mathcal{H} : (x_1, x_2, x_3, x_4) \mapsto \begin{pmatrix} x_1 + x_2 + x_3 + x_4, \\ x_1 + x_2 - x_3 - x_4, \\ x_1 - x_2 + x_3 - x_4, \\ x_1 - x_2 - x_3 + x_4 \end{pmatrix}.$$

The coordinate square operation $\mathcal{S} : \mathbb{P}^3 \to \mathbb{P}^3$ is

$$\mathcal{S} : (x_1, x_2, x_3, x_4) \mapsto (x_1^2, x_2^2, x_3^2, x_4^2).$$

The last operation is the coordinate scaling operation $\mathcal{C}_{(d_1,d_2,d_3,d_4)} : \mathbb{P}^3 \to \mathbb{P}^3$

$$\mathcal{C}_{(d_1,d_2,d_3,d_4)} : (x_1, x_2, x_3, x_4) \mapsto (x_1/d_1, x_2/d_2, x_3/d_3, x_4/d_4).$$

The figure 1 from [37] constructs the Richelot isogeny and its dual isogeny. Intermediate Kummer surface $\mathcal{K}^{Int}$ is defined in [37], and canonical Kummer surface $\mathcal{K}^{Can}$ is the same as $\mathcal{K}$ in equation (2). Besides, the map $\mathcal{C}$ is the abbreviation for $\mathcal{C}_{(a,b,c,d)}$, where

$$a^2 = \mu_1, \quad b^2 = \mu_2, \quad c^2 = \mu_3, \quad d^2 = \mu_4,$$

and it is analogous to $\hat{\mathcal{C}}$.

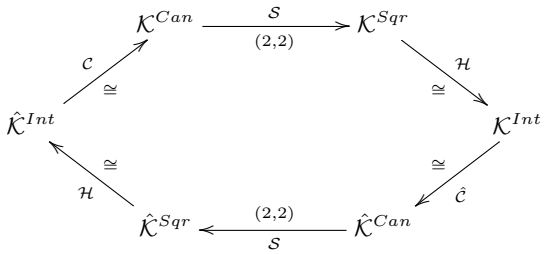

Fig. 1. Richelot Isogeny on Dual Squared Kummer Surface and Its Dual.

With the notations above, the map from $\mathcal{K}^{Sqr}$ to $\hat{\mathcal{K}}^{Sqr}$ is denoted as

$$\begin{aligned} \varphi : \mathcal{K}^{Sqr} &\to \hat{\mathcal{K}}^{Sqr} \\ \overline{P} &\mapsto (\mathcal{S} \circ \mathcal{C}_{(\hat{a},\hat{b},\hat{c},\hat{d})} \circ \mathcal{H})(\overline{P}). \end{aligned}$$

The only difference from $\hat{\mathcal{K}}^{Sqr}$ to $\mathcal{K}^{Sqr}$ is the parameters of the map $\mathcal{C}$.

3.2 Richelot Isogenies

From what we have discussed, the map between squared Kummer surfaces and its dual is well-defined, hence the next issue is to construct the generic isogeny between squared Kummer surfaces.

Since the points on Kummer surface are not group, the isogenies should be modified. Due to the relations between Kummer surfaces and their corresponding Jacobian, it is direct to define isogenies on Kummer surfaces to be the isogenies induced by their corresponding Jacobians.

Definition 1. *For Kummer surfaces $\mathcal{K}_1$ and $\mathcal{K}_2$, they are isogenous if there exists an isogenous map between their corresponding Jacobian J_1 and J_2.*

The Richelot isogeny, whose kernel is a maximal 2-Weil isotropic subgroup of $J[2]$, is the most simple isogeny on Jacobians. Therefore, we consider the Richelot isogenies between Kummer surfaces induced by their Jacobians, and the kernel consists four different points. For instance, the map between $\mathcal{K}^{Sqr}$ and $\hat{\mathcal{K}}^{Sqr}$ is Richelot isogeny with kernel

$$O = \Big((\mu_1, \mu_2, \mu_3, \mu_4), (\mu_2, \mu_1, \mu_4, \mu_3), (\mu_3, \mu_4, \mu_1, \mu_2), (\mu_4, \mu_3, \mu_2, \mu_1)\Big).$$

In 2018, Costello [12] proposed an approach to compute Richelot isogenies between special Kummer surfaces. Following his work, we extend this approach to generic squared Kummer surfaces.

For the Jacobian J, there are 15 maximal 2-Weil isotropic subgroups and one of which is O as the kernel of isogeny to the dual Kummer surface. Let P_1, P_2 be two points in J associated with O, there are 8 maximal isotropic subgroups that can generate $J[2]$ with P_1, P_2, and we choose the following two special sets as the kernel of Richelot isogenies.

Let $(\alpha, \tilde{\alpha})$, $(\beta, \tilde{\beta})$, $(\gamma, \tilde{\gamma})$ be the roots of $x^2 - Fx + 1$, $x^2 - Gx + 1$, $x^2 - Hx + 1$, respectively. Then we can define two sets in $\mathcal{K}^{Sqr}$ as

$$\Upsilon = ((\mu_1, \mu_2, \mu_3, \mu_4),\ (1, 0, 0, \alpha),\ (1, 0, \beta, 0),\ (1, \gamma, 0, 0))$$

and

$$\tilde{\Upsilon} = \Big((\mu_1, \mu_2, \mu_3, \mu_4),\ (1, 0, 0, \tilde{\alpha}),\ (1, 0, \tilde{\beta}, 0),\ (1, \tilde{\gamma}, 0, 0)\Big),$$

which correspond two maximal isotropic groups on the Jacobian.

We intend to deduce the Richelot isogenies whose kernels are Υ or $\tilde{\Upsilon}$. Our fundamental idea is to seek an isomorphic surface such that the points in the image of Υ or $\tilde{\Upsilon}$ under map can play the same role as O for φ_O, and the other points in two other sets act as points in $\{\Upsilon, \tilde{\Upsilon}\}$.

Since operations $\mathcal{H}, \mathcal{C}_{(d_1,d_2,d_3,d_4)}$ are isomorphic operations, we can analyze the subgroups in $\hat{\mathcal{K}}^{Can}$ after two steps around the hexagon in figure 1. Writing O^{Can} as the image of O after two operations in the hexagon, and similar to Υ^{Can} and $\tilde{\Upsilon}^{Can}$, reveals as

$$\begin{aligned}
O^{Can} &= \Big((\hat{a}, \hat{b}, \hat{c}, \hat{d}),\ (\hat{a}, -\hat{b}, \hat{c}, -\hat{d}),\ (\hat{a}, -\hat{b}, -\hat{c}, \hat{d}),\ (\hat{a}, \hat{b}, -\hat{c}, -\hat{d})\Big);\\
\Upsilon^{Can} &= \Big((\hat{a}, \hat{b}, \hat{c}, \hat{d}),\ (\hat{d}, \hat{c}, \hat{b}, \hat{a}),\ (\hat{c}, \hat{d}, \hat{a}, \hat{b}),\ (\hat{b}, \hat{a}, \hat{d}, \hat{c})\Big);\\
\tilde{\Upsilon}^{Can} &= \Big((\hat{a}, \hat{b}, \hat{c}, \hat{d}),\ (\hat{d}, -\hat{c}, -\hat{b}, \hat{a}),\ (\hat{c}, -\hat{d}, \hat{a}, -\hat{b}),\ (\hat{b}, \hat{a}, -\hat{d}, -\hat{c})\Big),
\end{aligned}$$

where $(\hat{a}, \hat{b}, \hat{c}, \hat{d}) = \mathcal{H}(a, b, c, d)$.

All points in O^{Can} turn to identity point $\mathcal{O}_{\hat{\mathcal{K}}^{Sqr}} = (\hat{\mu}_1, \hat{\mu}_2, \hat{\mu}_3, \hat{\mu}_4)$ after the coordinate square map $\mathcal{S}$, i.e., $\mathcal{S}(O^{Can}) = \mathcal{O}_{\hat{\mathcal{K}}^{Sqr}}$, so the points in the image of Υ^{Can} or $\tilde{\Upsilon}^{Can}$ under the desired map should be identical after map $\mathcal{S}$.

If Υ is the intended kernel, the Hadamard transform $\mathcal{H}$ is what we look forward to, and the image of Υ^{Can} under $\mathcal{H}$ is a $(2,2)$-subgroup

$$O_1^{Can} = ((a_1, b_1, c_1, d_1),\ (a_1, -b_1, -c_1, d_1),\ (a_1, -b_1, c_1, -d_1),\ (a_1, -b_1, -c_1, d_1))$$

in $\mathcal{K}_1^{Can}$ whose elements act as the elements in O^{Can}, where $(a_1, b_1, c_1, d_1) = \mathcal{H}(\hat{a}, \hat{b}, \hat{c}, \hat{d})$.

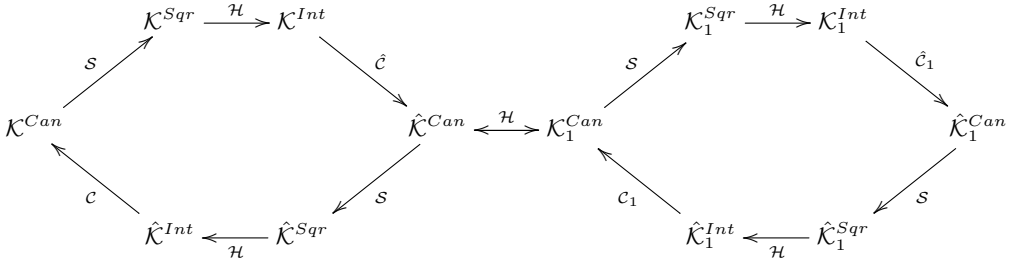

Fig. 2. Richelot Isogeny with Kernel Υ

After the Hadamard transform $\mathcal{H}$ called the *conjunction map*, we move to a new Kummer surface different from $\mathcal{K}^{Can}$ and $\hat{\mathcal{K}}^{Can}$. Meanwhile, the points in the sets $\mathcal{H}(O^{Can})$ and $\mathcal{H}(\tilde{\Upsilon}^{Can})$ can act as those in two subgroups Υ^{Can} and $\tilde{\Upsilon}^{Can}$ on the isomorphic Kummer surface. Then, we can obtain isogenous squared Kummer surface $\mathcal{K}_1^{Sqr}$ immediately after the coordinate square action $\mathcal{S}$, and the conjunction map in figure 2 connects two hexagons.

Let $\overline{P} \in \mathcal{K}^{Sqr}$ be a point of odd order ℓ, the full Richelot isogeny with kernel Υ is defined as

$$\begin{aligned} \varphi_\Upsilon : \mathcal{K}^{Sqr} &\to \mathcal{K}_1^{Sqr} \\ \overline{P} &\mapsto \mathcal{S} \circ \mathcal{H} \circ \mathcal{C} \circ \mathcal{H}(\overline{P}). \end{aligned}$$

In practice, the target surface $\mathcal{K}_1^{Sqr}$ can be recovered from $\mathcal{O}_{\mathcal{K}_1^{Sqr}}$ as the image of $\mathcal{O}_{\mathcal{K}^{Sqr}}$.

When the $(2,2)$-subgroup is $\tilde{\Upsilon}$, the case is a little tough. To this issue, a modified transform

$$\tilde{\mathcal{H}} : (X_1, X_2, X_3, X_4) \mapsto \mathcal{H}(-X_1, X_2, X_3, X_4)$$

is introduced. Replacing the conjunction operation $\mathcal{H}$ with $\tilde{\mathcal{H}}$, we obtain a new squared Kummer surface $\tilde{\mathcal{K}}^{Sqr}$.

One thing we should take care of is the map $\mathcal{C}$ using square roots of $\hat{\mu}_i$ for $i = 1, 2, 3, 4$. Since operations are all in a projective plane, fixed $\hat{d}$ as 1, there are eight possible surfaces after four fundamental operations when different square roots are chosen. The identity points of isogenous squared Kummer surfaces are

$$\begin{aligned} &\mathcal{O}_1 = (a_1^2, b_1^2, c_1^2, d_1^2),\ \mathcal{O}_2 = (c_1^2, d_1^2, a_1^2, b_1^2),\ \mathcal{O}_3 = (d_1^2, c_1^2, b_1^2, a_1^2),\ \mathcal{O}_4 = (b_1^2, a_1^2, d_1^2, c_1^2), \\ &\mathcal{O}_5 = (a_2^2, b_2^2, c_2^2, d_2^2),\ \mathcal{O}_6 = (c_2^2, d_2^2, a_2^2, b_2^2),\ \mathcal{O}_7 = (d_2^2, c_2^2, b_2^2, a_2^2),\ \mathcal{O}_8 = (b_2^2, a_2^2, d_2^2, c_2^2), \end{aligned}$$

with $(a_2, b_2, c_2, d_2) = \tilde{\mathcal{H}}(\hat{a}, \hat{b}, \hat{c}, \hat{d})$.

The surfaces defined by identities $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4$ are isomorphic by a permutation of (X_1, X_2, X_3, X_4) and it is the same for the surfaces with identity points $\mathcal{O}_5, \mathcal{O}_6, \mathcal{O}_7, \mathcal{O}_8$, so there are only two disparate isogenous squared Kummer surfaces up to isomorphism. In addition, for fixed choice $(\hat{a}, \hat{b}, \hat{c}, \hat{d})$, if $\mathcal{O}_1$ is the identity point, the surface determined by $\mathcal{O}_5$ can be obtained after replacing the conjunction operation $\mathcal{H}$ with $\tilde{\mathcal{H}}$, and $\mathcal{O}_5$ can be regarded as the image of Richelot isogeny with kernel $\tilde{\Upsilon}$.

From what we have discussed, we get that

Theorem 1. *Let $\mathcal{K}^{Sqr}$ be a squared Kummer surface, and Υ, $\tilde{\Upsilon}$ be two sets on $\mathcal{K}^{Sqr}$, then we can get two isogenous Kummer surfaces $\mathcal{K}^{Sqr}_{\Upsilon}$, $\mathcal{K}^{Sqr}_{\tilde{\Upsilon}}$ with kernel Υ, $\tilde{\Upsilon}$ respectively.*

After the full Richelot isogeny map, we move to the hexagon of $\mathcal{K}_1^{Sqr}$ and its dual. Notice $\varphi_{\Upsilon}(O) = \varphi_{\Upsilon}(\tilde{\Upsilon}) = O_1$, where O_1 is the kernel of Richelot map from $\mathcal{K}_1^{Sqr}$ to its dual $\hat{\mathcal{K}}_1^{Sqr}$. Therefore, the new sets Υ_1 and $\tilde{\Upsilon}_1$ can be constructed by solving three quadratic equations. The same operations act on the group Υ_1 or $\tilde{\Upsilon}_1$, then a new squared Kummer surface $\mathcal{K}_2^{Sqr}$ is acquired. As a result, a different hexagon links the second hexagon at $\hat{\mathcal{K}}_1^{Can}$ in figure 2.

The same operations can be done repeatedly, i.e., we could glue a new hexagon at the end of the hexagon, so we obtain a chain of squared Kummer surfaces as the SIDH-style computations in elliptic curves. In other words, a chain of hexagons is linked up by the Hadamard transform $\mathcal{H}$.

3.3 Dual Isogenies

For the isogeny between two different Kummer surfaces, the dual isogeny is essential for cryptographic implementations. The squared Kummer surface $\mathcal{K}_1^{Sqr}$ is the isogenous surface we obtained in figure 2, and we require to come back to the initial surface $\mathcal{K}^{Sqr}$ for the dual in some situations.

In the beginning, with the technique in Section 3.1, the dual squared Kummer surface $\hat{\mathcal{K}}_1^{Sqr}$ is obtained via the map φ_{O_1}. Indeed, the kernel of map from $\mathcal{K}_1^{Sqr}$ to $\hat{\mathcal{K}}_1^{Int}$ contains $O, \Upsilon, \tilde{\Upsilon}$, which is $J[2]$ on corresponding Jacobian, therefore $\mathcal{K}_1^{Sqr}$ is isomorphic to $\mathcal{K}$.

After Hadamard transform, we move to intermediate Kummer surface $\hat{\mathcal{K}}_1^{Int}$. By carefully choosing the square roots of $(a_1^2, b_1^2, c_1^2, d_1^2)$, corresponding to the identity point (a_1, b_1, c_1, d_1) in $\mathcal{K}_1^{Can}$, the canonical Kummer surface $\mathcal{K}_1^{Can}$ we acquired in the hexagon can be obtained by $\mathcal{C}_{(a_1,b_1,c_1,d_1)}$.

Now, the Hadamard isomorphism $\mathcal{H}$ takes us back to the right hexagon in figure 2. The remained operations can be done in the hexagon of the initial surface $\mathcal{K}^{Sqr}$ and its dual surface $\hat{\mathcal{K}}^{Sqr}$. Then, $\mathcal{K}^{Sqr}$ is acquired after the several maps in the left hexagon in figure 2.

To sum up, for the dual isogeny $\hat{\varphi}_{\Upsilon}$, we construct the isogeny from $\hat{K}_1^{Sqr}$ to $\hat{K}^{Sqr}$ along with two dual squared Kummer surfaces computations, i.e.,

$$\begin{aligned}\hat{\varphi}_{\Upsilon} : \mathcal{K}_1^{Sqr} &\to \mathcal{K}^{Sqr} \\ \overline{P} &\mapsto \varphi_{\hat{O}} \circ \varphi_{\hat{\Upsilon}_1} \circ \varphi_{O_1}(\overline{P}),\end{aligned}$$

where $\hat{\Upsilon}_1$ is the cautiously chosen $(2,2)$-subgroup in $\hat{\mathcal{K}}_1^{Sqr}$.

For the chain of Richelot isogenies from $\mathcal{K}_0^{Sqr}$ to $\mathcal{K}_n^{Sqr}$, we could deal with the isogenous chain from $\hat{\mathcal{K}}_n^{Sqr}$ to $\hat{\mathcal{K}}_0^{Sqr}$, and two more calculations are requisite at the beginning and in the end. For each step, the square roots must correspond to the identity point in $\mathcal{K}_i^{Can}$ which generates the i-th hexagon of the chain of hexagons for $i = n, \cdots, 1$.

3.4 Comparison

With some notations of van Wamelem [40], the correspondences between Mumford representations on Jacobians of genus two and sixteen theta functions of level 4 are given in [11, Appendix], then Cosset and Robert [11] presented the algorithm to compute the (ℓ,ℓ)-isogenies between Jacobians with theta functions. In their approach, the information about Jacobian of an abelian variety A can be translated into theta null point $(\theta_i(0))_{i\in\frac{1}{\ell}\mathbb{Z}^g/\mathbb{Z}^g}$.

To compute (ℓ,ℓ)-isogenies, the basis $\{e_1, e_2\}$ of a maximal isotropic subgroup $L \subset A[\ell]$ should be chosen from the Jacobian A. Writing the affine theta coordinates of e_1, e_2 and $e_1 + e_2$, all theta coordinates of points in L should be computed, and the operations of points on Jacobians only rely on Riemann relations using theta functions [26, 35].

Let N be a matrix of rank r such that ${}^tNN = \ell Id_r$, and $j = (k, 0\cdots, 0)N^{-1}$, then the theta null point of $B = A/L$ is evaluated by

$$\theta_k^B(0)\theta_0^B(0)\cdots\theta_0^B(0) = \sum_{\substack{t_1,\cdots,t_r\in L \\ (t_1,\cdots,t_r)N=(0,\cdots,0)}} \theta_{j_1}^A(t_1)\cdots\theta_{j_r}^A(t_r). \tag{5}$$

The Rosenhain form $C_{\lambda,\mu,\upsilon}$ is obtained from the theta constants.

For points on Jacobian, there is an algorithm to compute the image under map. Let $\tilde{P}'$ be any affine lift of point $P' \notin L$, and $(m_1, \cdots, m_r) = (\ell, 0, \cdots, 0)N^{-1}$. The image of a point under the above isogeny can be computed by

$$\theta_k^B(\ell\tilde{Q}')\theta_0^B(0)\cdots\theta_0^B(0) = \sum_{\substack{t_1,\cdots,t_r\in L \\ (t_1,\cdots,t_r)N=(0,\cdots,0)}} \theta_{j_1}^A(m_1\tilde{P}' + t_1)\cdots\theta_{j_r}^A(m_r\tilde{P}' + t_r). \tag{6}$$

Compared with Richelot isogenies between squared Kummer surfaces, the crucial difference is that their method handles odd ℓ, while Richelot isogenies compute $(2,2)$-isogenies. For the chain of isogenies, Richelot isogenies need to solve three quadratic equations, nevertheless, the (ℓ,ℓ)-isogenies must generate

a new maximal isotropic subgroup for the next isogeny, and obtain all theta coordinates in the subgroup, requiring much more storage space.

Suppose we require to compute the dual isogenies, Richelot isogenies only require determining three symbols of the square roots of the constants. However, when computing (ℓ,ℓ)-isogenies with kernel L, we must seek a new subgroup $L' \subset A[\ell]$ such that two subgroups generate $A[\ell]$, and calculate the image of L' under (ℓ,ℓ)-isogenies, which is supposed to be the kernel of dual isogeny. So the chain of isogenies needs much more resources to recover the initial curve for the (ℓ,ℓ)-isogenies.

4 Squared Pairings on Hyperelliptic Curves

Lubicz and Robert proposed an efficient algorithm [29] to compute pairings on abelian varieties. In their work, theta functions over complex field $\mathbb{C}$ are the fundamental tools. Based on their method, we renew pairings using the formula on squared Kummer surfaces in this section.

4.1 Squared Pairings

Let $A = \mathbb{C}^g/\Lambda_\Omega$ be the associated complex abelian variety with Λ_Ω, and denote $\pi : \mathbb{C}^g \to A$ the natural projection. For two ℓ-torsion points P, Q of an abelian variety A, writing with their theta coordinates, a formula to obtain Weil pairings is introduced.

Proposition 2 ([29]). *Let $\Omega \in \mathbb{H}_g$, and fix $a, b \in \mathbb{Q}^g$. Let ℓ be a positive integer and let $z_P, z_Q \in \mathbb{C}^g$ such that $\ell \cdot z_P = \ell \cdot z_Q = 0 \bmod \Lambda_\Omega$. Let $P, Q \in A$ satisfy $P = \pi(z_P), Q = \pi(z_Q)$ and set*

$$T(z_P, z_Q) = \frac{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](\ell \cdot z_P + z_Q, \Omega)}{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z_Q, \Omega)} \frac{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](0, \Omega)}{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](\ell \cdot z_P, \Omega)},$$

$$R(z_P, z_Q) = \frac{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](\ell \cdot z_Q + z_P, \Omega)}{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z_P, \Omega)} \frac{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](0, \Omega)}{\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](\ell \cdot z_Q, \Omega)}.$$

If $T(z_P, z_Q)$ and $R(z_P, z_Q)$ are well-defined and non zero, we have

$$e_\ell(P, Q) = T(z_P, z_Q)^{-1} \cdot R(z_P, z_Q). \tag{7}$$

For the Jacobian J of genus-2 curve over finite field $\mathbb{F}_q$, associated with $\Omega \in \mathbb{H}_g$, set $\Omega' = \frac{\Omega}{2}$, for any point $P \in A$, we have $\theta_i(P, \frac{\Omega}{2}) = \theta_i(P, \Omega')$ for $i \in \frac{1}{2}\mathbb{Z}^2/\mathbb{Z}^2$, corresponding to four fundamental theta functions in (1).

Four fundamental theta coordinates of P, Q and $P + Q$ along with identity point $\mathcal{O}$ can be translated into a point on the canonical Kummer surface $\mathcal{K}^{Can}$

in [25]. As a result, for $i = 1, 2, 3, 4$, $\theta_i(\ell P)$, $\theta_i(\ell Q)$, $\theta_i(\ell P + Q)$, $\theta_i(\ell Q + P)$ are calculated by the pseudo-additions on $\mathcal{K}^{Can}$. Thus Weil pairings turn to be

$$e_\ell(P,Q)^2 = \frac{\theta_i(P+\ell Q)}{\theta_i(Q+\ell P)}\frac{\theta_i(Q)}{\theta_i(P)}\frac{\theta_i(\ell Q)}{\theta_i(\ell P)} \tag{8}$$

in [29], and Weil pairings can be implemented with arithmetic on canonical Kummer surfaces only. In addition, computation of Tate pairings $t_\ell(P,Q)$ has similar formula which is exploited in [29].

The points on squared Kummer surface have coordinates

$$\left(\theta_1(z)^2, \theta_2(z)^2, \theta_3(z)^2, \theta_4(z)^2\right),$$

and the arithmetic on $\mathcal{K}^{Sqr}$ is well-defined. It is obvious that pairings can be computed with coordinates on squared Kummer surfaces.

For odd integer ℓ, unity group μ_ℓ has order ℓ, and the square map $\tau : x \mapsto x^2$ induces an isomorphism on μ_ℓ, therefore we adopt the *squared Weil pairings* defined as

$$e'_\ell(P,Q) = \tau^2(e_\ell(P,Q)) = e_\ell(P,Q)^4.$$

The formula to compute squared Weil pairings becomes

$$e'_\ell(P,Q) = \frac{\theta_i(P+\ell Q)^2}{\theta_i(Q+\ell P)^2}\frac{\theta_i(Q)^2}{\theta_i(P)^2}\frac{\theta_i(\ell Q)^2}{\theta_i(\ell P)^2}. \tag{9}$$

All the theta coordinates in (9) are the i-th coordinates of points on the squared Kummer surface. Moreover, as discussed in [29], any affine lifts of points $P, Q, P+Q$ and identity point $\mathcal{O}$ do not influence the values of Weil pairings, so the result is independent with the choice of lifts.

Theorem 2. *The squared Weil pairing is a non-degenerate bilinear map.*

Proof. First, a permutation is induced by square map τ on μ_ℓ, hence e'_ℓ is not degenerate since the original Weil pairing e_ℓ is non-degenerate.

For $P, Q, R \in J$, we have

$$\begin{aligned} e'_\ell(P, Q+R) &= e_\ell(P, Q+R)^4 \\ &= e_\ell(P,Q)^4 e_\ell(P,R)^4 \\ &= e'_\ell(P,Q)e'_\ell(P,R), \end{aligned}$$

and in the same way, we obtain

$$e'_\ell(P+Q, R) = e'_\ell(P,R)e'_\ell(Q,R),$$

therefore, squared Weil pairing is a bilinear map. □

The Weil pairings on the isogenous curves have the following property.

Theorem 3. *Let $\varphi : J_1 \to J_2$ be an isogenous map, and $\hat{\varphi}$ be its dual isogeny. Then for all ℓ-torsion points $P \in J_1[\ell]$ and $Q \in J_2[\ell]$, we have*

$$e_\ell\left(P, \hat{\varphi}(Q)\right) = e_\ell\left(\varphi(P), Q\right). \tag{10}$$

Similarly, for the squared Weil pairings, we have

$$e'_\ell\left(P, \hat{\varphi}(Q)\right) = e'_\ell\left(\varphi(P), Q\right). \tag{11}$$

Proof. The first equation comes from [33, P. 186]. After two squares on both sides of the equation (10), we obtain the second equation. □

The Tate pairings have the same properties as the Weil pairings, therefore, we omit the analysis.

From what we have discussed, the squared Weil pairing on Jacobian is a well-defined non-degenerate bilinear map, which can be implemented with coordinates on squared Kummer surfaces. Unfortunately, it is hard to define the pairings on Kummer surfaces, and we make some attempts in the next subsection.

4.2 Squared Pairings on Squared Kummer Surfaces

Apart from nodes (points of order two), a point $\overline{P}$ on Kummer variety represents the quotient of a point P and opposite point $-P$ on corresponding Jacobian J. Therefore, the definition of pairings on Kummer surfaces should be modified.

Let A be an abelian variety defined by $\Omega \in \mathbb{H}_g$, and the Kummer surface associated with A is deduced from the level 2 theta functions defined by Ω. Let $\xi : A \to \mathcal{K}$ be the natural projection, and $\overline{P}$ is denoted as the image of points $\pm P$ under ξ. Although $\mathcal{K}$ does not preserve the group structure of A, $\mathcal{K}$ inherits from A of scalar-multiplication on its points which can be accomplished by Riemann relations on Kummer surfaces. In addition, it is well-known that we can compute $\overline{P+Q}$ from the knowledge of $\overline{P}, \overline{Q}, \overline{P-Q}$.

Let e be a well-defined pairing on A over field K, e.g., Weil pairing or Tate pairing, and let $\overline{K}_0^*$ be the quotient of $\overline{K}^*$ by the action of the automorphism -1. Set the natural projection by $\xi_0 : \overline{K}^* \to \overline{K}_0^*$, then the pairing e induces a well-defined application

$$\begin{aligned} \overline{e} : \mathcal{K}(\overline{K}) \times \mathcal{K}(\overline{K}) &\to \overline{K}_0^* \\ (\overline{P}, \overline{Q}) &\mapsto \xi_0(e(P,Q)). \end{aligned}$$

We observe that there is a bijection from $\overline{K}_0^*$ to the set $S = \{x + \frac{1}{x} \mid x \in \overline{K}^*\}$, hence we have another expression of pairing $\overline{e}$ defined as

$$\overline{e} : (\overline{P}, \overline{Q}) \mapsto e(P,Q) + e(-P,Q).$$

This idea has been introduced first in [24] to compute pairings only with x-coordinates, then an efficient algorithm to compute the symmetric pairings is exhibited on Kummer varieties with theta functions [29].

On Kummer surfaces, the result of differential addition of two points $\overline{P}$ and $\overline{Q}$ is not determined, because there are two possible points $\overline{P+Q}$ and $\overline{P-Q}$ which are the images of $P+Q$ and $P-Q$. For the pairing $\overline{e}$ on Kummer surface, we have

Theorem 4. *Let $\overline{P}, \overline{Q}, \overline{R}$ be three points on the Kummer surface, and $\overline{e}$ be the pairing on Kummer surface, then we have*

$$\overline{e}(\overline{P}, \overline{R}) \cdot \overline{e}(\overline{Q}, \overline{R}) = \overline{e}(\overline{P+Q}, \overline{R}) + \overline{e}(\overline{P-Q}, \overline{R}), \tag{12}$$

and

$$\overline{e}(\overline{P}, \overline{Q}) \cdot \overline{e}(\overline{P}, \overline{R}) = \overline{e}(\overline{P}, \overline{Q+R}) + \overline{e}(\overline{P}, \overline{Q-R}). \tag{13}$$

Proof. Let P, Q, R be the preimage of $\overline{P}, \overline{Q}, \overline{R}$ under ξ, and e be the pairing on the abelian variety associated with Kummer surfaces. We have

$$\begin{aligned}
&\overline{e}(\overline{P}, \overline{R}) \cdot \overline{e}(\overline{Q}, \overline{R}) \\
=&(e(P,R) + e(-P,R)) \cdot (e(Q,R) + e(-Q,R)) \\
=&e(P+Q,R) + e(-P-Q,R) + e(P-Q,R) + e(-P+Q,R) \\
=&\overline{e}(\overline{P+Q}, \overline{R}) + \overline{e}(\overline{P-Q}, \overline{R}).
\end{aligned}$$

The second equation is obtained in the same way. □

Therefore, symmetric pairing is not a bilinear map, but it can represent the value of pairings on Kummer surfaces under the projective map ξ. Moreover, we can compute the $\mathbb{Z}$-action on these symmetric pairings.

As a consequence, for the identity point $\mathcal{O}_{\mathcal{K}}$, one can obtain that

$$\overline{e}(\overline{P}, \mathcal{O}_{\mathcal{K}}) = e(P, \mathcal{O}_{\mathcal{K}}) + e(-P, \mathcal{O}_{\mathcal{K}}) = 2,$$

and

$$\overline{e}(\mathcal{O}_{\mathcal{K}}, \overline{P}) = 2e(\mathcal{O}_{\mathcal{K}}, \overline{P}) = 2.$$

At the same time, we acquire the following result

$$\overline{e}(\overline{P}, \overline{Q})^2 = \overline{e}(2\overline{P}, \overline{Q}) + 2 = \overline{e}(\overline{P}, 2\overline{Q}) + 2.$$

Squared Symmetric Pairings. The symmetric pairings can be computed by the theta functions, as the addition of two pairings. On squared Kummer surfaces, we define the squared symmetric pairing, as a new *self-contained* symmetric pairing, to be implemented with pseudo-group operations on the same surfaces.

Definition 2. *Let $\overline{P}$ and $\overline{Q}$ be two points of order ℓ on squared Kummer surfaces. Let P and Q be corresponding points on the Jacobian. Then the squared symmetric pairings are defined as*

$$\overline{e}'(\overline{P}, \overline{Q}) = e'(P, Q) + e'(-P, Q), \tag{14}$$

Immediately, we have

$$\overline{e}'(\overline{P},\overline{Q}) = e(P,Q)^4 + e(-P,Q)^4. \tag{15}$$

From equation (9), $e'(P,Q)$ and $e'(-P,Q)$ can be implemented with coordinates $\left(\theta_1(z)^2,\theta_2(z)^2,\theta_3(z)^2,\theta_4(z)^2\right)$. Therefore, the computation of squared symmetric pairing only relies on arithmetic on squared Kummer surface.

The squared symmetric pairing should represent the information of the symmetric pairings, and it is true for squared Kummer surface over a finite field $\mathbb{F}_q$. For the pairings, let ℓ be an odd integer, and k be the embedding degree, which reveals $\mu_\ell \subset \mathbb{F}_{q^k}$. Then the set S becomes $S_\ell = \{x + \frac{1}{x} \in \mathbb{F}_{q^k} \mid x \in \mu_\ell\}$.

Lemma 1. *The map $\sigma : x \mapsto x^2 - 2$ over finite field $\mathbb{F}_{q^k}$ induces a bijection from S_ℓ to itself.*

Proof. Let $x \in \mu_\ell$ be one unity root, then

$$\left(x + \frac{1}{x}\right)^2 - 2 = x^2 + \frac{1}{x^2} \in S_\ell,$$

so the map is well-defined.

For any $x \in \mu_\ell$, since ℓ is odd, we obtain

$$\left(x^{\frac{1+\ell}{2}} + \frac{1}{x^{\frac{1+\ell}{2}}}\right)^2 - 2 = x^{1+\ell} + \frac{1}{x^{1+\ell}} = x + \frac{1}{x},$$

therefore, σ is surjective. The set is finite, and σ is an injection, which leads to an bijection from S_ℓ to itself. □

For the pairing $\overline{e}'$ on squared Kummer surface over $\mathbb{F}_q$, we have

$$\overline{e}'(\overline{P},\overline{Q}) = \sigma^2\left(\overline{e}(\overline{P},\overline{Q})\right)$$

for ℓ-torsion points $\overline{P},\overline{Q} \in \mathcal{K}^{Sqr}$. Therefore, squared symmetric pairing can be regarded as a variant of symmetric pairing which is implemented with squared Kummer surfaces totally.

Remark 1. The canonical Kummer surfaces play the same role as squared Kummer surfaces in the hexagon. Therefore, for $\overline{P}_1,\overline{Q}_1$ on canonical Kummer surface $\mathcal{K}^{Can}$, the squared symmetric pairings could be defined as

$$\tilde{e}'(\overline{P}_1,\overline{Q}_1) = e(P_1,Q_1)^2 + e(-P_1,Q_1)^2,$$

where P_1, Q_1 are corresponding points of $\overline{P}_1,\overline{Q}_1$ on Jacobians. It is self-contained as well.

In a word, we have introduced a self-contained symmetric pairing $\overline{e}'$ on squared Kummer surfaces.

5 Applications

Two applications based on squared Kummer surfaces are introduced in this section. One is the Verifiable Delay Functions proposed in [1], and another is the Delay Encryption introduced in [5].

5.1 Verifiable Delay Functions

A VDF contains three algorithms:

1. $\mathsf{Setup}(\lambda, T) \rightarrow (ek, vk)$: is an algorithm whose inputs are the security parameter λ and a delay parameter T. The outputs are an evaluation key ek and a verification key vk.
2. $\mathsf{Eval}(ek, s) \rightarrow (a, \tau)$: is a procedure to evaluate on input s. The outputs consist of a from s, and a (possibly empty) proof τ. The requirement of this procedure is the time of computation can not be less than T.
3. $\mathsf{Verify}(vk, s, a, \tau) \rightarrow \{\text{True, False}\}$: is a procedure to verify that a is the correct output for s, with the help of proof τ if necessary.

The VDF should satisfy three security properties: *Correctness*, stating that an honest evaluator never fails verification, *Soundness*, stating that a lying evaluator passes the verification with negligible probability, and *Sequentiality*, stating that it is impossible to correctly evaluate the VDF in time less than $T - o(T)$, even when using $poly(T)$ parallel processors. For more detailed discussions, please refer to [1,15].

Our VDF Scheme. Let $\overline{e}'$ be a well-defined squared symmetric Weil pairing on the squared Kummer surfaces. Our VDF is described as follows:

$\mathsf{Setup}(\lambda, T)$

1. Choose primes p, ℓ, according to the security parameter λ;
2. Select a genus two hyperelliptic curve $C_{\lambda,\mu,\upsilon}$ and convert it to a squared Kummer surface $\mathcal{K}_1^{Sqr}$ with the identity point $\mathcal{O}_1$;
3. Select a point P of order ℓ, and convert it to the point $\overline{P}$ on $\mathcal{K}_1^{Sqr}$;
4. Compute T Richelot isogenies $\varphi : \mathcal{K}_1^{Sqr} \rightarrow \mathcal{K}_2^{Sqr}$ and $\mathcal{O}_2 = \varphi(\mathcal{O}_1), \varphi(\overline{P})$;
5. Output $(ek, vk) = (\varphi, (\mathcal{O}_1, \mathcal{O}_2, \overline{P}, \varphi(\overline{P})))$.

$\mathsf{Eval}(\varphi, \overline{Q} \in \mathcal{K}_2^{Sqr})$

1. Compute and output $\hat{\varphi}(\overline{Q})$.

$\mathsf{Verify}(\mathcal{K}_1^{Sqr}, \mathcal{K}_2^{Sqr}, \overline{P}, \overline{Q}, \varphi(\overline{P}), \hat{\varphi}(\overline{Q}))$

1. Verify that $\hat{\varphi}(\overline{Q})$ is a point on $\mathcal{K}_2^{Sqr}$;
2. Verify that $\overline{e}'_1(\overline{P}, \hat{\varphi}(\overline{Q})) = \overline{e}'_2(\varphi(\overline{P}), \overline{Q})$.

In the above scheme, ℓ is a prime, and we use the Kummer surface $\mathcal{K}_1^{Sqr}$ over finite field $\mathbb{F}_p$. For T Richelot isogenies, the kernel of O induces a dual Richelot isogeny, and via choosing different square roots, $\Upsilon, \hat{\Upsilon}$ can exchange their role, so we suppose to use the same kernel Υ of the intermediate Kummer surfaces.

As for the dual isogeny, the choice of square roots must be equal to what we require, which needs more information to be determined, and our method is to record the symbols of the identity point before the map $\mathcal{S}$ when computing the isogenies. The output ek contains the information about the identity point of $\mathcal{K}_i^{Can}$, e,g, which coordinates are quadratic residue, so that we can obtain the right Kummer surface.

Remark 2. The scheme can be converted to the Jacobians of genus-2 hyperelliptic curves, using the techniques (ℓ, ℓ)-isogenies proposed in [11] and squared pairings in Section 4.1.

Below we prove that our new VDF scheme satisfies the security requirements.

Theorem 5. *The VDF scheme above is correct and sound.*

Proof. Considering equation (11), we obtain $e_1'\left(P, \hat{\varphi}(Q)\right) = e_2'\left(\varphi(P), Q\right)$, and $e_1'\left(-P, \hat{\varphi}(Q)\right) = e_2'\left(\varphi(-P), Q\right)$, where $\overline{P} \in \mathcal{K}_1^{Sqr}$ together with P (one of related points on J_1), and similar for $\overline{Q} \in \mathcal{K}_2^{Sqr}, Q \in J_2$. Then it implies

$$\overline{e}_1'(\overline{P}, \hat{\varphi}(\overline{Q})) = \overline{e}_2'(\varphi(\overline{P}), \overline{Q}).$$

Therefore, verification succeeds if the outputs are correct.

Since $R \mapsto e_1'(R, P)$ is a surjective group homomorphism, and $x, x^{-1} \in \mu_\ell$ correspond the same value in S_ℓ, the probability of a random point $\overline{R} \in \mathcal{K}_2^{Sqr}$ satisfying $\overline{e}_1'(\overline{P}, \overline{R}) = \overline{e}_2'(\varphi(\overline{P}), \overline{Q})$ is no more than $2/\ell$. □

The property of sequentiality is satisfied, and we will explain it later.

5.2 Delay Encryption

A Delay Encryption consists of four algorithms:

1. $\mathsf{Setup}(\lambda, T) \to (ek, pk)$. Take a security parameter λ, a delay parameter T as inputs, and produce public parameters consisting of an extraction key ek and an encryption key pk. Setup must run in time $poly(\lambda, T)$ and the encryption key pk must have size $poly(\lambda)$, but the evaluation key ek is allowed to have size $poly(\lambda, T)$.
2. $\mathsf{Extract}(ek, id) \to idk$. Take the extraction key ek and a session identifier $id \in \{0,1\}^*$ as inputs, and output a session key idk. Extract is expected to run in time exactly T.
3. $\mathsf{Encaps}(pk, id) \to (c, k)$. Take the encryption key pk and a session identifier $id \in \{0,1\}^*$ as inputs, and output a ciphertext $c \in \mathbf{C}$ and a key $k \in \mathbf{K}$. Encaps must run in time $poly(\lambda)$.

4. $\mathsf{Decaps}(pk, id, idk, c) \to k$. Take the encryption key pk, a session identifier id, a session key idk, a ciphertext $c \in \mathbf{C}$ as inputs, and output a key $k \in \mathbf{K}$. Decaps must run in time $poly(\lambda)$.

A Delay Encryption scheme is correct if for any $(\mathsf{ek}, \mathsf{pk}) = \mathsf{Setup}(\lambda, \mathsf{T})$ and any $id \in \{0,1\}^*$,

$$idk = \mathsf{Extract}(ek, id) \wedge (c, k) = \mathsf{Encaps}(pk, id) \Rightarrow \mathsf{Decaps}(pk, id, idk, c) = k.$$

The security requirements of Delay Encryption are the same as VDF.

Our Delay Encryption Scheme. Similarly, we obtain a Delay Encryption scheme by modifying the IBE scheme in [2]: the master secret is replaced by a long chain of Richelot isogenies on squared Kummer surfaces, while the session key plays the role of identities so that producing the decryption key for a given identifier becomes a slow operation.

Before introducing the scheme, we need two secure hash functions. Let $H_1 : \{0,1\}^\lambda \to \mathcal{K}_2^{Sqr}[\ell]$ be used to hash id to points of order ℓ, where ℓ is a large prime, and $H_2 : \mathbb{F}_{q^k} \to \{0,1\}^\lambda$ be a key derivation. In the same time, $\overline{e}'_1$ and $\overline{e}'_2$ are squared symmetric ℓ-Weil pairings on $\mathcal{K}_1^{Sqr}$ and $\mathcal{K}_2^{Sqr}$. The computation of Richelot isogeny and squared pairing is what we use in our VDF scheme.

The Delay Encryption based on squared Kummer surfaces is described as follows:

$\mathsf{Setup}(\lambda, T)$.

1. Choose primes p, ℓ, according to the security parameter λ;
2. Select a genus two hyperelliptic curve $C_{\lambda,\mu,\upsilon}$ and convert it to a squared Kummer surface $\mathcal{K}_1^{Sqr}$ with the identity point $\mathcal{O}_1$;
3. Select a point P of order ℓ, and convert it to the point $\overline{P}$ on $\mathcal{K}_1^{Sqr}$;
4. Compute T Richelot isogenies $\varphi : \mathcal{K}_1^{Sqr} \to \mathcal{K}_2^{Sqr}$ and $\mathcal{O}_2 = \varphi(\mathcal{O}_1), \varphi(\overline{P})$;
5. Output $(ek, vk) = (\varphi, (\mathcal{O}_1, \mathcal{O}_2, \overline{P}, \varphi(\overline{P})))$.

$\mathsf{Extract}(\mathcal{K}_1^{Sqr}, \mathcal{K}_2^{Sqr}, \varphi, id)$.

1. Let $\overline{Q} = H_1(id)$;
2. Output $\hat{\varphi}(\overline{Q})$.

$\mathsf{Encaps}(\mathcal{K}_1^{Sqr}, \mathcal{K}_2^{Sqr}, \overline{P}, \varphi(\overline{P}), id)$.

1. Select a uniformly random $r \in \mathbb{Z}/\ell\mathbb{Z}$;
2. Let $\overline{Q} = H_1(id)$;
3. Compute $k = \overline{e}'_2(\varphi(\overline{P}), r\overline{Q})$;
4. Output $(r\overline{P}, H_2(k))$.

$\mathsf{Decaps}(\mathcal{K}_1^{Sqr}, \mathcal{K}_2^{Sqr}, \varphi(\overline{Q}), r\overline{P})$.

1. Let $k = \overline{e}'_1(r\overline{P}, \hat{\varphi}(\overline{Q}))$;
2. Output $H_2(k)$.

The properties of correctness and soundness are similar to our VDF scheme on squared Kummer surfaces, so we omit the proof. The property of sequentiality will be analyzed in the next subsection.

5.3 Sequentiality

The most important property of two schemes is sequentiality [15] in common, hence, we analyze it in this subsection. The sequentiality is defined as

Definition 3 (Sequentiality). *A scheme is sequential if no pair of randomized algorithms $\mathcal{A}_0$, which runs in total time poly(T, λ), and $\mathcal{A}_1$, which runs in parallel time less than T, can win with nonnegligible probability the following sequentiality game*

1. $(ek, vk) \leftarrow \mathsf{Setup}(\lambda, T)$, *where the random input tape to* Setup *is filled with uniformly distributed bits,*
2. $A \leftarrow \mathcal{A}_0(\lambda, ek, vk, T)$,
3. $\overline{Q} \leftarrow \mathcal{K}_2^{Sqr}$, *uniformly sampled,*
4. $\overline{Q}_0' \leftarrow \mathcal{A}_1(A, vk, \overline{Q})$,

where wining is defined as outputting

$$\overline{e}_1'(\overline{P}, \overline{Q}_0') = \overline{e}_2'(\varphi(\overline{P}), \overline{Q}). \tag{16}$$

An evident way to break sequentiality is to search for a shortcut of known isogeny. In our work, the isogeny shortcut problem [15] is defined to find a point $\overline{Q}_0' \in \mathcal{K}_1^{Sqr}$ satisfying equation (16) in parallel time less than T. Our schemes can resist evident attacks on the isogeny shortcut problem.

Pairing Inversion. For a random point $\overline{Q}' \in \mathcal{K}_1^{Sqr}$ with associated point Q', we can acquire the value of $e_1'(P, Q')$ and $e_2'(\varphi(P), Q)$ (up to an inverse). Thus, the pairing inverse problem, i.e., solving the equation $\overline{e}_1'(\overline{P}, \cdot) = \overline{e}_2'(\varphi(\overline{P}), \overline{Q})$, turns to find $r \in \mathbb{Z}/\ell\mathbb{Z}$ such that

$$e_2'(\varphi(P), Q) = e_1'(P, \hat{\varphi}(Q)) = e_1'(P, Q')^r,$$

which is equivalent to the discrete logarithm problem in $\mathbb{F}_{q^k}$.

Given a curve with embedding degree 2 or 1, the best algorithm to solve DLP is the Number Field Sieve (NFS) for $\mathbb{F}_{p^2}$ with (heuristic) complexity $L_p(1/3)$. It is tougher to solve DLP under a higher embedding degree, but the implementation efficiency decreases as well.

Finally, after carefully choosing the parameters λ and q, our schemes are secure under some necessary hypotheses.

Computing Shortcuts. To break VDF and Delay Encryption based on elliptic curve isogenies, the original method is to find a simple isogeny path that can be used to compute the same codomain. Due to researches on elliptic curves isogeny graph, there are numerous tools, e.g., the KLPT algorithm [28] constructs a smooth ℓ^n isogeny, so this scheme might be insecure.

However, concerning squared Kummer surfaces, as special abelian varieties, there are no practical methods to compute (ℓ, ℓ)-isogeny. Only when $\ell = 3, 5$,

there are several works [4, 21] about the explicit formula of (ℓ, ℓ)-isogeny, but the formula becomes much more complicated, including the method introduced in Section 3.4. At the same time, the structures of the (ℓ, ℓ)-isogeny graphs are much complicated than elliptic curves, and we do not know the properties for $\ell = 2$. Some properties are discussed in [20], and only the local neighborhoods in the $(2, 2)$-isogeny graph have been exploited in [19].

As a result, finding a shortcut of known isogeny is a hard problem now.

Attacks on the Computation. In our schemes, computing the chain of Richelot isogenies costs most time, where the security relies. The question becomes if one can find a fast algorithm to speed up computing a chain of Richelot isogenies.

The fundamental idea is to find a better parallel algorithm to compute the chain of isogenies. However, every Richelot isogeny ϕ_i must use the output of the last isogeny ϕ_{i-1}. Hence, an adversary can not accelerate the computation even using $poly(\lambda)$ processors.

5.4 Computational Efficiency

In this subsection, the computational efficiency of our schemes is investigated. Two schemes are similar, so we only concern about VDF on squared Kummer surfaces. In a nutshell, the Richelot isogenies and pairings are the fundamental building blocks of our schemes.

The Richelot isogeny with kernel group $\{\Upsilon, \tilde{\Upsilon}\}$ on squared Kummer surfaces consists of three fundamental operations: $\mathcal{S}$, $\mathcal{H}$, $\mathcal{C}$. The map $\mathcal{S}$ only needs four squares, and the Hadamard transform $\mathcal{H}$ only needs eight additions [3, Algorithm 4] over finite fields. But the most complicated operation is the map $\mathcal{C}$ with four constants in our schemes. Four squared roots of coordinates $(\theta_1(z)^2, \theta_2(z)^2, \theta_3(z)^2, \theta_4(z)^2)$ on dual squared Kummer surfaces are required for the computation, and the last square root can be omitted by multiplying $\theta_4(z)^2$ to the coordinates. Sometimes, we need to extend the field to acquire square roots, and all choices are feasible for computation. As a result, one Richelot isogeny only needs 3 square roots, 4 square, and 10 multiplications.

The pairings in our schemes are the squared symmetric Weil pairings. For two points $\overline{P}, \overline{Q} \in \mathcal{K}^{Sqr}$ of order ℓ, the computation requires the coordinates of $\overline{\ell P}$, $\overline{\ell Q}$, $\overline{Q + \ell P}$ and $\overline{P + \ell Q}$. With PseudoAdd Algorithm presented in [29], the scheme asks $O(\log \ell)$ steps to compute a verification, satisfying requirements in VDF.

For a 128-bit secure VDF, ℓ should be a prime of 256 bits. To this goal, we choose a hyperelliptic curve, whose Jacobian contains at least two ℓ-torsion points that are not in one maximal isotropic ℓ-subgroup.

6 Conclusion and Perspective

We study Richelot isogenies between squared Kummer surfaces and the dual isogenies. Moreover, the image of the point can be obtained with the same

operations, which is critical for numerous applications. At the same time, we introduce the first self-contained symmetric pairings which can be evaluated with arithmetic on the same surface. At last, based on Richelot isogenies and squared symmetric pairings of genus-2 curves, the first VDF and Delay Encryption are instituted as two applications.

Unlike elliptic curves, there are few works about the isogenies between hyperelliptic curves, including Kummer surfaces. Moreover, the studies of hyperelliptic curves in mathematics will have profound influences on our algorithms. Meanwhile, a more efficient way to compute (ℓ, ℓ)-isogenies is also an open problem to be investigated, especially for large ℓ. Even for Richelot isogenies, our work does not include all situations, and it will be our next research object. Thus, there are works needed to be done in the future.

For two applications, we only concern about the isogeny shortcut problem, and they are not quantum-resistant since pairings are not secure under quantum computers. As a result, we look forward to researches on actual quantum-resistant VDFs and Delay Encryptions.

References

1. Boneh, D., Bonneau, J., Bünz, B., Fisch, B.: Verifiable delay functions. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 757–788. Springer (2018)
2. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
3. Bos, J.W., Costello, C., Hisil, H., Lauter, K.E.: Fast cryptography in genus 2. J. Cryptol. **29**(1), 28–60 (2016)
4. Bruin, N., Flynn, E.V., Testa, D.: Descent via $(3,3)$-isogeny on Jacobians of genus 2 curves. Acta Arith. **165**(3), 201–223 (2014)
5. Burdges, J., De Feo, L.: Delay encryption. In: Canteaut, A., Standaert, F. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 302–326. Springer (2021)
6. Castryck, W., Decru, T.: Multiradical isogenies. IACR Cryptol. ePrint Arch. **2021**, 1133, https://eprint.iacr.org/2021/1133
7. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S.D. (eds.) ASIACRYPT 2018. LNCS, vol. 11274, pp. 395–427. Springer (2018)
8. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. J. Cryptol. **22**(1), 93–113 (2009)
9. Chudnovsky, D.V., Chudnovsky, G.V.: Sequences of numbers generated by addition in formal groups and new primality and factorization tests. Adv. in Appl. Math. **7**(4), 385–434 (1986)
10. Cosset, R.: Factorization with genus 2 curves. Math. Comput. **79**(270), 1191–1208 (2010)
11. Cosset, R., Robert, D.: Computing (ℓ,ℓ)-isogenies in polynomial time on jacobians of genus 2 curves. Math. Comput. **84**(294), 1953–1975 (2015)
12. Costello, C.: Computing supersingular isogenies on Kummer surfaces. In: Peyrin, T., Galbraith, S.D. (eds.) ASIACRYPT 2018. LNCS, vol. 11274, pp. 428–456. Springer (2018)

13. Costello, C., Smith, B.: The supersingular isogeny problem in genus 2 and beyond. In: Ding, J., Tillich, J. (eds.) PQCrypto 2020. LNCS, vol. 12100, pp. 151–168. Springer (2020)
14. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Math. Cryptol. **8**(3), 209–247 (2014)
15. De Feo, L., Masson, S., Petit, C., Sanso, A.: Verifiable delay functions from supersingular isogenies and pairings. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11921, pp. 248–277. Springer (2019)
16. Döttling, N., Garg, S., Malavolta, G., Vasudevan, P.N.: Tight verifiable delay functions. In: Galdi, C., Kolesnikov, V. (eds.) SCN 2020. LNCS, vol. 12238, pp. 65–84. Springer (2020)
17. Duquesne, S., Frey, G.: Implementation of pairings. In: Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F. (eds.) Handbook of Elliptic and Hyperelliptic Curve Cryptography, pp. 389–404. Chapman and Hall/CRC (2005)
18. Ephraim, N., Freitag, C., Komargodski, I., Pass, R.: Continuous verifiable delay functions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12107, pp. 125–154. Springer (2020)
19. Florit, E., Smith, B.: An atlas of the Richelot isogeny graph. IACR Cryptol. ePrint Arch. **2021**, 13, https://eprint.iacr.org/2021/013
20. Florit, E., Smith, B.: Automorphisms and isogeny graphs of abelian varieties, with applications to the superspecial richelot isogeny graph. CoRR **abs/2101.00919** (2021)
21. Flynn, E.V.: Descent via $(5, 5)$-isogeny on Jacobians of genus 2 curves. J. Number Theory **153**, 270–282 (2015)
22. Flynn, E.V., Ti, Y.B.: Genus two isogeny cryptography. In: Ding, J., Steinwandt, R. (eds.) PQCrypto 2019. LNCS, vol. 11505, pp. 286–306. Springer (2019)
23. Galbraith, S.D., Hess, F., Vercauteren, F.: Hyperelliptic pairings. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 108–131. Springer (2007)
24. Galbraith, S.D., Lin, X.: Computing pairings using x -coordinates only. Des. Codes Cryptogr. **50**(3), 305–324 (2009)
25. Gaudry, P.: Fast genus 2 arithmetic based on theta functions. J. Math. Cryptol. **1**(3), 243–265 (2007)
26. Igusa, J.i.: Theta functions. Die Grundlehren der mathematischen Wissenschaften, Band 194, Springer-Verlag, New York-Heidelberg (1972)
27. Jao, D., et al.: SIKE supersingular isogeny key encapsulation (2017), https://sike.org
28. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion ℓ-isogeny path problem. LMS J. Comput. Math. **17**(suppl. A), 418–432 (2014)
29. Lubicz, D., Robert, D.: Efficient pairing computation with theta functions. In: Hanrot, G., Morain, F., Thomé, E. (eds.) Algorithmic Number Theory, 2010. LNCS, vol. 6197, pp. 251–269. Springer (2010)
30. Lubicz, D., Robert, D.: Computing isogenies between abelian varieties. Compos. Math. **148**(5), 1483–1515 (2012)
31. Lubicz, D., Robert, D.: Arithmetic on abelian and Kummer varieties. Finite Fields Appl. **39**, 130–158 (2016)
32. Miller, V.S.: The Weil pairing, and its efficient calculation. J. Cryptol. **17**(4), 235–261 (2004)

33. Mumford, D.: Abelian varieties, Tata Institute of Fundamental Research Studies in Mathematics, vol. 5. Published for the Tata Institute of Fundamental Research, Bombay; Oxford University Press, London (1970)
34. Mumford, D.: Tata lectures on theta. I. Modern Birkhäuser Classics, Birkhäuser Boston, Inc., Boston, MA (2007), with the collaboration of C. Musili, M. Nori, E. Previato and M. Stillman, Reprint of the 1983 edition
35. Mumford, D.: Tata lectures on theta. II. Modern Birkhäuser Classics, Birkhäuser Boston, Inc., Boston, MA (2007), jacobian theta functions and differential equations, With the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura, Reprint of the 1984 original
36. Pietrzak, K.: Simple verifiable delay functions. In: Blum, A. (ed.) ITCS 2019. LIPIcs, vol. 124, pp. 60:1–60:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
37. Renes, J., Smith, B.: qDSA: Small and secure digital signatures with curve-based Diffie-Hellman key pairs. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10625, pp. 273–302. Springer (2017)
38. Smith, B.: Explicit endomorphisms and correspondences. Bulletin of the Australian Mathematical Society (2006)
39. Vélu, J.: Isogénies entre courbes elliptiques. C. R. Acad. Sci. Paris Sér. A-B **273**, A238–A241 (1971)
40. van Wamelen, P.: Equations for the Jacobian of a hyperelliptic curve. Trans. Amer. Math. Soc. **350**(8), 3083–3106 (1998)
41. Wesolowski, B.: Efficient verifiable delay functions. J. Cryptol. **33**(4), 2113–2147 (2020)