A collision attack on the LTZ hash function based on a conjecture on supersingular non-superspecial isogeny graphs of dimension 2

Ryo Ohashi[1] and Hiroshi Onuki[1]

The University of Tokyo, Japan
{ryo-ohashi,hiroshi-onuki}@g.ecc.u-tokyo.ac.jp

Abstract. In 2023, LeGrow, Ti, and Zobernig proposed a cryptographic hash function (we refer to it as the LTZ hash function in this paper) based on a certain $(2,2)$-isogeny graph between supersingular non-superspecial abelian surfaces over $\mathbb{F}_{p^4}$. The authors estimated that the time and space complexities required to find a collision in the LTZ hash function are both given by $\widetilde{O}(p^3)$. In this paper, we first propose a mathematical conjecture on the number of vertices defined over the smaller field $\mathbb{F}_{p^2}$ in the isogeny graphs used in the LTZ hash function. Based on this conjecture, we then construct a collision-finding algorithm for the LTZ hash function, which preserves the time complexity $\widetilde{O}(p^3)$, while reducing the required memory to $O(\log(p)^2)$. We implemented this algorithm in Rust, and successfully found a collision for parameters claimed to provide 35–37 bits of security, within 26.2MB of memory usage on average in 20.2 hours.

Keywords: Hash function, Isogeny-based cryptography, supersingular non-superspecial abelian surfaces

1 Introduction

Isogeny-based cryptography is one of the promising candidates for post-quantum cryptography, which is secure against quantum computers. The security of isogeny-based cryptography is based on the difficulty of finding isogenies between elliptic curves over a finite field. Currently, most of the isogeny-based cryptographic protocols use supersingular elliptic curves because the isogeny graph of supersingular elliptic curves has an attractive structure for cryptographic applications.

The first isogeny-based cryptographic protocol that uses supersingular elliptic curves is the cryptographic hash function proposed by Charles, Lauter, and Goren [6], referred to as the *CGL hash function*. Roughly speaking, the computation of the CGL hash function is based on a random walk in the isogeny graph of supersingular elliptic curves. Since an elliptic curve is a one-dimensional abelian variety, it is natural to generalize the CGL hash function to hash functions using higher-dimensional abelian varieties. The first such hash function was proposed by Takashima [24], which uses superspecial abelian surfaces (i.e., two-dimensional abelian varieties). Flynn and Ti [11] pointed out a vulnerability

in Takashima's hash function, and Castryck, Decru, and Smith [5] proposed a fix for it. The resulting hash function is referred to as the *CDS hash function*. LeGrow, Ti, and Zobernig [20] proposed another hash function using abelian surfaces, but they used supersingular abelian surfaces which are not superspecial. Recently, Kunzweiler et al. [19] proposed an efficient computing method for the CDS hash function via theta functions, along with a hash function based on superspecial abelian varieties of dimension 3.

The collision resistance of the above hash functions relies on the difficulty of finding an endomorphism with a certain property of an abelian variety, including an elliptic curve. The problem of finding such an endomorphism is equivalent to that of finding a cycle in the isogeny graph of abelian varieties. There are several approaches to find a cycle in the isogeny graphs, and the characteristic p of the finite field is determined by the computational complexity of these approaches:

- As a generic approach for finding a cycle in the graph, the Pollard-rho style algorithm is known. In this approach, we start from the initial vertex of the target hash function, and repeatedly perform a random walk in the isogeny graph. If we walk through approximately $\sqrt{N}$ vertices, where N denotes the number of vertices in the isogeny graph, then we can expect to find the same vertex twice by the birthday paradox. Since we need to store the vertices we have visited, the time and space complexity of this approach is $O(N^{1/2})$ up to a polynomial factor in $\log(p)$.
- In the case of the CGL hash function, the size of the isogeny graph of supersingular elliptic curves is approximately $p/12$. For this graph, a more efficient approach to finding a cycle was proposed by Eisenträger, Hallgren, Leonardi, Morrison, and Park [9]. This approach requires $\widetilde{O}(p^{1/2})$ time complexity as same as the Pollard-rho style algorithm, but its space complexity is bounded by some polynomial in $\log(p)$.
- For the high-dimensional case, it is known that the number of vertices in the isogeny graph of superspecial abelian varieties of dimension g is $\Theta(p^{g(g+1)/2})$. In the case where $g \geq 2$, we can reduce the problem of finding a cycle in the isogeny graph to that of finding a cycle in lower-dimensional isogeny graphs by using a method by Costello and Smith [8]. The time complexity of their approach is $\widetilde{O}(p^{g-1})$, and moreover the space complexity is bounded by some polynomial in $\log(p)$. On the other hand, we cannot apply this approach to the isogeny graph of supersingular non-superspecial abelian surfaces, which is used in the LTZ hash function.
- In the quantum setting, the claw-finding algorithm by Brassard, Høyer, and Tapp [1] and Tani [25] can be applied. This algorithm requires $O(N^{1/3})$ time and space complexity up to a polynomial factor in $\log(p)$, where N denotes the number of vertices in the isogeny graph. However, Jaques and Schanck showed in [18] that the quantum claw-finding algorithm does not outperform the classical Pollard-rho style algorithm due to the required data structure in the isogeny setting. The attacks by Eisenträger et al. and Costtelo-Smith can be combined with Grover's algorithm [13], reducing the time complexity to $\widetilde{O}(p^{1/4})$ and $\widetilde{O}(p^{(g-1)/2})$, respectively.

Table 1. Time and space complexities of the attacks on the hash functions using the isogeny graphs of abelian varieties. The characteristic p of the finite field is chosen such that the classical (resp. quantum) security level is λ bits (resp. $\lambda/2$ bits).

Hash function	Dim.	Complexity of classical attack		Complexity of quantum attack		Base field	Size of p
		time	space	time	space		
CGL	1	$\widetilde{O}(p^{1/2})$	poly.	$\widetilde{O}(p^{1/4})$	poly.	$\mathbb{F}_{p^2}$	$2^{2\lambda}$
CDS	2	$\widetilde{O}(p)$	poly.	$\widetilde{O}(p^{1/2})$	poly.	$\mathbb{F}_{p^2}$	2^{λ}
Kunzweiler et al.	3	$\widetilde{O}(p^2)$	poly.	$\widetilde{O}(p)$	poly.	$\mathbb{F}_{p^2}$	$2^{\lambda/2}$
LTZ (original)	2	$\widetilde{O}(p^3)$	$\widetilde{O}(p^3)$	$\widetilde{O}(p^3)$	$\widetilde{O}(p^3)$	$\mathbb{F}_{p^4}$	$2^{\lambda/3}$
LTZ (our)	2	$\widetilde{O}(p^3)$	**poly.**	$\boldsymbol{\widetilde{O}(p^{3/2})}$	**poly.**	$\mathbb{F}_{p^4}$	$2^{\lambda/3}$

Here, Table 1 summarizes the time and space complexities of the attacks on the hash functions using the isogeny graphs of abelian varieties. The analysis by Kunzweiler et al. [19] shows that the hash function based on superspecial abelian varieties of dimension 3 achieves greater computational efficiency than the CGL and CDS hash functions. Although they do not provide a direct comparison with the LTZ hash function, the parameter scaling suggests a practical advantage of the LTZ hash function from an implementation standpoint. To target a security level λ, LTZ can use a prime $p \approx 2^{\lambda/3}$, whereas their hash function requires $p \approx 2^{\lambda/2}$. Consequently, even at higher security levels the LTZ prime often remains within the range of single-machine-word integers (e.g., 64-bit), which simplifies implementations. Therefore, it is important to analyze the collision resistance of the LTZ hash function. In particular, the following question arises:

> *Is there a collision attack on the LTZ hash function beyond the generic Pollard-rho style algorithm?*

In this paper, we give a positive answer to the above question.

Contribution. In this paper, we propose a memory-efficient collision attack on the LTZ hash function. Specifically, we focus on the vertices in the isogeny graph underlying the LTZ hash function that are defined over the smaller field $\mathbb{F}_{p^2}$. By using such vertices, we can construct a collision attack based on an idea similar to the one by Eisenträger et al. [9]. To prove the effectiveness of our attack, we show in Section 4 several mathematical properties of these $\mathbb{F}_{p^2}$-rational vertices, and also give a conjecture that their number is sufficiently large, which is verified by our experiments for small primes.

The time complexity of our attack is $\widetilde{O}(p^3)$, which is the same as that of the Pollard-rho style algorithm, while the space complexity is reduced to $O(\log(p)^2)$, in the classical setting. In the quantum setting, our attack can be combined with Grover's algorithm, reducing its time complexity to $\widetilde{O}(p^{3/2})$, which outperforms the claw-finding algorithm. A comparison of the time and space complexities of our attack and previous attacks is shown in Table 1.

Note that, although our attack does not improve the time complexity in the classical setting, it may still affect parameter choices for the LTZ hash function, since the Pollard-rho style algorithm requires exponential space complexity. For example, the parameter choice for SIKE [17], an isogeny-based key encapsulation mechanism and a Round 4 candidate in the NIST post-quantum standardization process, was based on the assumption that the available memory allows storing 2^{80} units. This results in smaller parameter sizes than those required to achieve a time complexity of 2^{λ} under the assumption of unlimited memory, where λ is the desired security level. Without our attack, the characteristic p of the finite field in the LTZ hash function could be chosen smaller than $2^{\lambda/3}$, under similar assumptions. In contrast, our attack shows that p should be at least $2^{\lambda/3}$.

Organization. In Section 2, we review several mathematical preliminaries, and in Section 3, we present the specific construction of the LTZ hash function. Next, we discuss properties and a conjecture on the $\mathbb{F}_{p^2}$-rational vertices in the isogeny graph underlying the LTZ hash function in Section 4. Based on this conjecture, Section 5 introduces our collision attack on the LTZ hash function, and provides an estimate of its computational complexity. Finally, in Section 6, we show experimental results of our attack, which we implemented in Rust.

2 Preliminaries

In this section, we summarize some mathematical background which will be used in later sections. Throughout, all curves and abelian varieties are assumed to be defined over a field k of characteristic $p > 5$ (a curve always refers to a *smooth* projective variety of dimension one). In addition, all isomorphisms between them are taken to be over the algebraic closure $\bar{k}$ of k.

2.1 Principally polarized abelian surfaces and their isogenies

A *principally polarized abelian surface* is defined as a pair $(A, \mathcal{L})$, where A is an abelian surface (i.e., a smooth projective variety of dimension 2 which is also an algebraic group) and $\mathcal{L} : A \to \hat{A}$ is a principal polarization (i.e., an isomorphism to the dual abelian surface, induced by an ample line bundle).

As is well known, every principally polarized abelian surface is isomorphic to either of the following two types:

- the product $E_1 \times E_2$ of two elliptic curves E_1, E_2 equipped with the product principal polarization, or
- the Jacobian variety $\mathrm{Jac}(H)$ of a genus-2 curve H equipped with its canonical principal polarization induced by the theta divisor.

In particular, we refer to the former case as a *decomposable* abelian surface. If a principal polarization on an abelian surface is not specified, then it is implicitly assumed that the canonical one is being considered.

Let $\mathcal{A}_2$ (resp. $\mathcal{M}_2$) denote the moduli space of principally polarized abelian surfaces (resp. genus-2 curves), then the Torelli theorem asserts that the map

$$\begin{aligned} \mathcal{M}_2 &\longrightarrow \mathcal{A}_2, \\ H &\longmapsto \mathrm{Jac}(H) \end{aligned}$$

is well-defined and injective. In other words, two genus-2 curves are isomorphic if and only if their Jacobian varieties are isomorphic. Moreover, it is known that the moduli space $\mathcal{M}_2$ of genus-2 curves has dimension 3, and classical invariants such as the *Igusa invariants* provide a system of generators for its function field. In particular, we review the definition of the absolute invariants of genus-2 curves as given in [2, pp. 73–74].

Definition 1. *For a genus-2 curve H, we define*

$$(j_1, j_2, j_3) \coloneqq \begin{cases} (J_2^5/J_{10}, J_2^3 J_4/J_{10}, J_2^2 J_6/J_{10}) & \text{if } J_2 \neq 0, \\ (0, J_4^5/J_{10}^2, J_4 J_6/J_{10}) & \text{if } J_2 = 0, J_4 \neq 0, \\ (0, 0, J_6^5/J_{10}^3) & \text{if } J_2 = J_4 = 0, \end{cases}$$

where J_2, J_4, J_6, J_{10} are the Igusa invariants of H. Remark that J_{10} is non-zero, and hence this definition is well-defined. Then, the triple (j_1, j_2, j_3) is called the ***absolute invariants*** *of H.*

An important feature is that, the set of k-rational points on $\mathcal{M}_2$ is in bijection with the set of triples $(j_1, j_2, j_3) \in k^3$. Then, the following facts hold:

- Two genus-2 curves are $\bar{k}$-isomorphic if and only if their absolute invariants are the same.
- For any genus-2 curve defined over k, its absolute invariants lie in k^3.

In addition, when k is a finite field, for any $(j_1, j_2, j_3) \in k^3$ there exists a genus-2 curve defined over k with these absolute invariants (cf. [28, Remark 1]).

Remark 2. For a genus-2 curve over k, the computation of its absolute invariants requires only a constant number of arithmetic operations in k, since J_2, J_4, J_6, J_{10} are given by polynomials over k of bounded degree (cf. [16, Sections 3–4]).

Fix an integer $N > 1$ coprime to p. For an abelian surface A, let $A[N]$ denote the N-torsion subgroup of A, then we obtain an isomorphism $A[N] \cong (\mathbb{Z}/N\mathbb{Z})^4$. Also, there exists a non-degenerate alternating bilinear form $A[N] \times \hat{A}[N] \to \mu_N$, where μ_N is the set of N-th roots of unity in $\bar{k}$. In particular, when A has a principal polarization, it induces a map

$$e_N : A[N] \times A[N] \longrightarrow \mu_N,$$

called the *N-Weil pairing* of A. We now introduce the notions of isotropic subgroups and symplectic bases of $A[N]$ with respect to the N-Weil pairing:

Definition 3. *Let A be a principally polarized abelian surface. A subgroup G of its N-torsion subgroup $A[N]$ is called* ***isotropic*** *if $e_N(P, Q) = 1$ for all $P, Q \in G$. Also, a basis $(P_1, P_2; Q_1, Q_2)$ of $A[N]$ is called* ***symplectic*** *if*

- $e_N(P_1, P_2) = e_N(Q_1, Q_2) = 1$,
- $e_N(P_1, Q_1) = e_N(P_2, Q_2) = \zeta_N$, *and*
- $e_N(P_1, Q_2) = e_N(P_2, Q_1) = 1$,

where $\zeta_N \in \mu_N$ *is a primitive* N*-th root of unity.*

In the context of isogeny-based cryptography, we are often interested in isogenies of principally polarized abelian surfaces of the following form:

Definition 4. *Let* A *and* A' *be principally polarized abelian surfaces. An isogeny (i.e., finite surjective homomorphism) from* A *to* A' *is called an* $\boldsymbol{(N,N)}$***-isogeny*** *when its kernel is a maximal isotropic subgroup with respect to inclusion. If such an isogeny exists, we say that* A *and* A' *are* $\boldsymbol{(N,N)}$***-isogenous****.*

Two (N, N)-isogenies from A to A' with the same kernel are equivalent up to an automorphism of A', and hence they are identified. For any principally polarized abelian surface A, the number of distinct (N, N)-isogenies with the domain A is known (cf. [3, Lemma 2.1]) to be

$$S(N) \coloneqq N^3 \prod_{\text{primes } \ell \mid N} \left(1 + \frac{1}{\ell}\right)\left(1 + \frac{1}{\ell^2}\right). \tag{2.1}$$

Moreover, the (N, N)-isogenies whose domain is the codomain of a given (N, N)-isogeny can be classified into 3 types according to the structure of the kernel of their composition:

Definition 5. *For two* (N, N)*-isogenies* $\phi : A \to A'$ *and* $\phi' : A' \to A''$ *between principally polarized abelian surfaces,*

- *If* $\ker \phi' = \phi(A[N])$*, then we say that* ϕ' *is the* ***dual extension*** *of* ϕ*.*
- *If* $\ker \phi' \cap \phi(A[N]) = \{0\}$*, then we say that* ϕ' *is a* ***good extension*** *of* ϕ*.*
- *Otherwise, we say that* ϕ' *is a* ***bad extension*** *of* ϕ*.*

By [3, Lemma 2.1] again, for any (N, N)-isogeny, there exist exactly N^3 distinct good extensions. For an integer $n > 0$, any (N^n, N^n)-isogeny $\psi : A \to A'$ can be written as a composition of (N, N)-isogenies

$$A = A_0 \xrightarrow{\phi_1} A_1 \xrightarrow{\phi_2} A_2 \longrightarrow \cdots \longrightarrow A_{n-1} \xrightarrow{\phi_n} A_n = A',$$

where ϕ_{i+1} is a good extension of ϕ_i for all $i \in \{1, \ldots, n-1\}$. For the notational convenience, we introduce the following terminology.

Definition 6. *A sequence* $(\phi_1, \ldots, \phi_n)$ *of* (N, N)*-isogenies of principally polarized abelian surfaces is called a* ***good*** $\boldsymbol{(N,N)}$***-sequence*** *if* ϕ_{i+1} *is a good extension of* ϕ_i *for all* $i \in \{1, \ldots, n-1\}$*.*

2.2 Supersingular abelian surfaces

An elliptic curve E is said to be *supersingular* if its p-torsion subgroup is trivial; otherwise E is called *ordinary*. As a two-dimensional analogue of a supersingular elliptic curve, we define a supersingular abelian surface as follows:

Definition 7. *Let A be an abelian surface.*

- *We say that A is **supersingular** if A is isogenous (over $\bar{k}$) to the product of two supersingular elliptic curves, without polarizations.*
- *In particular, we say that A is **superspecial** if A is isomorphic (over $\bar{k}$) to the product of two supersingular elliptic curves, without polarizations.*

It follows immediately that any superspecial abelian surface is supersingular.

By the Deligne-Ogus-Shioda theorem, any two products of supersingular elliptic curves are isomorphic over $\bar{k}$ to each other as unpolarized abelian surfaces, which shows that all superspecial abelian surfaces are isomorphic without polarizations. The number of principal polarizations on a fixed superspecial abelian surface A, up to the action of automorphisms of A, is approximately $p^3/2880$.

The supersingularity and superspeciality of the Jacobian variety of a genus-2 curve H can be completely determined from the Cartier-Manin matrix of H (we refer to [15, Lemma 1.1] for the criterion). For example, by explicitly computing the Hasse-Witt matrix of $H : y^2 = x^5 - 1$, one can show the following fact:

Proposition 8 ([15, Proposition 1.13]). *For a genus-2 curve $H : y^2 = x^5 - 1$, we have the following:*

1. *If $p \equiv 1 \pmod 5$, then $\mathrm{Jac}(H)$ is not supersingular.*
2. *If $p \equiv 2, 3 \pmod 5$, then $\mathrm{Jac}(H)$ is supersingular and not superspecial.*
3. *If $p \equiv 4 \pmod 5$, then $\mathrm{Jac}(H)$ is superspecial.*

We note that $H : y^2 = x^5 - 1$ is classified as Class (6) in [15], and it is the unique genus-2 curve (up to isomorphism) whose automorphism group has order 10.

Remark 9. The product $E_1 \times E_2$ of two elliptic curves E_1, E_2 is supersingular if and only if both E_1 and E_2 are supersingular. In that case, the abelian surface is automatically superspecial; therefore, a decomposable abelian surface cannot be supersingular without being superspecial.

Next, we consider an abelian surface A defined over $k = \mathbb{F}_q$ with $q = p^r$. The q-th power Frobenius endomorphism on A satisfies the equation

$$\Phi(X) \coloneqq X^4 + a_1 X^3 + a_2 X^2 + q a_1 X + q^2 = 0 \tag{2.2}$$

for some integers $a_1, a_2 \in \mathbb{Z}$, and $\Phi(X) \in \mathbb{Z}[X]$ is called the *characteristic polynomial* of A over $\mathbb{F}_q$. This polynomial can be factored as

$$\Phi(X) = \prod_{i=1}^{4} (X - \alpha_i),$$

where each α_i is an algebraic integer such that $|\alpha_i| = \sqrt{q}$. Moreover, the following fact is also well-known (see, for example [27, Section 1]).

Proposition 10. *With the above notation, the characteristic polynomial $\Phi_n(X)$ of A over $\mathbb{F}_{q^n}$ with $n \geq 1$ is written as*

$$\Phi_n(X) = \prod_{i=1}^{4}(X - \alpha_i^n).$$

In addition, the number of $\mathbb{F}_{q^n}$-rational points of A equals $\Phi_n(1)$.

As the following theorem shows, the supersingularity of an abelian surface is closely related to its characteristic polynomial:

Theorem 11 ([12, Theorem 6]). *If the characteristic polynomial of an abelian surface A over $\mathbb{F}_q$ with $q = p^r$ is given in the form (2.2), then A is supersingular if and only if $p^{\lceil r/2 \rceil} \mid a_1$ and $p^r \mid a_2$.*

In particular, one can determine the characteristic polynomials of the Jacobian variety of $H : y^2 = x^5 - 1$ over $\mathbb{F}_p$ in the supersingular non-superspecial case:

Proposition 12 ([7, Example 5.1]). *We assume that $p \equiv 2, 3 \pmod 5$. Then, the characteristic polynomial of the Jacobian variety of $H : y^2 = x^5 - 1$ over $\mathbb{F}_p$ is given as $\Phi(X) = X^4 + p^2$.*

In the following, we fix a supersingular elliptic curve E defined over $\bar{k}$. Then, by [21, Corollary 7], for any supersingular abelian surface A over $\bar{k}$, there exists a short exact sequence

$$0 \longrightarrow \alpha_p \xrightarrow{\iota_t} E \times E \twoheadrightarrow A \longrightarrow 0,$$

where α_p denotes the kernel of the p-th power Frobenius endomorphism on the additive group $\mathbb{G}_a$. The possible embeddings $\iota_t : \alpha_p \to E \times E$ are parameterized by $t \in \mathbb{P}^1(\bar{k})$, which is called the *embedding parameter* in [20, Section 2.3]. Also, it follows from [21] that the abelian surface $A = (E \times E)/\iota_t(\alpha_p)$ is superspecial if and only if $t \in \mathbb{P}^1(\mathbb{F}_{p^2})$. Our interest lies in the supersingular non-superspecial case, in particular, when $t \in \mathbb{F}_{p^4} \setminus \mathbb{F}_{p^2}$. For such supersingular abelian surfaces, the following results by Ibukiyama are known:

Proposition 13 ([14, Proposition 2.3]). *Every supersingular abelian surface with embedding parameter in $\mathbb{F}_{p^4} \setminus \mathbb{F}_{p^2}$ is isomorphic (over $\bar{k}$) to any other such surface, as unpolarized abelian surfaces.*

Theorem 14 ([14, Theorems 1.1 and 1.2]). *Let A be a supersingular abelian surface with embedding parameter in $\mathbb{F}_{p^4} \setminus \mathbb{F}_{p^2}$. The number of principally polarizations of A up to the action of automorphisms on A equals*

$$h = \begin{cases} \frac{p^2(p^2-1)^2}{2880} & \text{if } p \equiv 1, 4 \pmod 5, \\ 1 + \frac{(p-3)(p+3)(p^2-3p+8)(p^2+3p+8)}{2880} & \text{if } p \equiv 2, 3 \pmod 5. \end{cases}$$

If $p \equiv 2, 3 \pmod 5$, then there exists a unique principal polarization $\mathcal{L}$ on A, up to isomorphism, such that $\mathrm{Aut}(A, \mathcal{L}) \cong \mathbb{Z}/10\mathbb{Z}$. For all other principal polarizations $\mathcal{L}$ on A, we have $\mathrm{Aut}(A, \mathcal{L}) = \{\pm 1\}$.

It follows from Proposition 13 that the $\bar{k}$-isomorphism class of supersingular abelian surfaces with embedding parameter in $\mathbb{F}_{p^4} \setminus \mathbb{F}_{p^2}$ is unique, and we denote it by A_0. For any principal polarization $\mathcal{L}$ on A_0, the principally polarized abelian surface $(A_0, \mathcal{L})$ is isomorphic to the Jacobian variety of a genus-2 curve equipped with its canonical principal polarization by Remark 9, and we write $A_{\mathcal{L}}$ for this. Then, we define the set of $\bar{k}$-isomorphism classes of $A_{\mathcal{L}}$ by

$$\mathcal{V}_p \coloneqq \{A_{\mathcal{L}} \mid \mathcal{L} \text{ is a principal polarization on } A_0\}/\cong,$$

which depends only on p. It is clear that the cardinality of $\mathcal{V}_p$ equals h, given in Theorem 14. Additionally, it follows from [20, Lemma 3] that every supersingular principally polarized abelian surface $A \in \mathcal{V}_p$ is defined over $\mathbb{F}_{p^4}$. We finally define the isogeny graph of these principally polarized abelian surfaces as follows:

Definition 15. *For a prime $\ell \neq p$, we denote by $\mathcal{G}(\ell, p)$ the graph defined as*

- *The vertex set of $\mathcal{G}(\ell, p)$ is $\mathcal{V}_p$, and*
- *The edges in $\mathcal{G}(\ell, p)$ are (ℓ, ℓ)-isogenies between two vertices, where two isogenies with the same kernel are identified as a single edge.*

Recall from (2.1) that $\mathcal{G}(\ell, p)$ is an $S(\ell) = (\ell+1)(\ell^2+1)$-regular multigraph. An important fact, proved in [20, Lemma 4], is that $\mathcal{G}(\ell, p)$ is connected.

2.3 Algorithm for computing $(2, 2)$-isogenies

In this subsection, we describe the algorithms used for computing $(2, 2)$-isogenies, which will be necessary later. For simplicity, we assume that principally polarized abelian surfaces in this subsection are all indecomposable (i.e., isomorphic to the Jacobian variety of a genus-2 curve).

In what follows, let $\mathrm{M}_{\mathbb{F}_p}$ denote the cost of a multiplication in $\mathbb{F}_p$. Note that the cost of an addition in $\mathbb{F}_p$ is bounded by $\mathrm{M}_{\mathbb{F}_p}$, and that the cost of an inversion is in $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$. First, we need to define an efficient representation of a $(2, 2)$-isogeny between principally polarized abelian surfaces:

Definition 16. *Let A and A' be principally polarized abelian surfaces over $\mathbb{F}_{p^4}$, and $\phi : A \to A'$ be a $(2, 2)$-isogeny between them. An **efficient representation** of ϕ is a data $\mathcal{D}$ of size $O(\log(p))$ together with the following three algorithms:*

- $\mathsf{Domain}(\mathcal{D})$ *returns one of the defining equations of the genus-2 curve underlying A in $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$ time.*
- $\mathsf{Codomain}(\mathcal{D})$ *returns one of the defining equations of the genus-2 curve underlying A' in $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$ time.*
- $\mathsf{Eval}(\mathcal{D}, P)$ *takes a point $P \in A(\mathbb{F}_{p^{4n}})$ with an integer $n \geq 1$, and returns $\phi(P)$ in $O(n \log(p)\mathrm{M}_{\mathbb{F}_p})$ time.*

For an example of efficient representations of $(2, 2)$-isogenies ϕ, see Appendix A. In the following, we suppose that every $(2, 2)$-isogeny is expressed in an efficient representation. For constructing the LTZ hash function and our collision attack for it, we define the following algorithms:

- $\mathsf{CodomainInvariant}(\phi)$: It takes a $(2,2)$-isogeny $\phi : A \to A'$, and returns the absolute invariants of A'.
- $\mathsf{Conjugate}(\phi)$: It takes a $(2,2)$-isogeny $\phi : A \to A'$ over $\mathbb{F}_{p^4}$, and returns an efficient representation of its p^2-conjugate $\phi^{(p^2)} : A^{(p^2)} \to A'^{(p^2)}$.
- $\mathsf{Dual}(\phi)$: It takes a $(2,2)$-isogeny $\phi : A \to A'$, and returns an efficient representation of its dual $\hat{\phi} : A' \to A$.
- $\mathsf{GoodExtension}(\phi, i)$: It takes a $(2,2)$-isogeny $\phi : A \to A'$ and $i \in \{0, \ldots, 7\}$, and returns an efficient representation of the i-th good extension of ϕ, where the order of the good extensions is determined by a representation of ϕ.
- $\mathsf{IsGoodExtension}(\phi, \phi')$: It takes $(2,2)$-isogenies $\phi : A \to A'$ and $\phi' : A' \to A''$, and returns whether ϕ' is a good extension of ϕ.

These algorithms can be implemented with time complexity $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$, and space complexity $O(\log(p))$. An explicit description of each algorithm is given in Appendix A. To simplify the notation, we will omit writing Dual and $\mathsf{Conjugate}$ in our algorithms; instead, we denote $\mathsf{Dual}(\phi)$ by $\hat{\phi}$, and $\mathsf{Conjugate}(\phi)$ by $\phi^{(p^2)}$.

3 LeGrow-Ti-Zobernig's hash function

A *cryptographic hash function* is a deterministic function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^d$ that takes an arbitrary-length input and returns a fixed-length hash value. It is required to satisfy the following properties:

- *pre-image resistance*: Given a hash value $h \in \{0,1\}^d$, it is computationally hard to find an input m such that $\mathcal{H}(m) = h$.
- *second pre-image resistance*: Given an input m and its hash value $h = \mathcal{H}(m)$, it is computationally hard to find an input $m' \neq m$ such that $\mathcal{H}(m') = h$.
- *collision resistance*: It is computationally hard to find two inputs m and m' such that $m \neq m'$ and $\mathcal{H}(m) = \mathcal{H}(m')$.

Note that the collision resistance is the strongest property among the three.

In this section, we review the construction of the cryptographic hash function proposed by LeGrow, Ti, and Zobernig [20], which is referred as to the *LTZ hash function*. After that, we focus on the collision resistance of the LTZ hash function and describe an existing attack method that solve it.

3.1 Construction

Roughly speaking, the LTZ hash function is derived from the CDS hash function by replacing the $(2,2)$-isogeny graph of superspecial principally polarized abelian surfaces with the graph $\mathcal{G}(2,p)$ introduced in Definition 15. More concretely, for a prime $p > 5$, the LTZ hash function

$$\mathcal{H}_{\mathrm{LTZ}} : \{0,1\}^* \longrightarrow (\mathbb{F}_{p^4})^3$$

is described by the following procedure:

Algorithm 1: $\mathcal{H}_{\mathrm{LTZ}}$

Input: A bit string of a message $m \in \{0,1\}^*$.
Output: The hash value of m.
Let $\phi_0 : A_{-1} \to A_0$ be the $(2,2)$-isogeny defined as in Section 3.2;
Parse m as an octal sequence $m_n \cdots m_2 m_1$ with $m_i \in \{0,\ldots,7\}$;
$\phi \leftarrow \phi_0$;
for $i \leftarrow 1$ **to** n **do**
  $\phi \leftarrow \mathsf{GoodExtension}(\phi, m_i)$;
return $\mathsf{CodomainInvariant}(\phi)$;

Set-up. We fix an initial edge $\phi_0 : A_{-1} \to A_0$ on the graph $\mathcal{G}(2,p)$ appropriately (we explain later how to choose this). Parse an input $m \in \{0,1\}^*$ as an octal sequence $m_n \cdots m_2 m_1$ where each $m_i \in \{0,\ldots,7\}$.

Walking. There are 8 good extensions of ϕ_0, and we label them $\phi_{0,0},\ldots,\phi_{0,7}$ in a deterministic order. Then, define the next isogeny $\phi_1 = \phi_{0,m_1} : A_0 \to A_1$ and define $\phi_{1,0},\ldots,\phi_{1,7}$ be its 8 good extensions. Next, setting $\phi_2 = \phi_{1,m_2}$, repeat this procedure until the last isogeny $\phi_n : A_{n-1} \to A_n$ is obtained.

Output. The last vertex A_n must be isomorphic to the Jacobian variety of some genus-2 curve H_n by Remark 9, and then we output the absolute invariants (j_1, j_2, j_3) of H_n. We note that j_1, j_2, and j_3 all belong to $\mathbb{F}_{p^4}$, since $A_n \in \mathcal{V}_p$ is defined over $\mathbb{F}_{p^4}$.

Here, the reason why we restrict the isogenies to good extensions of the previous isogeny is to avoid trivial collisions (see [5, Section 7.1] for more details). For the reader's convenience, we give an explicit pseudocode in Algorithm 1. If necessary, one can convert the output of $\mathcal{H}_{\mathrm{LTZ}}(m)$ into a fixed-length bit string.

3.2 How to choose the initial edge

Although the original paper [20] does not make explicit how the initial edge ϕ_0 is defined, we must choose it carefully to avoid trivial collisions. In this subsection, we discuss an appropriate choice.

In what follows, we restrict our attention to the case where $p \equiv 2, 3 \pmod 5$. Let A_{start} be the Jacobian variety of the genus-2 curve $y^2 = x^5 - 1$ over $\mathbb{F}_p$, then by Proposition 8, this is supersingular but non-superspecial. Also, since A_{start} is the unique principally polarized abelian surface (up to isomorphism) which has the automorphism group of order 10, the latter assertion of Theorem 14 implies that A_{start} belongs to $\mathcal{V}_p$, as shown in [20, Lemma 5].

Remark 17. It is not appropriate to simply set $A_0 \coloneqq A_{\mathrm{start}}$. The reason is that, due to the large automorphism group of A_{start}, it has only three adjacent vertices (cf. [10, Section 4.15]), and therefore no matter how the initial edge ϕ_0 is chosen, there are two distinct good extensions of ϕ_0 with the same codomain. This leads to trivial collisions in the LTZ hash function.

Then, our solution is to employ a technique similar to that used in the CDS hash function, which is described as follows: we compute a $(2^{d+1}, 2^{d+1})$-isogeny with domain A_{start} deterministically, which can be decomposed into

$$A_{\mathrm{start}} \longrightarrow A_{-d} \longrightarrow A_{-d+1} \longrightarrow \cdots \longrightarrow A_{-1} \xrightarrow{\phi_0} A_0$$

for a non-negative integer $d \geq 0$, and define $\phi_0 : A_{-1} \to A_0$ to be the initial edge in the LTZ hash function (for example, in analogy with the CDS hash function, we can choose $d := 10$).

Note that this choice is not secure in the setting of the CDS hash function as suggested in [4, Section 5.4]. This attack on the CDS hash function uses the fact that the construction of ϕ_0 starts from the product of two supersingular elliptic curves. Since this is not the case for the LTZ hash function, the attack does not apply to it (at least in a trivial way). Nevertheless, the choice of ϕ_0 still remains an important issue, and a more conservative approach would be to employ some method to conceal the path from A_{start} to A_0.

3.3 Underlying mathematical problems

In this subsection, we provide a brief review of the security analysis of the LTZ hash function, as presented in [20, Section 5]. We introduce the following two problems as the underlying mathematical problems of the LTZ hash function:

Problem 18. Given an edge $\phi_0 : A_{-1} \to A_0$ in $\mathcal{G}(2, p)$ and a vertex $A \in \mathcal{V}_p$, find a good $(2, 2)$-sequence

$$A_{-1} \xrightarrow{\phi_0} A_0 \longrightarrow A_1 \longrightarrow \cdots \longrightarrow A_n \longrightarrow A$$

for some $n \in \mathbb{N}$.

Problem 19. Given an edge $\phi_0 : A_{-1} \to A_0$ in $\mathcal{G}(2, p)$, find $A \in \mathcal{V}_p$ and two distinct good $(2, 2)$-sequences

$$A_{-1} \xrightarrow{\phi_0} A_0 \longrightarrow A_1 \longrightarrow \cdots \longrightarrow A_n \longrightarrow A$$

and

$$A_{-1} \xrightarrow{\phi_0} A_0 \longrightarrow B_1 \longrightarrow \cdots \longrightarrow B_{n'} \longrightarrow A$$

for some $n, n' \in \mathbb{N}$.

It is obvious that the pre-image resistance and the second pre-image resistance of the LTZ hash function both reduce to Problem 18, and the collision resistance of the LTZ hash function reduces to Problem 19. Moreover, Problem 19 reduces to Problem 18. Therefore, for the security of the LTZ hash function, we need to analyze the hardness of Problem 19.

In the original paper [20], the authors claimed that the best known algorithm to solve Problem 19 is a Pollard-rho style algorithm, which is roughly described by the following procedure:

Set-up. Let N be $\lfloor \log_8(p^3) \rfloor$.

Storing. Compute all the $(2^n, 2^n)$-isogenies from A with $n \leq N$, and store the absolute invariants of the genus-2 curves underlying their codomains together with the data specifying the isogenies in a hash table.

Output. Find a pair of isogenies ψ, ψ' in the hash table such that the absolute invariants of the underlying genus-2 curves of their codomains are equal, and we output them.

The number of $(2^n, 2^n)$-isogenies from A with $n \leq N$ is given by

$$\sum_{n=1}^{N} S(2^n) = \sum_{n=1}^{N} (15 \cdot 8^{n-1}) = 15 \cdot \frac{8^N - 1}{7} = \Theta(p^3),$$

with the notation as in (2.1). On the other hand, the cardinality of $\mathcal{V}_p$ is $\Theta(p^6)$ by Theorem 14, and therefore the birthday paradox implies that the above algorithm succeeds in solving Problem 19 with non-negligible probability. This algorithm requires computing and storing $O(p^3)$ vertices and edges in $\mathcal{G}(2, p)$, and thus its time complexity and space complexity are both $\widetilde{O}(p^3)$.

As a consequence of the above discussion, collisions on the LTZ hash function can be found in time and space complexities $\widetilde{O}(p^3)$. Accordingly, the authors in the original paper [20] claimed that one can choose $p \approx 2^{\lambda/3}$ to achieve the λ-bit security level for the LTZ hash function.

Implicit security assumption. Although not stated explicitly in the original paper [20], the above discussion requires the following assumption related to the isogenies computed in the LTZ hash function:

Assumption 20. For sufficiently large n, the codomains of the $(2^n, 2^n)$-isogenies with fixed domain $A \in \mathcal{V}_p$ are almost uniformly distributed over $\mathcal{V}_p$.

This assumption is based on the weaker conjecture stated below, which is inspired by the conjecture in the superspecial case (cf. [5, Conjecture 3]).

Conjecture 21. For every two vertices $A, A' \in \mathcal{V}_p$, there exists a $(2^n, 2^n)$-isogeny from A to A' for some n.

We note that Conjecture 21 is used implicitly in Problem 18. To the best of our knowledge and efforts, there is no proof of this conjecture. We leave the proof of this conjecture along with the analysis of Assumption 20 as future work.

4 Theoretical foundations

In this section, we describe the mathematical facts and a conjecture underlying our collision attack on the LTZ hash function. We continue to assume that $p > 5$ is a prime satisfying $p \equiv 2, 3 \pmod 5$, as stated in Section 3.2.

4.1 On the number of vertices defined over $\mathbb{F}_{p^2}$

The purpose of this subsection is to present a conjecture (Conjecture 28) on the number of vertices in $\mathcal{G}(2,p)$, as given in Definition 15, that are defined over $\mathbb{F}_{p^2}$. For this purpose, we begin by establishing several lemmas:

Lemma 22. *For any supersingular principally polarized abelian surface $A \in \mathcal{V}_p$, the characteristic polynomial of A over $\mathbb{F}_{p^4}$ is given as $(X+p^2)^4$. Therefore, all the edges on the graph $\mathcal{G}(2,p)$ are defined over $\mathbb{F}_{p^4}$.*

Proof. Let A_{start} be the Jacobian variety of the genus-2 curve $y^2 = x^5 - 1$, which lies in the vertex set $\mathcal{V}_p$, from the assumption that $p \equiv 2, 3 \pmod 5$. Recall from Proposition 12 that the characteristic polynomial of A_{start} over $\mathbb{F}_p$ is given by

$$\Phi(X) = X^4 + p^2 = \prod_{i=1}^{4}(X - \alpha_i),$$

where $\alpha_1, \ldots, \alpha_4$ are the fourth roots of $-p^2$. Then, the characteristic polynomial of A_{start} over $\mathbb{F}_{p^4}$ is given by

$$\Phi_4(X) = \prod_{i=1}^{4}(X - \alpha_i^4) = (X+p^2)^4$$

by using Proposition 10. Equivalently, the p^4-th power Frobenius endomorphism on A_{start} is identical to the multiplication-by-$(-p^2)$ map on A_{start}, which implies that all the $(2,2)$-isogenies from A_{start} are defined over $\mathbb{F}_{p^4}$. Therefore, by Tate's isogeny theorem [26, Theorem 1(c)], each vertex adjacent to A_{start} on $\mathcal{G}(2,p)$ has the characteristic polynomial over $\mathbb{F}_{p^4}$ equal to $(X+p^2)^4$, which is the same as that of A_{start}.

Repeating this inductively, we conclude from the connectivity of $\mathcal{G}(2,p)$ that the characteristic polynomial of any vertex $A \in \mathcal{V}_p$ over $\mathbb{F}_{p^4}$ is given by $(X+p^2)^4$. Consequently, applying Tate's isogeny theorem again, we find that all the edges on $\mathcal{G}(2,p)$ are defined over $\mathbb{F}_{p^4}$, as desired. □

Lemma 23. *Let $A \in \mathcal{V}_p$ be a supersingular principally polarized abelian surface over $\mathbb{F}_{p^2}$. Then, the rank of $A(\mathbb{F}_{p^2})[2]$ is 2.*

Proof. First, we show that the characteristic polynomial $\Phi(X) = \prod_{i=1}^{4}(X - \alpha_i)$ of A over $\mathbb{F}_{p^2}$ is given by $\Phi(X) = (X^2+p^2)^2$. It follows from Proposition 10 that the characteristic polynomial of A over $\mathbb{F}_{p^4}$ is given by $\prod_{i=1}^{4}(X - \alpha_i^2)$, which is equal to $(X+p^2)^4$ by Lemma 22. Hence, we have that

$$(X^2+p^2)^4 = \prod_{i=1}^{4}(X^2 - \alpha_i^2) = \prod_{i=1}^{4}(X - \alpha_i) \cdot \prod_{i=1}^{4}(X + \alpha_i),$$

and this polynomial is divisible by $\prod_{i=1}^{4}(X - \alpha_i) = \Phi(X) \in \mathbb{Z}[X]$. Since the only possible polynomial of degree 4 with integer coefficients that divides $(X^2+p^2)^4$ is $(X^2+p^2)^2$, the claim follows as desired.

Then, again by Proposition 10, the order of $A(\mathbb{F}_{p^2})$ equals $\Phi(1) = (p^2+1)^2$, which is divisible by 4 and not by 8. Hence, the rank of $A(\mathbb{F}_{p^2})[2]$ is 1 or 2. Next, let P be a point of $A(\mathbb{F}_{p^2})[2]$, and let Q_1, Q_2, Q_3 be points on A of order 2 such that (P, Q_1, Q_2, Q_3) is a basis of $A[2]$. Denoting by π the p^2-th power Frobenius endomorphism on A, we obtain that $Q_i + \pi(Q_i) \in A(\mathbb{F}_{p^2})[2]$ for all $i \in \{1,2,3\}$. If $Q_i + \pi(Q_i) \neq P$ for some i, then either Q_i or $Q_i + \pi(Q_i)$ is a point in $A(\mathbb{F}_{p^2})[2]$ different from P. If $Q_i + \pi(Q_i) = P$ for all i, then $Q_1 + Q_2$ is a point in $A(\mathbb{F}_{p^2})[2]$ different from P. In either case, the rank of $A(\mathbb{F}_{p^2})[2]$ is at least 2. Summarizing the above discussion, the rank of $A(\mathbb{F}_{p^2})[2]$ equals 2, as desired. □

Lemma 24. *Let $A \in \mathcal{V}_p$ be a supersingular principally polarized abelian surface over $\mathbb{F}_{p^2}$. Then, there exists a symplectic basis $(P_1, P_2; Q_1, Q_2)$ of $A[2]$ such that*

- *both P_1 and P_2 are defined over $\mathbb{F}_{p^2}$,*
- *$(Q_1 + \pi(Q_1), Q_2 + \pi(Q_2)) = (P_1, P_2)$ or (P_2, P_1),*

where π denotes the p^2-th power Frobenius endomorphism on A.

Proof. First, consider the map

$$A[2] \longrightarrow A(\mathbb{F}_{p^2})[2]\,;\, Q \longmapsto Q + \pi(Q).$$

This is a group homomorphism whose kernel is $A(\mathbb{F}_{p^2})[2]$, and hence it is surjective by Lemma 23. Thus, for any $P' \in A(\mathbb{F}_{p^2})[2]$, there exists $Q \in A[2]$ such that $Q + \pi(Q) = P'$. Let e_2 denote the 2-Weil pairing on A, then for $P, P' \in A(\mathbb{F}_{p^2})[2]$ with $Q \in A[2]$ and $Q + \pi(Q) = P'$, we have

$$e_2(P, P') = e_2(P, Q + \pi(Q)) = e_2(P, Q)e_2(P, \pi(Q)) = e_2(P, Q)^2 = 1,$$

which shows that e_2 is trivial on $A(\mathbb{F}_{p^2})[2]$. Therefore, one can choose a symplectic basis $(P_1, P_2; Q_1, Q_2)$ of $A[2]$ such that both P_1 and P_2 are in $A(\mathbb{F}_{p^2})$.

Let $R_1 \coloneqq Q_1 + \pi(Q_1)$ and $R_2 \coloneqq Q_2 + \pi(Q_2)$, which are points in $A(\mathbb{F}_{p^2})[2]$. Then, we have

$$\begin{aligned} e_2(Q_1, R_2)e_2(Q_2, R_1) &= e_2(Q_1, \pi(Q_2))e_2(Q_2, \pi(Q_1)) \\ &= e_2(Q_1, \pi(Q_2))^2 = 1. \end{aligned}$$

This means that $e_2(Q_1, R_2) = e_2(Q_2, R_1)$. If $e_2(Q_1, R_2) = e_2(Q_2, R_1) = 1$, then we have $R_1 = P_1$ and $R_2 = P_2$. Otherwise (i.e., $e_2(Q_1, R_2) = e_2(Q_2, R_1) = -1$), we have $R_1 \in \{P_2, P_1 + P_2\}$ and $R_2 \in \{P_1, P_1 + P_2\}$ with $R_1 \neq R_2$.

- In the case where $R_1 = P_2$ and $R_2 = P_1$, there is nothing more to prove.
- In the case where $R_1 = P_1 + P_2$, it follows that $R_2 = P_1$. By replacing (P_1, Q_2) with $(P_1 + P_2, Q_1 + Q_2)$, one may assume that $R_1 = P_1$ and $R_2 = P_2$.
- In the case where $R_1 = P_2$, it follows that $R_2 = P_1 + P_2$. By replacing (P_2, Q_1) with $(P_1 + P_2, Q_1 + Q_2)$, one may assume that $R_1 = P_1$ and $R_2 = P_2$.

In each case, the resulting symplectic basis $(P_1, P_2; Q_1, Q_2)$ of $A[2]$ satisfies the desired condition. □

Here, we introduce the following definition:

Definition 25. *For a supersingular principally polarized abelian surface $A \in \mathcal{V}_p$ over $\mathbb{F}_{p^2}$, let $(P_1, P_2; Q_1, Q_2)$ be a symplectic basis of $A[2]$ in Lemma 24. Then, we classify A into the following two types:*

*(i) We say that A is of **Type I** if $Q_1 + \pi(Q_1) = P_1$ and $Q_2 + \pi(Q_2) = P_2$,*
*(ii) We say that A is of **Type II** if $Q_1 + \pi(Q_1) = P_2$ and $Q_2 + \pi(Q_2) = P_1$,*

where π denotes the p^2-th power Frobenius endomorphism on A. In other words, the abelian surface A is Type I if and only if $e_2(P, \pi(P)) = -1$ for some $P \in A[2]$, and is Type II if and only if $e_2(P, \pi(P)) = 1$ for all $P \in A[2]$.

The following lemma ensures that the type of A is invariant under isomorphisms over $\overline{\mathbb{F}}_p$, and therefore, the above definition is well-defined:

Lemma 26. *Let $A, A' \in \mathcal{V}_p$ be a supersingular principally polarized abelian surface over $\mathbb{F}_{p^2}$, where A is isomorphic to A'. Then A is of Type I (resp. Type II) if and only if A' is of Type I (resp. Type II).*

Proof. By Theorem 14, the automorphism group of A' is either trivial (i.e., equal to $\{\pm 1\}$) or isomorphic to $\mathbb{Z}/10\mathbb{Z}$. In the following, we distinguish the two cases separately, depending on the automorphism group of A'.

If $\mathrm{Aut}(A') = \{\pm 1\}$, let $\iota : A \to A'$ be an isomorphism. Since $\iota^{(p^2)} \circ \iota^{-1}$ is an automorphism of A', we have $\iota^{(p^2)} \circ \iota^{-1} = [\pm 1]$, that is, $\iota^{(p^2)} = [\pm 1]\iota$. Hence, for all $P \in A[2]$, we have

$$\begin{aligned} e_2(P, \pi(P)) = e_2(\iota^{(p^2)}(P), \iota^{(p^2)}(\pi(P))) &= e_2([\pm 1]\iota(P), \pi(\iota(P))) \\ &= e_2(\iota(P), \pi(\iota(P))). \end{aligned}$$

Since $\iota(P)$ runs through all $A'[2]$ as P runs through $A[2]$, the types of A and A' are the same.

Otherwise (i.e., $\mathrm{Aut}(A') \cong \mathbb{Z}/10\mathbb{Z}$), as described after Proposition 8, both A and A' are isomorphic to the Jacobian variety of $H : y^2 = x^5 - 1$. Let ι denote such an isomorphism from A to $\mathrm{Jac}(H)$. Since $\iota^{(p^2)} \circ \iota^{-1}$ is an automorphism of $\mathrm{Jac}(H)$, we have $\iota^{(p^2)} \circ \iota^{-1} = [\pm\zeta_5^k]$ for some $k \in \{0, 1, 2, 3, 4\}$. Here, for each k, we write $[\pm\zeta_5^k]$ for the automorphism of $\mathrm{Jac}(H)$ induced from $(x, y) \mapsto (\zeta_5^k x, \pm y)$ on H, where ζ_5 is a primitive fifth roots of unity in $\overline{\mathbb{F}}_p$.

– In the case where $k = 0$, then we obtain $\iota^{(p^2)} = [\pm 1]\iota$. Let Q be the order-2 point $(1, 0) - (\zeta_5, 0)$ on $\mathrm{Jac}(H)$ and $P \coloneqq \iota^{-1}(Q) \in A[2]$. Then, we have

$$\begin{aligned} e_2(P, \pi(P)) = e_2(\iota^{(p^2)}(P), \iota^{(p^2)}(\pi(P))) &= e_2([\pm 1]\iota(P), \pi(\iota(P))) \\ &= e_2(Q, \pi(Q)) = -1, \end{aligned}$$

since $\pi(Q) = (1, 0) - (\zeta_5^4, 0)$ from the assumption that $p \equiv 2, 3 \pmod 5$.

– In the case where $k \neq 0$, then we obtain $\iota^{(p^2)} = [\pm\zeta_5^k]\iota$. Let Q be the order-2 point $(1,0) - \infty$ on $\mathrm{Jac}(H)$ and $P \coloneqq \iota^{-1}(Q) \in A[2]$. Then, we have

$$\begin{aligned} e_2(P, \pi(P)) = e_2(\iota^{(p^2)}(P), \iota^{(p^2)}(\pi(P))) &= e_2([\pm\zeta_5^k]\iota(P), \pi(\iota(P))) \\ &= e_2([\zeta_5^k]Q, \pi(Q)) = -1, \end{aligned}$$

since $[\zeta_5^k]Q = (\zeta_5^k, 0) - \infty$ and $\pi(Q) = (1,0) - \infty$.

In any case, there exists $P \in A[2]$ such that $e_2(P, \pi(P)) = -1$, and this tells us that A is of Type I. Similarly, one can show that A' is of Type I, and hence the type of A and A' are the same. □

Remark 27. In the discussion of this subsection so far, the congruence condition that $p \equiv 2, 3 \pmod 5$ is employed essentially, *only* at the beginning of the proof of Lemma 22. It should be noted that if $p \equiv 1, 4 \pmod 5$, then all $A' \in \mathcal{V}_p$ have trivial automorphism groups, and thus the latter case distinction in the proof of Lemma 26 is not needed.

Moreover, even if $p \equiv 1, 4 \pmod 5$, as long as there exists *at least one* $A \in \mathcal{V}_p$ whose characteristic polynomial over $\mathbb{F}_{p^4}$ is $(X + p^2)^4$, then Lemma 22 still holds by the same proof as the original one. Therefore, in that case, all the subsequent statements after Lemma 22 remain valid; in particular, the type classification in Definition 25 is also well-defined.

We are now ready to state our conjecture:

Conjecture 28. For primes $p > 5$ with $p \equiv 2, 3 \pmod 5$, we define

$$\begin{aligned} N_{1,p} &\coloneqq \#\{A \in \mathcal{V}_p \mid A \text{ is defined over } \mathbb{F}_{p^2} \text{ and of Type I}\}, \text{ and} \\ N_{2,p} &\coloneqq \#\{A \in \mathcal{V}_p \mid A \text{ is defined over } \mathbb{F}_{p^2} \text{ and of Type II}\}. \end{aligned}$$

Then, we have the following:

– If $p \equiv 1 \pmod 4$, then

$$N_{1,p} = \left(\frac{p+1}{2}\right)\left(\frac{p-1}{4}\right)^2, \quad N_{2,p} = \frac{2}{3}\left(\frac{p+1}{2}\right)\left(\frac{p-1}{4}\right)^2.$$

– If $p \equiv 3 \pmod 4$, then

$$N_{1,p} = \left(\frac{p-1}{2}\right)\left(\frac{p+1}{4}\right)^2, \quad N_{2,p} = 0.$$

We have verified that Conjecture 28 holds for any prime p such that $5 < p < 100$ and $p \equiv 2, 3 \pmod 5$ by a computer experiments.

Remark 29. For a prime $5 < p < 100$, even if $p \equiv 1, 4 \pmod 5$, it is also verified that every $A \in \mathcal{V}_p$ has the characteristic polynomial $(X + p^2)^4$ over $\mathbb{F}_{p^4}$, and that Conjecture 28 is true (note that, the definitions of $N_{1,p}$ and $N_{2,p}$ make sense by the discussion in Remark 27).

4.2 On the number of edges with good p^2-conjugate

In our collision attack on the LTZ hash function described in Section 5, an edge on the graph $\mathcal{G}(2,p)$ with the following property plays a central role:

Definition 30. *For an edge $\phi : A \to A'$ on the graph $\mathcal{G}(2,p)$ where A is defined over $\mathbb{F}_{p^2}$, we say that ϕ has the **good p^2-conjugate** if it is a good extension of the dual $\hat{\phi}^{(p^2)} : A'^{(p^2)} \to A^{(p^2)} = A$ of its p^2-conjugate.*

Proposition 31. *For a supersingular principally polarized abelian surface $A \in \mathcal{V}_p$ over $\mathbb{F}_{p^2}$, we have the following:*

(i) If A is of Type I, then there exist exactly twelve $(2,2)$-isogenies from A that are not defined over $\mathbb{F}_{p^2}$. Among these, eight have the good p^2-conjugate.
(ii) If A is of Type II, then there exist exactly eight $(2,2)$-isogenies from A that are not defined over $\mathbb{F}_{p^2}$. Any of them has the good p^2-conjugate.

Proof. For a $(2,2)$-isogeny ϕ from A with kernel G, we have that

- ϕ is not defined over $\mathbb{F}_{p^2}$ if and only if $G \neq \pi(G)$,
- ϕ has the good p^2-conjugate if and only if $G \cap \pi(G) = \emptyset$,

where π denotes the p^2-th power Frobenius endomorphism on A.

Let $(P_1, P_2; Q_1, Q_2)$ is a symplectic basis of $A[2]$ satisfying the conditions in Lemma 24. All the maximal isotropic subgroups of $A[2]$ together with the action of π on them are summarized in Table 2 for Type I and in Table 3 for Type II. These results prove the proposition. □

Finally, we assume the following heuristic based on Conjecture 28, which will be required for the discussion of the termination of our collision attack:

Heuristic 32. Let ϕ be an edge on the graph $\mathcal{G}(2,p)$. Then, there exists at least one good $(2,2)$-sequence $(\phi, \phi_1, \ldots, \phi_n)$ with $n \leq \log_2(p)$ such that the codomain of ϕ_n is defined over $\mathbb{F}_{p^2}$ and $\hat{\phi}_n^{(p^2)}$ is a good extension of ϕ_n.

In the following, we explain why Heuristic 32 is reasonable. By Theorem 14, the cardinality of $\mathcal{V}_p$ is less than $p^6/2880$. Moreover, according to Conjecture 28, the number of elements of $\mathcal{V}_p$ defined over $\mathbb{F}_{p^2}$ is greater than $p^3/32$. Hence, the ratio of the latter to the former is greater than $p^3/32 \cdot 2880/p^6 = 90/p^3$. On the other hand, the number of good $(2,2)$-extensions $(\phi, \phi_1, \ldots, \phi_n)$ with $n \coloneqq \lfloor \log_2(p) \rfloor$ is given by $8^n > p^3/8$. By assuming the rapid mixing property of good extensions, we expect the codomains of ϕ_n to be almost uniformly distributed over $\mathcal{V}_p$. This implies that the set of the codomains of ϕ_n contains approximately $90/8 = 11.25$ elements of $\mathcal{V}_p$ defined over $\mathbb{F}_{p^2}$. In addition, for every ϕ_n with such a codomain, Proposition 31 says that, the probability that $\hat{\phi}_n^{(p^2)}$ has the good p^2-conjugate is at least $2/3$. Therefore, we conclude that Heuristic 32 is reasonable.

Table 2. Maximal isotropic subgroups G of $A[2]$, its p^2-th power Frobenius conjugate $\pi(G)$, and the intersection $G \cap \pi(G)$ for a Type I abelian surface $A \in \mathcal{V}_p$.

G	$\pi(G)$	$G \cap \pi(G)$
$\langle P_1, P_2 \rangle$	$= G$	G
$\langle P1_1 + P_2, Q_1, Q_2 \rangle$	$= G$	G
$\langle P_1 + P_2, P_1 + Q_1 + Q_2 \rangle$	$= G$	G
$\langle P_1, Q_2 \rangle$	$\langle P_1, P_2 + Q_2 \rangle$	$\langle P_1 \rangle$
$\langle P_2, Q_1 \rangle$	$\langle P_2, P_1 + Q_1 \rangle$	$\langle P_2 \rangle$
$\langle Q_1, Q_2 \rangle$	$\langle P_1 + Q_1, P_2 + Q_2 \rangle$	$\emptyset$
$\langle Q_1, P_2 + Q_2 \rangle$	$\langle P_1 + Q_1, Q_2 \rangle$	$\emptyset$
$\langle P_1 + Q_2, P_1 + P_2 + Q_1 \rangle$	$\langle P_1 + P_2 + Q_2, P_2 + Q_1 \rangle$	$\emptyset$
$\langle P_1 + Q_2, P_2 + Q_1 \rangle$	$\langle Q_1 + Q_2, P_1 + P_2 + Q_1 \rangle$	$\emptyset$

Table 3. Maximal isotropic subgroups G of $A[2]$, its p^2-th power Frobenius conjugate $\pi(G)$, and the intersection $G \cap \pi(G)$ for a Type II abelian surface $A \in \mathcal{V}_p$.

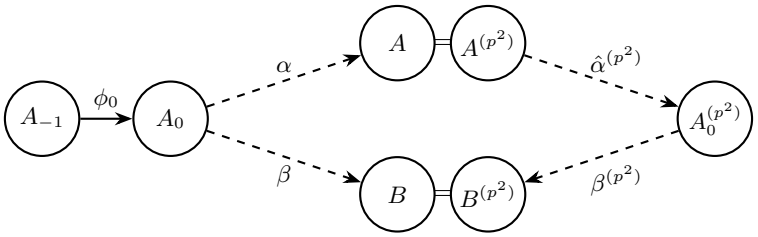
G	$\pi(G)$	$G \cap \pi(G)$
$\langle P_1, P_2 \rangle$	$= G$	G
$\langle P_1 + P_2, Q_1, Q_2 \rangle$	$= G$	G
$\langle P_1 + P_2, P_1 + Q_1 + Q_2 \rangle$	$= G$	G
$\langle P_1, Q_2 \rangle$	$= G$	G
$\langle P_2, Q_1 \rangle$	$= G$	G
$\langle P_1, P_2 + Q_2 \rangle$	$= G$	G
$\langle P_2, P_1 + Q_1 \rangle$	$= G$	G
$\langle Q_1, Q_2 \rangle$	$\langle P_2 + Q_1, P_1 + Q_2 \rangle$	$\emptyset$
$\langle Q_1, P_2 + Q_2 \rangle$	$\langle P_2 + Q_1, P_1 + P_2 + Q_2 \rangle$	$\emptyset$
$\langle P_1 + Q_1, Q_2 \rangle$	$\langle P_1 + P_2 + Q_1, P_1 + Q_2 \rangle$	$\emptyset$
$\langle P_1 + Q_1, P_2 + Q_2 \rangle$	$\langle Q_1 + Q_2, P_1 + P_2 + Q_1 \rangle$	$\emptyset$

5 Our collision attack on the LTZ hash function

In this section, we describe our algorithm for finding a collision on the LTZ hash function. We first outline the attack in Section 5.1, and then provide its concrete pseudocode (Algorithm 2) in Section 5.2. The computational complexity of this algorithm is estimated in Theorem 34.

5.1 Overview

Our collision attack is based on the idea in [9], and can be sketched as follows. Let $\phi_0 : A_{-1} \to A_0$ be the initial edge of the LTZ hash function as defined in Section 3.2. First, we find distinct good $(2,2)$-sequences $(\phi_0, \phi_1, \ldots, \phi_n)$ and $(\phi_0, \phi'_1, \ldots, \phi'_m)$ where the codomains of ϕ_n and ϕ'_m are defined over $\mathbb{F}_{p^2}$. Let α (resp. β) be the isogeny obtained by composing $\phi_1, \ldots, \phi_n$ (resp. $\phi'_1, \ldots, \phi'_m$) and A (resp. B) be its codomain. We then obtain two isogenies $\beta^{(p^2)} \circ \hat{\alpha}^{(p^2)} \circ \alpha$ and β from A_0 to B since $A^{(p^2)} = A$ and $B^{(p^2)} = B$. The following diagram illustrates the situation:



If $\beta^{(p^2)} \circ \hat{\alpha}^{(p^2)} \circ \alpha$ induces a good $(2,2)$-sequence (that is, it is a $(2^{2m+n}, 2^{2m+n})$-isogeny), then the sequences induced from $\beta^{(p^2)} \circ \hat{\alpha}^{(p^2)} \circ \alpha$ and β give a collision on the LTZ hash function. The remaining problem is to find such sequences, and we explain how to do this in the next subsection.

5.2 Description

In this subsection, we explicitly describe our algorithm on the LTZ hash function. For this purpose, we define a tree structure of good extensions:

Definition 33. *For an edge ϕ on the graph $\mathcal{G}(2,p)$, we define the **tree of good extensions** of ϕ as follows:*

- *The root is ϕ.*
- *Each node is a good extension of its parent.*

In addition, if the codomain of a node is isomorphic to $A \in \mathcal{V}_p$ defined over $\mathbb{F}_{p^2}$, then we take the node as an isogeny to A.

In this tree structure, since each node has exactly eight children, the number of nodes at depth n is equal to 8^n.

Additionally, we define the following subroutine to search in the tree of good extensions.

- $\mathsf{DFSGoodExt}(\phi, N, \mathtt{bool})$: It takes an edge ϕ on $\mathcal{G}(2,p)$, and returns a good $(2,2)$-sequence $(\phi, \phi_2, \ldots, \phi_n)$ of length $\leq N$ such that the codomain of the last isogeny ϕ_n is defined over $\mathbb{F}_{p^2}$, by performing a depth-first search in the tree of good extensions of ϕ. If $\mathtt{bool}$ is given by $\mathtt{true}$, we continue the search until $\hat{\phi}_n^{(p^2)}$ is a good extension of ϕ_n.

Then, our collision attack is described in Algorithm 2.

We finally prove our main theorem:

Theorem 34. *Assuming Heuristic 32, Algorithm 2 always succeeds. Moreover, its time complexity is $O(p^3 \log(p) \mathrm{M}_{\mathbb{F}_p})$ and its space complexity is $O(\log(p)^2)$.*

Proof. First, we show the correctness of Algorithm 2.

- Since the codomain of ϕ_n (resp. ϕ'_m) is defined over $\mathbb{F}_{p^2}$, it is isomorphic to the codomain of $\phi_n^{(p^2)}$ (resp. $\phi'^{(p^2)}_m$). Hence, the output sequences correspond to two isogenies from A_0 to an isomorphic vertex.

Algorithm 2: Finding a collision in the LTZ hash function

Input: The initial edge $\phi_0 : A_{-1} \to A_0$ of the LTZ hash function.
Output: Two distinct good $(2,2)$-sequences that yield a collision on the LTZ hash function.

1 $\phi_1 \leftarrow \mathsf{GoodExtension}(\phi_0, 0)$;
2 $i \leftarrow 1$;
3 $\phi_1' \leftarrow \mathsf{GoodExtension}(\phi_0, i)$;
4 **while** *not* $\mathsf{GoodExtension}(\hat{\phi}_1, \phi_1')$ **do**
5 $\quad i \leftarrow i + 1$;
6 $\quad \phi_1' \leftarrow \mathsf{GoodExtension}(\phi_0, i)$;
7 $N \leftarrow \lfloor \log_2(p) \rfloor$;
8 $(\phi_1, \ldots, \phi_n) \leftarrow \mathsf{DFSGoodExt}(\phi_1, N, \texttt{false})$;
9 **if** $\mathsf{GoodExtension}(\phi_n, \hat{\phi}_n^{(p^2)})$ **then**
10 $\quad (\phi_1', \ldots, \phi_m') \leftarrow \mathsf{DFSGoodExt}(\phi_1', N, \texttt{false})$;
11 $\quad$ **return** $(\phi_1, \ldots, \phi_n, \hat{\phi}_n^{(p^2)}, \ldots, \hat{\phi}_1^{(p^2)}, {\phi_1'}^{(p^2)}, \ldots, {\phi_m'}^{(p^2)})$ *and* $(\phi_1', \ldots, \phi_m')$;
12 **else**
13 $\quad (\phi_1', \ldots, \phi_m') \leftarrow \mathsf{DFSGoodExt}(\phi_1', N, \texttt{true})$;
14 $\quad$ **return** $(\phi_1, \ldots, \phi_n)$ *and* $(\phi_1', \ldots, \phi_m', \hat{\phi}_m'^{(p^2)}, \ldots, \hat{\phi}_1'^{(p^2)}, \phi_1^{(p^2)}, \ldots, \phi_n^{(p^2)})$;

- Since ϕ_1' is a good extension of $\hat{\phi}_1$, we have that ${\phi_1'}^{(p^2)}$ (resp. $\phi_1^{(p^2)}$) is a good extension of $\hat{\phi}_1^{(p^2)}$ (resp. $\hat{\phi}_1'^{(p^2)}$). Therefore, both of the output sequences are good $(2,2)$-sequences by construction.
- Since the output sequences have different lengths, they are distinct.

Consequently, the output sequences induce a collision on the LTZ hash function, as desired.

Next, we show the termination of Algorithm 2. By Heuristic 32, the calls to DFSGoodExt in the algorithm succeed, and thus the only part whose termination is non-trivial is the **while** loop in lines 4–6. This **while** loop ends after at most six iterations. Indeed, the number of $(2,2)$-isogenies with domain A is 15, among which there are 8 good extensions of ϕ_0 and also 8 good extensions of $\hat{\phi}_1$. Hence, by the pigeonhole principle, there exists at least one ϕ_1' that is a good extension of both ϕ_0 and $\hat{\phi}_1$. Therefore, the algorithm always succeeds.

Finally, let us estimate the computational complexity of Algorithm 2. The algorithm performs a constant number of calls to GoodExtension, IsGoodExtension, and DFSGoodExt. As stated in Section 2.3, the former two run in $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$ time and $O(\log(p))$ space. In the calls to DFSGoodExt, we have

- The number of detected nodes is at most $\sum_{i=1}^{N} 8^i = (8^{N+1} - 1)/7 = O(p^3)$.
- The number of stored nodes is at most N.

For each node, we call CodomainInvariant and GoodExtension once (in particular, if the third argument of DFSGoodExt is `true`, then IsGoodExtension is also called

Table 4. Experimental results of Algorithm 2 for primes p with $p \equiv 2, 3 \pmod 5$ and $35 \leq \log_2(p) \leq 37$, showing the numbers of searched nodes in the depth-first searches in line 8 (resp. line 10 or 13) divided by p^3, the number of cases satisfying the condition in line 9, the running time, and the maximum resident set size.

		#(searched nodes)/p^3		The first	Time	Max RSS
		first	second	is good?	(seconds)	(kB)
$p \equiv 1$	Average	1.1350%	1.4764%	48/61	62,109	25,742
(mod 4)	Worst	5.6579%	7.0193%	-	256,715	26,152
$p \equiv 3$	Average	1.6520%	1.7365%	40/57	83,787	25,807
(mod 4)	Worst	9.6246%	9.5718%	-	346,364	26,168

once). Since these algorithms run in $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$ time and $O(\log(p))$ space, as explained in Section 2.3, we obtain

- The time complexity of DFSGoodExt call is in $O(p^3 \log(p)\mathrm{M}_{\mathbb{F}_p})$.
- The space complexity of DFSGoodExt call is in $O(N \log(p)) = O(\log(p)^2)$.

This constitutes the dominant part of the algorithm. □

6 Experimental results

We implemented our collision attack on the LTZ hash function, using Rust programming language. Our code in available in the Github repository

`https://github.com/hiroshi-onuki/LTZ_hash_collision`.

We conducted experiments on our attack for all primes p with $p \equiv 2, 3 \pmod 5$ and $35 \leq 3\log_2(p) \leq 37$, which are parameters expected to yield security levels between 35 and 37 bits. Our experiments were conducted on a machine running Ubuntu 22.04.5 LTS with an Intel Core i9-14900 CPU (24 cores and 48 threads), 128 GB of RAM, and Rust 1.88.0 with the `--release` flag enabled. The results are summarized in Table 4.

In Table 4, the second and third columns show the numbers of searched nodes in the depth-first searches in line 8 and line 10 or 13, respectively, divided by p^3. By Heuristic 32, these values are expected to be below 100%. The actual values are much smaller than 100%, supporting the validity of our heuristic. The fourth column shows the number of the cases where the sequence $(\phi_1, \ldots, \phi_n)$ found in line 8 satisfies that $\hat{\phi}_n^{(p^2)}$ is a good extension of ϕ_n. The fifth column reports the time required for our collision attack, and the last column reports the maximum resident set size (RSS) in kilobytes, which are measured by `time -v` commands on Ubuntu. The last column implies that our collision attack runs in a practical memory size of about 25MB. The detailed results for each prime p are shown in Tables 5, 6, 7, and 8.

Remark 35. If we use the meet-in-the-middle approach described in Section 3.3, then it is necessary to store all the vertices in the middle layer, whose number is approximately $\sqrt{p^6/2880} \approx p^3/54$. Hence, the memory required for the meet-in-the-middle approach exceeds 1 gigabyte for the primes p used in our experiments even if we assume that each vertex is stored in 1 bytes.

References

1. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 163–169. Springer, Berlin, Heidelberg (Apr 1998). https://doi.org/10.1007/bfb0054319
2. Cardona, G., Quer, J.: Field of moduli and field of definition for curves of genus 2. In: Computational aspects of algebraic curves, Lecture Notes Ser. Comput., vol. 13, pp. 71–83. World Sci. Publ., Hackensack, NJ (2005). https://doi.org/10.1142/9789812701640_0006
3. Castryck, W., Decru, T.: Multiradical isogenies. In: Arithmetic, geometry, cryptography, and coding theory 2021, Contemp. Math., vol. 779, pp. 57–89. Amer. Math. Soc., [Providence], RI (2022). https://doi.org/10.1090/conm/779/15671
4. Castryck, W., Decru, T., Kutas, P., Laval, A., Petit, C., Ti, Y.B.: KLPT^2: Algebraic pathfinding in dimension two and applications. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology – CRYPTO 2025. LNCS, vol. 16000, pp. 167–200. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01855-7_6
5. Castryck, W., Decru, T., Smith, B.: Hash functions from superspecial genus-2 curves using Richelot isogenies. J. Math. Cryptol. **14**(1), 268–292 (2020). https://doi.org/10.1515/jmc-2019-0021
6. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. Journal of Cryptology **22**(1), 93–113 (Jan 2009). https://doi.org/10.1007/s00145-007-9002-x
7. Choie, Y.J., Jeong, E.K., Lee, E.J.: Supersingular hyperelliptic curves of genus 2 over finite fields. Appl. Math. Comput. **163**(2), 565–576 (2005). https://doi.org/10.1016/j.amc.2004.03.030
8. Costello, C., Smith, B.: The supersingular isogeny problem in genus 2 and beyond. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. pp. 151–168. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44223-1_9
9. Eisenträger, K., Hallgren, S., Leonardi, C., Morrison, T., Park, J.: Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. In: ANTS XIV—Proceedings of the Fourteenth Algorithmic Number Theory Symposium. Open Book Ser., vol. 4, pp. 215–232. Math. Sci. Publ., Berkeley, CA (2020). https://doi.org/10.2140/obs.2020.4.215
10. Florit, E., Smith, B.: An atlas of the Richelot isogeny graph. In: Theory and Applications of Supersingular Curves and Supersingular Abelian Varieties, RIMS Kôkyûroku Bessatsu, vol. B90, pp. 195–219. Res. Inst. Math. Sci. (RIMS), Kyoto (2022)
11. Flynn, E.V., Ti, Y.B.: Genus two isogeny cryptography. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 286–306. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_16

12. Galbraith, S.D.: Supersingular curves in cryptography. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 495–513. Springer, Berlin, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1_29
13. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC. pp. 212–219. ACM Press (May 1996). https://doi.org/10.1145/237814.237866
14. Ibukiyama, T.: Principal polarizations of supersingular abelian surfaces. Journal of the Mathematical Society of Japan **72**(4), 1161–1180 (2020). https://doi.org/10.2969/jmsj/82528252
15. Ibukiyama, T., Katsura, T., Oort, F.: Supersingular curves of genus two and class numbers. Compositio Math. **57**(2), 127–152 (1986)
16. Igusa, J.: Arithmetic variety of moduli for genus two. Annals of Mathematics **72**(3), 612–649 (1960). https://doi.org/10.2307/1970233
17. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalali, A., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Renes, J., Soukharev, V., Urbanik, D., Pereira, G., Karabina, K., Hutchinson, A.: SIKE. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
18. Jaques, S., Schanck, J.M.: Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 32–61. Springer, Cham (Aug 2019). https://doi.org/10.1007/978-3-030-26948-7_2
19. Kunzweiler, S., Maino, L., Moriya, T., Petit, C., Pope, G., Robert, D., Stopar, M., Ti, Y.B.: Radical 2-isogenies and cryptographic hash functions in dimensions 1, 2 and 3. In: Jager, T., Pan, J. (eds.) PKC 2025, Part III. LNCS, vol. 15676, pp. 265–299. Springer, Cham (May 2025). https://doi.org/10.1007/978-3-031-91826-1_9
20. LeGrow, J.T., Ti, Y.B., Zobernig, L.: Supersingular non-superspecial abelian surfaces in cryptography. Mathematical Cryptology **3**(2), 11–23 (2023)
21. Oort, F.: Which abelian surfaces are products of elliptic curves? Math. Ann. **214**, 35–47 (1975). https://doi.org/10.1007/BF01428253
22. Oudompheng, R., Pope, G.: A note on reimplementing the castryck-decru attack and lessons learned for SageMath. Cryptology ePrint Archive, Report 2022/1283 (2022), https://eprint.iacr.org/2022/1283
23. Smith, B.: Explicit endomorphism and correspondence. Thesis, University of Sydney (2005)
24. Takashima, K.: Efficient algorithms for isogeny sequences and their cryptographic applications. In: Mathematical Modelling for Next-Generation Cryptography: CREST Crypto-Math Project, Math. Ind. (Tokyo), vol. 29, pp. 97–114. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5065-7_6
25. Tani, S.: Claw finding algorithms using quantum walk. Theoret. Comput. Sci. **410**(50), 5285–5297 (2009). https://doi.org/10.1016/j.tcs.2009.08.030
26. Tate, J.: Endomorphisms of abelian varieties over finite fields. Invent. Math. **2**, 134–144 (1966). https://doi.org/10.1007/BF01404549
27. Xing, C.: On supersingular abelian varieties of dimension two over finite fields. Finite Fields Appl. **2**(4), 407–421 (1996). https://doi.org/10.1006/ffta.1996.0024
28. Zobernig, L.: Genus 2 curves in small characteristic. arXiv:2111.07270 [math.AG] (2021), https://arxiv.org/abs/2111.07270

A An efficient representation of a $(2,2)$-isogeny

In this appendix, we give an example of efficient representations of $(2,2)$-isogenies between two indecomposable principally polarized abelian surfaces in Section 2.3, together with five algorithms such as GoodExtension.

First, any indecomposable principally polarized abelian surface over a field $k = \mathbb{F}_{p^4}$ of characteristic $p > 5$ is isomorphic to the Jacobian variety of a genus-2 curve

$$H : y^2 = f(x),$$

where $f(x) \in k[x]$ is a square-free polynomial of degree 5 or 6. Then, we consider the set of quadratic factors of $f(x)$ that are pairwise coprime as follows:

Definition 36. *A **quadratic splitting** of a square-free polynomial $f(x) \in k[x]$ of degree 5 or 6 is an unordered triple $\{f_1(x), f_2(x), f_3(x)\} \subset \bar{k}[x]$ of polynomials of degree at most 2 such that*

$$f_1(x)f_2(x)f_3(x) = f(x),$$

considered modulo the equivalence

$$\{f_1(x), f_2(x), f_3(x)\} \sim \{\alpha f_1(x), \beta f_2(x), \gamma f_3(x)\}$$

for all $\alpha, \beta, \gamma \in \bar{k}$ such that $\alpha\beta\gamma = 1$. We remark that if $\deg f(x) = 6$, then each of $f_i(x)$ has degree 2, whereas if $\deg f(x) = 5$, then one of them has degree 1.

There are 15 distinct quadratic splittings of $f(x)$, and they are in bijection with the $(2,2)$-isogenies from $\mathrm{Jac}(H)$, as shown in [23, Proposition 8.2.3]. For a $(2,2)$-isogeny $\phi : \mathrm{Jac}(H) \to \mathrm{Jac}(H')$, we let $\mathcal{D} = \{f_1(x), f_2(x), f_3(x)\}$ be the quadratic splitting of $f(x)$ corresponding to ϕ. The algorithms Domain, Codomain, Eval in Definition 16 can be implemented as follows:

- A defining equation of H is clearly recovered as $y^2 = f_1(x)f_2(x)f_3(x)$.
- A defining equation of H' can be computed as $y^2 = g_1(x)g_2(x)g_3(x)$, where we define the polynomials

$$\begin{aligned} g_1(x) &\coloneqq \frac{df_2(x)}{dx} f_3(x) - f_2(x)\frac{df_3(x)}{dx}, \\ g_2(x) &\coloneqq \frac{df_3(x)}{dx} f_1(x) - f_3(x)\frac{df_1(x)}{dx}, \\ g_3(x) &\coloneqq \frac{df_1(x)}{dx} f_2(x) - f_1(x)\frac{df_2(x)}{dx} \end{aligned} \tag{A.3}$$

  thanks to [23, Theorem 8.4.11].
- For each point $P \in \mathrm{Jac}(H)$, the image $\phi(P) \in \mathrm{Jac}(H')$ can be computed, for example, using the method of [22, Section 4.2].

In addition, one can show that these algorithms achieve the complexities stated in Definition 16. This shows that the data $\mathcal{D} = \{f_1(x), f_2(x), f_3(x)\}$ provides an efficient representation of ϕ.

Next, we fix a deterministic order $\prec$ on $\bar{k}[x]$, and represent ϕ by its efficient representation $\mathcal{D} = \{f_1(x), f_2(x), f_3(x)\}$ arranged so that $f_1(x) \prec f_2(x) \prec f_3(x)$. Let $g_1(x), g_2(x), g_3(x)$ be the polynomials defined as in (A.3) and write

$$\begin{aligned} g_1(x) &= \ell_1(x) \cdot \ell_2(x) \ \text{ with } \ \ell_1(x) \prec \ell_2(x), \\ g_2(x) &= \ell_3(x) \cdot \ell_4(x) \ \text{ with } \ \ell_3(x) \prec \ell_4(x), \\ g_3(x) &= \ell_5(x) \cdot \ell_6(x) \ \text{ with } \ \ell_5(x) \prec \ell_6(x), \end{aligned}$$

where $\ell_1(x), \ldots, \ell_6(x)$ are polynomials of degree at most 1. Then, it follows from [5, Proposition 1] that $\{g_1(x), g_2(x), g_3(x)\}$ is the quadratic splitting corresponding to the dual extension of ϕ. Also, the following assertion holds:

Proposition 37 ([5, Section 6.1]). *With the above notation, let $\phi_0, \ldots, \phi_7$ be the $(2,2)$-isogenies corresponding to the quadratic splitting*

$$\begin{aligned} &\{\ell_1(x)\ell_3(x), \ell_2(x)\ell_5(x), \ell_4(x)\ell_6(x)\}, \\ &\{\ell_1(x)\ell_3(x), \ell_2(x)\ell_6(x), \ell_4(x)\ell_5(x)\}, \\ &\{\ell_1(x)\ell_4(x), \ell_2(x)\ell_5(x), \ell_3(x)\ell_6(x)\}, \\ &\{\ell_1(x)\ell_4(x), \ell_2(x)\ell_6(x), \ell_3(x)\ell_5(x)\}, \\ &\{\ell_1(x)\ell_5(x), \ell_2(x)\ell_3(x), \ell_4(x)\ell_6(x)\}, \\ &\{\ell_1(x)\ell_5(x), \ell_2(x)\ell_4(x), \ell_3(x)\ell_6(x)\}, \\ &\{\ell_1(x)\ell_6(x), \ell_2(x)\ell_3(x), \ell_4(x)\ell_5(x)\}, \\ &\{\ell_1(x)\ell_6(x), \ell_2(x)\ell_4(x), \ell_3(x)\ell_5(x)\}, \end{aligned}$$

of $g_1(x)g_2(x)g_3(x)$, respectively. Then, they are good extensions of ϕ.

In what follows, let $\phi : \mathrm{Jac}(H) \to \mathrm{Jac}(H')$ be an edge on $\mathcal{G}(2,p)$ with $p \equiv 2, 3 \pmod 5$, where $H : y^2 = f(x)$ and $H' : y^2 = g(x)$. By Lemma 22, all the edges on $\mathcal{G}(2,p)$ are defined over $\mathbb{F}_{p^4}$, which shows that all the maximal isotropic subgroups of $\mathrm{Jac}(H)[2]$ and $\mathrm{Jac}(H')[2]$ are defined over $\mathbb{F}_{p^4}$. Then, it follows from [23, Table 8.1] that $f(x)$ and $g(x)$ are factored into polynomials of degree 1 over $\mathbb{F}_{p^4}$. Hence, for the efficient representation $\mathcal{D} = \{f_1(x), f_2(x), f_3(x)\}$ of ϕ, we have that each of $f_i(x), g_i(x), \ell_i(x)$ is a polynomial over $\mathbb{F}_{p^4}$. The five algorithms in Section 2.3 can be implemented as follows:

- $\mathsf{CodomainInvariant}(\phi)$: Output the absolute invariants of H'. Since H' is defined over $\mathbb{F}_{p^4}$, this can be computed by a constant number of arithmetic operations in $\mathbb{F}_{p^4}$, as explained in Remark 2.
- $\mathsf{Conjugate}(\phi)$: Output $\{f_1(x)^{(p^2)}, f_2(x)^{(p^2)}, f_3(x)^{(p^2)}\}$. Each polynomial $f_i(x)^{(p^2)}$ can be obtained by raising the coefficients of $f_i(x)$ to the p^2-th power, which takes $O(1)$ additions in $\mathbb{F}_{p^2}$. Indeed, if we construct $\mathbb{F}_{p^4} \cong \mathbb{F}_{p^2}[t]/(t^2 - d)$ with some $d \in \mathbb{F}_{p^2}$, then for any $u + vt \in \mathbb{F}_{p^4}$ with $u, v \in \mathbb{F}_{p^2}$, its p^2-th power can be computed as $u^{p^2} + v^{p^2} t^{(p^2)} = u + vt^{(p^2)} = u - vt$.
- $\mathsf{Dual}(\phi)$: Output $\{g_1(x), g_2(x), g_3(x)\}$.
- $\mathsf{GoodExtension}(\phi, i)$: Output ϕ_i defined in Proposition 37. It is necessary to factor each $g_i(x)$ into a product of two linear polynomials over $\mathbb{F}_{p^4}$, which requires computing a square root in $\mathbb{F}_{p^4}$.

Table 5. Experimental results on the LTZ hash function for $p \equiv 1 \pmod 4$.

p	#(searched nodes)/p^3 first	#(searched nodes)/p^3 second	The first is good?	Time (seconds)	Max RSS (kB)
3253	0.7295%	1.4687%	true	19,204	25,740
3257	2.6722%	2.7649%	false	55,338	25,780
3313	0.1344%	0.1417%	true	2,949	25,732
3373	2.9084%	2.2437%	true	57,360	25,736
3413	0.5283%	0.2829%	true	10,435	25,816
3433	1.2666%	1.9143%	true	49,374	25,816
3457	0.1248%	1.5181%	true	21,358	25,756
3517	1.4455%	1.8700%	false	42,531	25,732
3533	0.1840%	0.4340%	true	8,026	25,568
3557	1.4082%	0.2761%	false	25,222	25,764
3593	0.7774%	1.5078%	true	34,630	25,736
3613	0.7742%	1.5379%	true	36,599	25,704
3617	1.6467%	0.2175%	true	31,423	25,840
3637	0.9359%	1.1234%	true	27,196	25,796
3673	1.9685%	1.0904%	false	52,013	25,732
3677	0.8828%	2.2811%	true	41,712	26,024
3697	0.3188%	4.3422%	false	82,941	25,712
3733	0.3775%	0.0996%	true	8,141	25,708
3793	3.6662%	1.6804%	true	109,999	25,728
3797	0.1401%	1.0508%	false	19,109	25,792
3833	0.5760%	0.1109%	true	11,440	25,680
3853	1.8850%	0.7926%	true	52,193	25,564
3877	0.8926%	7.0193%	false	162,217	25,756
3917	0.5822%	2.3799%	true	56,441	25,860
4013	2.2232%	0.6236%	true	49,747	25,680

– $\mathsf{IsGoodExtension}(\phi, \phi')$: Output whether $\phi' = \phi_i$ for some $i \in \{0, \ldots, 7\}$.

From the above discussion, it follows that these algorithms run in $O(\log(p)\mathrm{M}_{\mathbb{F}_p})$ time and require $O(\log(p))$ space.

B Detailed results

In this appendix, we present the experimental results of our collision attack on the LTZ hash function, described in Section 6, for each prime p with $p \equiv 2, 3 \pmod 5$ and $35 \leq \log_2(p) \leq 37$. Tables 5 and 6 show the results for primes p with $p \equiv 1 \pmod 4$, while Tables 7 and 8 show the results for primes p with $p \equiv 3 \pmod 4$.

Table 6. Experimental results on the LTZ hash function for $p \equiv 1 \pmod 4$.

	#(searched nodes)/p^3		The first	Time	Max RSS
p	first	second	is good?	(seconds)	(kB)
4057	3.8460%	0.8768%	true	116,987	25,572
4073	0.3571%	1.1640%	true	34,901	25,564
4093	0.1093%	2.5058%	false	61,097	25,676
4133	0.4562%	5.4373%	true	119,781	25,772
4153	2.9480%	1.4037%	true	96,005	25,828
4157	0.3876%	2.4648%	true	59,477	25,760
4177	0.4718%	0.7907%	true	28,346	25,712
4217	0.1406%	1.9497%	false	48,797	25,828
4253	1.7533%	1.7847%	true	73,184	25,796
4273	0.0731%	0.5121%	false	14,036	25,804
4297	1.4714%	0.5602%	true	49,141	25,560
4337	0.1667%	0.2149%	false	9,243	26,152
4357	1.2628%	1.9573%	true	80,142	25,732
4373	1.0640%	1.8706%	false	85,576	25,676
4397	0.1719%	1.7028%	true	43,792	25,568
4457	1.5079%	0.2817%	true	50,771	26,136
4493	0.2623%	3.4839%	true	101,164	25,836
4513	0.5242%	1.3952%	true	56,060	25,708
4517	0.2961%	0.0735%	true	9,387	25,832
4597	0.1615%	1.5010%	true	56,241	25,552
4637	5.6579%	0.4922%	true	218,604	25,704
4657	1.0994%	0.9443%	true	67,466	25,760
4673	3.7523%	0.5686%	false	134,866	25,516
4733	0.9277%	0.9783%	true	55,318	25,712
4793	0.5844%	0.9697%	true	56,469	25,556
4813	1.5671%	2.1137%	true	113,390	25,784
4817	0.5109%	5.5784%	true	256,715	25,672
4877	1.9020%	0.3706%	true	78,673	25,684
4933	0.7224%	0.6621%	true	51,151	25,672
4937	1.0915%	0.6708%	true	64,869	25,816
4957	0.2418%	2.2691%	true	86,987	25,784
4973	0.3158%	0.2851%	true	25,922	25,776
4993	0.8801%	0.6917%	true	75,010	25,816
5077	2.5897%	0.0422%	true	94,844	25,824
5113	0.0687%	1.3861%	true	64,093	25,812
5153	0.8448%	1.3341%	true	82,537	25,572

Table 7. Experimental results on the LTZ hash function for $p \equiv 3 \pmod 4$.

	#(searched nodes)/p^3		The first	Time	Max RSS
p	first	second	is good?	(seconds)	(kB)
3307	0.8338%	1.1585%	true	21,528	25,804
3323	0.9915%	1.2879%	true	25,075	25,680
3343	0.7798%	0.4329%	true	14,282	26,116
3347	2.7564%	1.7676%	true	55,573	26,132
3407	0.4291%	2.0071%	true	34,567	25,732
3463	1.3460%	2.4295%	true	48,152	25,728
3467	1.1295%	1.1913%	false	28,234	25,732
3527	0.2397%	6.2534%	true	93,686	25,560
3547	0.7262%	3.5894%	true	55,953	25,836
3583	0.2578%	1.7910%	true	35,696	25,844
3607	0.6432%	2.0064%	true	39,979	26,000
3623	0.8533%	2.4450%	true	51,683	25,836
3643	0.0086%	1.3403%	true	22,294	25,724
3727	4.1225%	5.0300%	false	154,571	26,152
3767	1.0446%	0.8496%	true	36,550	25,848
3803	4.6619%	1.3121%	false	111,046	25,552
3823	0.9727%	1.7493%	true	47,020	25,676
3847	0.0510%	0.1876%	true	4,308	25,708
3863	2.5466%	1.5177%	true	72,808	25,856
3907	4.2112%	1.7867%	true	101,110	26,024
3923	1.7853%	1.1123%	true	60,548	25,628
3943	0.2812%	1.7321%	false	45,623	25,796
3947	4.8508%	0.8773%	false	124,157	25,744
3967	2.4865%	1.6090%	true	92,877	25,716
4003	2.7134%	1.3143%	true	75,936	25,776
4007	0.0630%	0.5205%	true	13,859	25,704
4027	1.5039%	0.0238%	true	33,873	25,988
4127	4.4480%	0.3602%	false	104,673	25,780
4243	0.5870%	1.6822%	false	50,468	25,808
4283	0.0217%	0.3162%	false	7,839	26,148
4327	1.6184%	4.5724%	true	156,197	25,832
4363	0.2152%	0.5513%	true	17,931	25,836
4423	2.0901%	1.9280%	true	107,997	25,704
4447	0.3790%	0.4015%	true	20,676	25,564
4463	3.0404%	0.7268%	false	104,570	25,680
4483	2.0335%	1.1217%	false	82,928	25,572
4507	1.1766%	0.0272%	false	33,235	26,076

Table 8. Experimental results on the LTZ hash function for $p \equiv 3 \pmod 4$.

p	#(searched nodes)/p^3 first	#(searched nodes)/p^3 second	The first is good?	Time (seconds)	Max RSS (kB)
4523	1.7553%	3.2035%	false	130,494	26,128
4547	0.7057%	6.1657%	false	189,359	25,888
4567	0.2397%	2.7024%	true	89,306	25,820
4583	0.7439%	9.5718%	true	313,677	25,752
4603	1.5322%	0.5375%	false	58,156	25,820
4643	0.1309%	0.3284%	true	13,471	25,728
4663	0.5830%	5.1941%	true	230,453	25,724
4703	1.0901%	1.4241%	false	83,571	26,100
4723	5.3840%	1.5733%	true	219,436	25,712
4783	3.4810%	0.6324%	true	145,468	25,556
4787	0.7933%	0.4165%	true	39,146	25,812
4903	3.2515%	0.8067%	true	153,387	25,828
4943	0.7374%	0.8937%	false	63,786	26,168
4967	0.6785%	0.7733%	true	59,894	25,564
4987	0.4022%	1.5804%	true	74,001	25,844
5003	1.6073%	1.2935%	true	109,181	25,568
5023	0.7718%	1.0298%	true	73,707	25,720
5087	1.7642%	0.6013%	true	114,135	25,684
5107	0.9886%	1.0880%	true	81,388	25,832
5147	9.6246%	0.1535%	false	346,364	25,848