Toward zkSNARK-assisted Isogeny-based Cryptography

Yi-Fu Lai and Luciano Maino

[1] Shanghai Jiao Tong University, Shanghai, China
yifu.lai@sjtu.edu.cn
[2] University of Birmingham, Birmingham, UK
l.maino@bham.ac.uk

Abstract. Zero-knowledge proofs are a fundamental building block of modern privacy-preserving systems. In isogeny-based cryptography, existing zero-knowledge proof constructions are either limited to chains of small-degree isogenies or are quite inefficient. As a result, many relations used in recent cryptosystems lack support in generic proof systems.
In this work, we take a step toward making zkSNARKs practically usable for a broader set of isogeny relations beyond the classical isogeny path knowledge language. Leveraging optimized Vélu-style formulas, we provide an efficient R1CS encoding for 3^m- and 4^n-isogenies, along with their masked evaluations. We also present an R1CS for non-smooth isogenies of special degree $q(2^e - q)$, where q is an odd integer, together with their evaluation. This latter encoding is based on the efficient formulas for $(2, 2)$-isogenies in the theta model.
Finally, we demonstrate several concrete applications of our tools. We present a compiler that removes the "one-more" evaluation assumption in the signature based on DeuringVRF. We also discuss how to eliminate the hint-based assumption in SQISign and explain how to construct a key-validation mechanism for recent public-key encryption designs, such as POKÉ on the concept level. We provide the experimental results with respect to the constraint numbers under various isogeny NIST-1 primes for reference. Under the setting, the proof sizes considered in this work are bounded by 400 KB by the default setting. We hope our results will inspire further advances in isogeny-based constructions.

1 Introduction

Zero-knowledge proofs of knowledge allow a prover to convince a verifier that they possess a secret (the witness) without revealing anything about that secret itself. Over the past few decades, these proofs have evolved from primarily theoretical constructs into fundamental cryptographic tools. Today, they are a core component of modern privacy-preserving systems and play an essential role in large-scale decentralized applications.

To be fully embedded in modern privacy-preserving systems, it is crucial that zero-knowledge proofs of knowledge are also quantum-resistant. Significant progress on this aspect has been made for lattice-based languages [3, 45, 73, 21, 7,

19, 20, 4, 23, 17]. While this provides a partial answer to the problem of having quantum-resistant solutions, it is widely recognized that relying on a single genre of assumptions is undesirable, emphasizing the value of maintaining "diversity" across security assumptions [2]. Hence, there is a need to explore alternative post-quantum solutions.

Isogeny-based cryptography offers an attractive and structurally different foundation, and is often highlighted for its compact public data [25, 39, 34, 80, 10, 68]. However, the "ecosystem" of advanced isogeny-based constructions is currently uneven. On the one hand, many advanced primitives [18, 41, 67, 16, 62, 42, 56] are built around isogeny group actions, where efficient and sufficiently quantum-safe instantiations remain an active area of research [38, 27, 5, 33] due to the subexponential attack [65, 66]. On the other hand, outside the group-action (oriented-isogeny) setting, expressive zero-knowledge support for isogeny-based languages is comparatively sparse, which limits the transfer of recent advances in isogeny constructions into applications that crucially rely on zero-knowledge proofs.

A major reason is that prior proofs of knowledge for isogeny relations are predominantly based on Σ-protocols [37, 9, 75, 34, 80, 10]. While conceptually simple, this approach typically faces a trade-off between efficiency and expressiveness. In particular, many constructions suffer from a small challenge space, requiring the need of several repetitions to amplify soundness and leading to significant performance overhead. Moreover, these techniques often support only restricted classes of relations, limiting their applicability in more complex protocols. This motivates the systematic study of zkSNARK-friendly encodings of isogeny relations.

The first attempt of applying zkSNARK to the isogeny-based construction was proposed in [26] using the classical sumcheck protocol [72] and PCP [6] to prove the knowledge of the "isogeny path". However, the straightforward encoding considered in their work makes the proof system inefficient. The first practical application result is due to Cong, Lai, and Levin [28], who suggested a more efficient encoding of isogeny path relations amenable to modern proof systems. Then, by using Aurora [14] on input the encoding, it demonstrates the comparable performance in both runtime and the proof size for the same language considered in [9] built from the sigma protocol. Follow-up works [57, 70, 58] explore nicer formulas or encodings also for the isogeny path knowledge. This specific focus is deliberate: it reflects the common belief that a straightforward zkSNARK treatment of most isogeny tasks would be prohibitively inefficient due to the underlying arithmetic complexity. Indeed, it takes tens of thousands of finite-field multiplications to compute an isogeny using Vélu formula under the traditional SIDH setting [31]. Therefore, zkSNARK applications do not yet cover most relations that arise naturally in modern isogeny-based signatures and public-key encryption schemes. This includes the basic isogeny evaluation relation, masked evaluation relation, the high dimensional walk and its evaluation relation [47, 12, 79, 10, 80, 68, 8].

This brings us to the main question of this work:

Can we expand the applications of zkSNARK beyond the isogeny path knowledge? Alternatively, can we obtain a zkSNARK friendly circuit for the relations used in modern isogeny-based cryptography?

Contributions

In this paper, we explore the full applicability of zkSNARKs in the R1CS language to isogeny relations that go beyond path proofs, with the goal of expanding the expressiveness of isogeny languages supported by generic proof systems and narrowing the gap between existing isogeny-based constructions and the capabilities of modern zero-knowledge proof systems.

Our contributions can be gathered as follows.

1. First, we show the optimized Vélu formula, overlooked in prior works, is as efficient as the state-of-the-art in the isogeny path knowledge in the R1CS metrics. Moreover, the formula can be extended to support evaluations naturally.
2. We introduce a R1CS system based on Vélu-like formulae for the following relations, where $d = 4^n$ or 3^m, for some $n, m > 0$.

$$\mathcal{R}_d := \{(E, E'), \varphi \mid \varphi : E \to E', \deg(\varphi) = d\}$$

$$\mathcal{R}_d^{\mathsf{mask-eval}} := \{(E, P, E', P'), (\varphi, m) \mid ((E, E'), \varphi) \in \mathcal{R}_d, \varphi(P) = [m]P'\}$$

3. We derive a R1CS for the following relations.

$$\mathcal{R}_{<2^n} := \{(E, E'), \varphi \mid \varphi : E \to E', \deg(\varphi) = q(2^n - q),\ 2 \not| q\}$$

$$\mathcal{R}_q^{\mathsf{eval}} := \{(E, P, E', P'), \varphi \mid ((E, E'), \varphi) \in \mathcal{R}_{<2^n}, \deg(\varphi) = q, \varphi(P) = P'\}$$

   Such R1CSes are derived from the formulas described in [35, 64] to compute chains of $(2, 2)$-isogenies, leveraging the symmetries of the so-called *theta model*.
4. Building on our R1CS constructions, we show several applications. First, we present a compiler that removes "one-more" evaluation assumption in DeuringVRF [68]. The same method also applies to PRISM [8]. Second, our R1CS gadgets enable a key-validation mechanism for recent public-key encryption designs, such as [11], allowing both parties to use static public keys.

1.1 Full Motivation of Encoding Isogeny Relations

Advanced cryptographic designs and applications rarely rely on a primitive alone. They typically combine a basic cryptographic mechanisms, with specific security notions, with a proof layer that *certifies* the correctness or well-formedness of the object used by the protocol. For example, it is used in the group signature

framework [13], the Fischlin paradigm of blind signatures [46], non-interactive blind signatures [55], Naor-Yung CCA transform [81], or upgrading an MPC scheme against malicious adversary [54].

The classical way to obtain zero-knowledge proofs for non-oriented isogeny relations is through Σ-protocols, starting from the SIDH-style identification paradigm [60]. After more than a decade of development, two main families of isogeny Σ-protocol diagrams have emerged: SIDH-style diagrams [60, 37, 9, 75] and SQISign-style diagrams [39, 34]. Both families have led to highly non-trivial constructions. However, in these protocols the proof diagram is usually intertwined with the algebraic construction itself. Changing the statement often means changing the commitment structure, the response structure, and the zero-knowledge analysis, rather than simply changing the relation being proved.

This limitation appears differently in the two families. SQISign-style protocols are extremely effective for compact signatures, but their proof layer certifies a statement tailored to the SQISign setting, essentially an endomorphism-type relation attached to one public curve [39, 34]. This is not the same interface as a proof of an isogeny between two prescribed curves, possibly together with evaluations on prescribed points. Moreover, in the higher-dimensional setting, the zero-knowledge analysis will rely on oracle-based simulations [34]. This makes the approach tricky to transfer even to primitives that are considered easy, such as ring signatures [22]. The issue is not that the SQISign paradigm is weak; rather, it is highly optimized for a specific family of statements.

The SIDH-style approach faces a different obstacle: it lacks the *agility* that is particularly desirable during the post-quantum transition. For example, since the attacks on SIDH [24, 74, 85], isogeny-based cryptography has evolved rapidly, and several new PKEs now rely on relations with rather different algebraic shapes [47, 12, 79]. At the same time, adaptive attacks on static-keys deployments continue to make key validation mechanism an important mechanism [52, 51, 76, 71]. However, the proof systems available for these relations have not evolved at the same pace [37, 75]. From our perspective, this gap comes from three related issues.

1. *Inefficiency.* Many isogeny Σ-protocols have a small challenge space and therefore require many repetitions to amplify soundness, which leads to large proofs and costly running time. A recent study [58] shows that even the performance of the simplest isogeny path relation considered in [9] is in the order of ten seconds.
2. *Convoluted zero-knowledge analysis.* The zero-knowledge analysis is often intricate and relation-specific [37, 9, 75]. Improving efficiency by reducing repetitions or changing the response structure may introduce additional isogeny-flavored assumptions (i.e. ad-hoc assumptions) or simulations that must be justified separately [37, 34, 75].
3. *Non-modularity.* The resulting protocols are usually tailored to a specific relation. Moving from one to another typically requires new designs of commitments, challenges and responses. The masked-degree and torsion techniques

of [75], for instance, are much more specialized than the earlier SIDH proofs of [37].

Together, these features turn the design of isogeny-based zero-knowledge proofs into a relation-by-relation engineering effort, which does not scale with the rapidly diversifying landscape of isogeny-based cryptography after the SIDH attacks.

From our point of view, an alternative with good potential is to use zkSNARKs as the proof layer. Its appeal is not merely succinctness, but the possibility of providing the agility that is currently missing in isogeny-based cryptography in the proof layer. This perspective addresses the three issues above as follows.

1. *Efficiency.* Once an isogeny relation is found a zkSNARK-friendly encoding, zkSNARK is likely to provide succinct proofs and efficient verification. For the same isogeny-path relation its first practical zkSNARK encoding of [28] already improves over the Σ-protocol baseline of [9] by a factor of 3–9×, while keeping proof sizes comparable. The recent tailored encoding [58] further increases this gap to about 34–89× in proving and 42–102× in verification, with smaller proofs.
2. *Simpler zero-knowledge analysis.* The zero-knowledge and proof-of-knowledge interface is supplied by the underlying zkSNARK in a largely relation-independent way. Thus, the proof layer can be analyzed through the generic security of the zkSNARK, rather than through a fresh ad-hoc isogeny assumption or a new simulator tailored to each individual relation.
3. *Modularity through arithmetic gadgets.* Standard zkSNARK interfaces, such as R1CS or polynomial-based arithmetizations, naturally support modular design. If we are able to find efficient encodings for codomain computation, point evaluation, and higher-dimensional isogeny counterparts are available as separate arithmetic gadgets, they can be assembled with existing encodings from the literature for scalar multiplication, scalar masking, field arithmetic, and pairing checks. This would give a flexible proof interface for a broad range of relations arising in modern isogeny-based cryptography, without redesigning from scratch for each relation.

Beyond the three points above, zkSNARKs may also offer additional advantages for isogeny-based cryptography.

First, they provide a possible way to avoid oracle-based simulation issues in some settings. For schemes where the prover currently publishes an isogeny representation and the zero-knowledge proof relies on an oracle-based simulator, one may instead consider a variant in which the representation is replaced by a zkSNARK proof. If the relevant statement can be encoded efficiently, the simulator of the underlying zkSNARK could then be used to simulate the proof, rather than relying on an isogeny-specific oracle output. This may increase the representation cost significantly, but it suggests a possible route for applying such techniques in future work.

Second, zkSNARKs naturally allow parts of the public algebraic structure to be treated as witnesses. For example, the curve itself can be hidden inside the witness, which is difficult to achieve with the above isogeny Σ-protocol

approaches. This feature is useful for relations appearing in more advanced primitives. As a concrete example, in the non-interactive blind-signature framework of [55, Scheme 5], the algorithm Obtain takes a verification key as input; in an isogeny-based instantiation, such a verification key would typically contain a curve. Being able to prove statements where this curve is part of the witness would not by itself give a complete isogeny-based non-interactive blind signature, since additional cryptographic components are still missing. Nevertheless, it represents an important step toward supporting such constructions.

Why R1CS? We therefore focus on efficient encodings of these important missing pieces. We use R1CS as our main representation because it provides a standard and portable constraint interface for arithmetic relations. The representation can be adapted to zkSNARKs' interfaces in general for further optimizations. Indeed, R1CS is only a convenient baseline interface: R1CS constraints can be arithmetized as quadratic arithmetic programs [53, 82], as multilinear-extension identities suitable for sum-check protocols [87, 90], or embedded into more general constraint abstractions such as CCS, which captures R1CS, Plonkish, and AIR-style constraints [88].

Limitations and Envision. We also highlight that while having a proof would still not give many isogeny-based constructions for the primitives mentioned above, since additional cryptographic components are still missing. Nevertheless, it represents an important step toward supporting such constructions by removing a critical obstacle. Our goal is to make isogeny relations available to generic proof systems in a form that is efficient enough to be useful, while remaining modular enough to accommodate the requirements of existing isogeny-based constructions and the new relations may be developed in the future.

1.2 Technical Overview

Before introducing our idea in the one-dimensional setting, we recall the standard kernel-to-codomain computation using the Vélu formulas. For a ℓ^a-torsion point $P \in E$, a naïve algorithm to compute the image curve $E/\langle P \rangle$ proceeds iteratively. First compute $Q = [\ell^{a-1}]P$, which has order ℓ; use $\langle Q \rangle$ as the kernel of a ℓ-isogeny to obtain $(E/\langle Q \rangle, P_1)$, where P_1 denotes the evaluation of P via the isogeny. Repeating this "multiplication-and-evaluation" process and computing $[\ell^{a-2}]P_1$, taking the corresponding ℓ-isogeny, and so on gives $E/\langle P \rangle$ after a steps.

Crucially, the cost of applying Vélu's formulas to evaluate a ℓ-isogeny is not the same as the cost of scalar multiplication. The classical optimization, called the isogeny strategy, reduces the asymptotic work to $O(a \log a)$ [60]. However, the concrete expense remains high, it still requires tens of thousands of $\mathbb{F}_p$ multiplications across all parameter sets [60, 31, 30]. Hence, verifying the isogeny knowledge through these computations directly using zkSNARK is impractical and avoided in the prior works [28, 57, 70].

Our idea here is surprisingly simple. The key observation is that we do not have to prove the full computation to ensure the isogeny knowledge. Instead, we refine the computation by proving the "Vélu chain" only. For the isogeny chain $E \to E_1 \to ... \to E_a$, we only prove for the Vélu computation for each $E_i \to E_{i+1}$ with respect to an ℓ-torsion point Q_i only. To prove the image curve computation, this is not sufficient. To avoid the malformed cases, we also have to prove the order of Q_i. The first idea can be illustrated as below.

$$\begin{matrix} E_0 \\ Q_0 \end{matrix} \xrightarrow[\substack{\ker(\phi_1)=\langle Q_0\rangle \\ \mathsf{ord}(Q_0)=\ell}]{\phi_1} \begin{matrix} E_1 \\ Q_1 \end{matrix} \xrightarrow[\substack{\ker(\phi_2)=\langle Q_1\rangle \\ \mathsf{ord}(Q_1)=\ell}]{\phi_2} \begin{matrix} E_2 \\ Q_2 \end{matrix} \xrightarrow[\substack{\ker(\phi_3)=\langle Q_2\rangle \\ \mathsf{ord}(Q_2)=\ell}]{\phi_3} \cdots \xrightarrow[\substack{\ker(\phi_n)=\langle Q_{n-1}\rangle \\ \mathsf{ord}(Q_{n-1})=\ell}]{\phi_n} E_n$$

While this avoids proving the costly scalar multiplication computation in the isogeny strategy, a direct instantiation is still more expensive than [28]. We optimize arithmetic on the Kummer varieties [31] for the zkSNARK purpose. In the end surprisingly, the resulting constraints and variables are reduced by factors 2.20 and 1.16 times compared to [28, 57] using the optimized modular polynomial relations. It is as efficient as the state-of-the-art [69] using the radical isogeny formula. Moreover, due to the existing Vélu pipeline, our design can support the isogeny evaluation easily by proving the evaluation $P_i = \phi_i(P_{i-1})$ consecutively where $\ker(\phi_i) = \langle Q \rangle$. Hence, by leveraging existing encodings for scalar multiplication and isogeny evaluation, one can readily extend them to support the masking relation $P' = [m]\phi(P)$. Moreover, we propose several optimizations that reduce the constraint cost of this masking relation by about 50%, and we also address the dual-masking relation $P' = [m]\phi(P) \ \wedge \ Q' = [m^{-1}]\phi(Q)$, which is used in [12, 11, 63].

Same as other approaches explored in the literature, encoding 1-dimensional relation is only possible if the prime ℓ is small, which is a natural limitation from the 1-dimensional formulas. In many cryptographic applications, however, we have to deal with isogenies ϕ of non-smooth degree, but still of a special form, *i.e.*, $\deg(\phi) = q(2^n - q)$ for some odd q (see, for instance, [12, 79, 8, 11, 68]). Such isogenies can be efficiently embedded into a two-dimensional isogeny Φ via Kani's Lemma [61].

The Φ itself is special: it is a $(2^n, 2^n)$-*isogeny*, *i.e.*, $\Phi = \Phi_n \circ \ldots \circ \Phi_1$, where Φ_i is a $(2, 2)$-isogeny. Concretely, this means that it can be efficiently computed using the so-called *theta model*, popularized in cryptography by [35]. The theta model enjoys several appealing features, such as a stricking internal symmetry that makes it amenable to be encoded in a R1CS.

In practice, let $\Phi_i : X \to X'$ be a $(2, 2)$-isogeny. The domain X and codomain X' can be represented by tuples of scalars, $(a : b : c : d)$ and $(a' : b' : c' : d')$ respectively. A point on X is represented as $(x_0 : x_1 : x_2 : x_3)$. In particular, we can compute $\Phi(x_0 : x_1 : x_2 : x_3)$ as follows:

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \overset{\mathcal{S}}{\longmapsto} \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} \overset{\mathcal{H}}{\longmapsto} \begin{pmatrix} X'_0 \\ X'_1 \\ X'_2 \\ X'_3 \end{pmatrix} \overset{\mathcal{S}ca\ell}{\longmapsto} \begin{pmatrix} \tilde{y}_0 \\ \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{pmatrix} \overset{\mathcal{H}}{\longmapsto} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

where

- $\mathcal{S}(x, y, z, t) := (x^2, y^2, z^2, t^2)$.
- $\mathcal{H}(x, y, z, t) := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix}$.
- $\mathcal{S}ca\ell(x, y, z, t) = (x/\tilde{a}, y/\tilde{b}, z/\tilde{c}, t/\tilde{d})$ and $(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}) = \mathcal{H}(a', b', c', d')$.

We encode such a formula above in a R1CS and obtain a proof for $\Phi = \Phi_n \circ \ldots \circ \Phi_1$, which even allows us to prove the evaluation of ϕ.

In contrast, extra care must be devoted to the isogenies Φ_1 and Φ_n, called the *gluing* and *splitting* steps respectively. This is because Φ is an isogeny between elliptic products, *i.e.*, $\Phi : E_1 \times E_2 \to E_3 \times E_4$. We combine the formula above with conversion formulas between Montgomery curves to provide a R1CS for Φ_1 and Φ_n. A natural approach is to encode the optimized gluing formulas of [35, 43]. However, these formulas are not well suited for R1CS encoding: the numerous branches in the underlying algorithm introduce substantial constraint overhead, which outweighs the savings suggested by the optimized multiplication count. Rather, our key observation is that there exists a *unique* symplectic transformation that allows us to endow $E_i \times E_{i+1}$, for $i = 1, 3$, with the correct theta structure. The uniqueness of such a transformation has already been observed in the literature but never been explicitly exploited.

Equipped with these R1CS gadgets, we then show a few applications to isogeny-based cryptography. We begin by showing how a simple compiler can be used to remove the *one-more evaluation* assumption in DeuringVRF, which is a verifiable pseudorandom function as well as a signature scheme. At a high level, DeuringVRF aims to prove knowledge of an isogeny ϕ_x whose kernel is generated by $\langle P + [x]Q \rangle$, where (P, Q) is a public prime-order torsion basis and x encodes the message. To realize this proof, the prover in fact publishes a kernel for a *two-dimensional* isogeny Φ that contains ϕ_x as one of its components. This, however, reveals substantially more information than what is needed to verify knowledge of ϕ_x with kernel $\langle P + [x]Q \rangle$. In particular, it enables the verifier to evaluate Φ on *arbitrary* torsion points, and thus to obtain evaluations for any component isogeny embedded in Φ. The one-more evaluation assumption therefore informally posits that such extra information about ϕ_x (for a given x) does not help when evaluating a distinct $\phi_{x'}$. Our simple compiler addresses this mismatch directly by proving only the *minimal* statement: the prover knows a two-dimensional isogeny with a component whose kernel is exactly $\langle P + [x]Q \rangle$. A similar idea applies to PRISM.

The other application is more straightforward. We can use the proofs developed in this work to validate the keys for recent public key encryption schemes such as POKE [11], which can prevent adaptive attacks [52, 71] and enables both parties to use static keys, a countermeasure used in the prior isogeny-based PKEs [37, 75].

Roadmap. We provide the essential background in Section 2. We present our R1CS constructions for the one-dimensional setting in Section 3, and our encodings for the two-dimensional setting in Section 5. We then describe cryptographic

applications in Section 6. Finally, Section A collects size and runtime figures for three distinct primes used in the literature, which we use as reference points for our performance estimates.

2 Preliminaries

In this section, we recall some background knowledge that will be useful in the rest of the paper. In Section 2.1, we summarize some fundamental facts about the mathematics behind isogeny-based cryptography. In Section 2.2, we recall the definition of Zero-knowledge Succinct Non-interactive Arguments of Knowledge (zkSnark). Finally, in Section 2.3, we recollect the notion of Rank-1 Constraint System (R1CS).

2.1 Abelian varieties and isogenies

In this section, we recall some background knowledge about abelian varieties and isogenies. Throughout the paper, we will assume that we are working over a finite field $\mathbb{F}_{p^2}$, where $p = 3 \pmod 4$. This restriction simplifies our narrative and is compatible with the applications we will describe in Section 6.

While we will recall some of the fundamental facts about elliptic curves and their isogenies, we assume some familiarity, referring the reader to [36, 89] for more details. For the sake of being self-contained, we will also recall some computational aspects about principally polarized abelian varieties and theta structures, specializing to the case of principally polarized abelian surfaces. Some references that we will cite throughout the paper include [35, 64, 43].

2.1.1 Elliptic curves and their isogenies In the rest of the paper, whenever we refer to an elliptic curve E, the reader should assume E is supersingular. Elliptic curves can be represented in several ways. In general, an elliptic curve E is represented via its *Weierstrass equation*, *i.e.*, $E: \ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_{p^2}$. We denote the *point at infinity* by $\mathcal{O}_E$. Arguably, the most common elliptic curve representation in isogeny-based cryptography is the *Montgomery form*. An elliptic curve E is expressed in the Montgomery form if its equation given by the equation $E : By^2 = x^3 + Ax^2 + x$, where $A, B \in \mathbb{F}_{p^2}$; we will say that A is its *Montgomery invariant*. The j-invariant of E is equal to $\frac{256(A^2-3)^3}{A^2-4}$. Sometimes, it will be useful to consider projective coefficients for the Montgomery form. In this case, an elliptic curve is given by the equation $E : By^2 = Cx^3 + Ax^2 + Cx$, where $(A : B : C) \in \mathbb{P}^2(\mathbb{F}_{p^2})$.

Curves in the Montgomery form enjoy several computational advantages; we refer the reader to [32] for a thorough discussion of Montgomery arithmetic. One of the most interesting properties is the fact that we can work on the *Kummer line* of a Montgomery curve E by simply dropping its y-coordinates. To be more precise, given a (projective) point $P = (x : y : z) \in E$, we can simply consider the xz-coordinates $(x : z)$ and work on the Kummer line.

An isogeny $\varphi : E_1 \to E_2$ between elliptic curves is a morphism such that $\varphi(\mathcal{O}_{E_1}) = \mathcal{O}_{E_2}$. A separable isogeny $\varphi : E_1 \to E_2$ is uniquely defined by its kernel $\ker(\varphi)$ and can be efficiently computed (from its kernel) as long as $\ker(\varphi)$ has smooth order and is contained in a small extension field of the field of definition of E; the degree of φ, denoted by $\deg(\varphi)$, coincides with the cardinality of $\ker(\varphi)$. The standard way to compute isogenies is via the so-called *Vélu's formulas*.

Theorem 1. *Let C be a finite subgroup of the elliptic curve*

$$E: \; y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

over K. Fix a partition $C = \{\mathcal{O}_E\} \cup C_2 \cup C^+ \cup C^-$, where C_2 are the order-2 points of C, and C^+ and C^- are such that for any $P \in C^+$ it holds that $-P \in C^-$. Write $S = C^+ \cup C_2$, and for $Q \in S$ define

$$g_Q^x = 3x(Q)^2 + 2a_2x(Q) + a_4 - a_1y(Q), \qquad g_Q^y = -2y(Q) - a_1x(Q) - a_3,$$

$$u_Q = (g_Q^y)^2, \qquad v_Q = \begin{cases} g_Q^x, & \text{if } 2Q = \mathcal{O}_E, \\ 2g_Q^x - a_1 g_Q^y, & \text{else,} \end{cases}$$

$$v = \sum_{Q \in S} v_Q, \qquad w = \sum_{Q \in S} \big(u_Q + x(Q)v_Q\big), \text{ and}$$

$$A_1 = a_1, \; A_2 = a_2, \; A_3 = a_3, \; A_4 = a_4 - 5v, \; A_6 = a_6 - (a_1^2 + 4a_2)v - 7w.$$

Then the separable isogeny φ with domain E and kernel C has codomain $E' = E/C$ with Weierstrass equation $E' : \; y^2 + A_1xy + A_3y = x^3 + A_2x^2 + A_4x + A_6$ over $\bar{K}$. Furthermore, for $P \in E$ we can compute the image of P as

$$u(\varphi(P)) = u(P) + \sum_{Q \in C \setminus \{\mathcal{O}_E\}} \big(u(P+Q) - u(Q)\big), \qquad u \in \{x, y\},$$

In the rest of the paper, we will also use *division polynomials*. Let E be an elliptic curve and let ψ_n denote the n-th division polynomial. For any point $P = (x(P), y(P)) \in E$ with $P \neq \mathcal{O}_E$, the vanishing of ψ_n characterizes n-torsion: $\psi_n(P) = 0$ iff $[n]P = \mathcal{O}_E$. Moreover, the size of ψ_n grows quadratically in n: viewed as a polynomial in the x-coordinate (after removing the trivial $2y$ factor when n is even), its degree is $\deg_x(\psi_n) = (n^2 - 1)/2$ for odd n, and $\deg_x(\psi_n) = (n^2 - 4)/2$ for even n [89]. Looking ahead, we will use the division polynomial relation to identify the order of a point.

2.1.2 Principally polarized abelian surfaces and their isogenies The higher-dimensional generalization of elliptic curves is *principally polarized abelian varieties*. An abelian variety of dimension two is called *abelian surface*. Up to isomorphisms (over the algebraic closure of their field of definition), principally polarized abelian surfaces (PPASes) come into two flavors: either Jacobians of genus-2 hyperelliptic curves or products of two elliptic curves. In what follows, we will only consider superspecial PPASes.

An isogeny between abelian surfaces $\Phi : A \to B$ is a surjective morphism with finite kernel. However, when A and B are principally polarized, the isogeny Φ needs to preserve the principal polarization. In practice, this means that there exists an integer $N > 0$ such that $\ker(\Phi) = \langle T_1, T_2 \rangle \simeq \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$, $e_n(T_1, T_2) = 1$ and there exists no other subgroup $K \supsetneq \ker(\Phi)$ of $A[N]$ such that, for all $S_1, S_2 \in K$: $e_n(S_1, S_2) = 1$. We will call such an isogeny an (N, N)-isogeny.

An efficient way to represent PPASes is to consider the so-called *theta model*. Before introducing the theta model, we have to introduce the concept of *symplectic decomposition*. Let A be a PPAS. We say that $\langle S_1, S_2 \rangle \oplus \langle T_1, T_2 \rangle$ is a symplectic decomposition of $A[N]$ if

1. (S_1, S_2, T_1, T_2) is a basis of $A[N]$;
2. $e_N(S_1, S_2) = e_N(S_1, S_2) = e_N(T_1, S_2) = e_N(T_2, S_1) = 1$;
3. $e_N(S_1, T_1) = e_N(S_2, T_2) = \zeta$, where ζ is a Nth-primitive root of unity.

A symplectic decomposition of $A[4]$ induces a morphism $\theta : A/\langle -1 \rangle \hookrightarrow \mathbb{P}^3$. In other words, it gives a description of the *Kummer surface* of A. When working with such an embedding, we will say that A is represented in the theta model. Given $P \in A$, we will say that $\theta(P) = (x : y : z : t)$ are the *theta coordinates* of P. Henceforth, we will omit the θ when talking about the theta coordinates of P; we will simply write $P = (x : y : z : t)$.

The theta model of A is uniquely determined by $\mathcal{O}_A = (a : b : c : d) \in \mathbb{P}^3(\mathbb{F}_{p^2})$, where $\mathcal{O}_A$ denotes the identity of A. The symplectic decomposition $\langle S_1, S_2 \rangle \oplus \langle T_1, T_2 \rangle$ of $A[4]$ induces the theta structure in the following sense. Assume that we are using the same ordering of [35]; this will be our choice for the rest of the paper. Define $S_i' = [2]S_i$ and $T_i' = [2]T_i$. If $P = (x : y : z : t)$, we have $S_1' = (b : a : d : c)$ and $P + S_1' = (y : x : t : z)$, $S_2' = (c : d : a : b)$ and $P + S_2' = (z : t : x : y)$, $T_1' = (a : -b : c : -d)$ and $P + T_1' = (x : -y : z : -t)$, $T_2' = (a : b : -c : -d)$ and $P + T_2' = (x : y : -z : -t)$.

Another main advantage of the theta model is that it gives relations between $(2, 2)$-isogenous PPASes as explained in [35, §2.2]; such relations are called the *duplication formula*.[3] Let A be a PPAS represented in the theta model induced by the symplectic decomposition $\langle S_1, S_2 \rangle \oplus \langle T_1, T_2 \rangle$ of $A[4]$. Now, let us consider the isogeny $\Phi : A \to B$ such that $\ker(\Phi) = \langle T_1, T_2 \rangle$. The PPAS B can be expressed in the theta model, with a decomposition compatible with $\langle S_1, S_2 \rangle \oplus \langle T_1, T_2 \rangle$. Let $P, Q \in A$ and define

- $(x_{P+Q} : y_{P+Q} : z_{P+Q} : t_{P+Q})$ to be the theta coordinates of $P + Q$;
- $(x_{P-Q} : y_{P-Q} : z_{P-Q} : t_{P-Q})$ to be the theta coordinates of $P - Q$;
- $(x_{\Phi(P)} : y_{\Phi(P)} : z_{\Phi(P)} : t_{\Phi(P)})$ to be the theta coordinates of $\Phi(P)$;
- $(x_{\Phi(Q)} : y_{\Phi(Q)} : z_{\Phi(Q)} : t_{\Phi(Q)})$ to be the theta coordinates of $\Phi(Q)$.

Then, we have that

$$\begin{aligned} \mathcal{H}\left((x_{P+Q} : y_{P+Q} : z_{P+Q} : t_{P+Q}) \star (x_{P-Q} : y_{P-Q} : z_{P-Q} : t_{P-Q})\right) = \\ \mathcal{H}(x_{\Phi(P)} : y_{\Phi(P)} : z_{\Phi(P)} : t_{\Phi(P)}) \star \mathcal{H}(x_{\Phi(Q)} : y_{\Phi(Q)} : z_{\Phi(Q)} : t_{\Phi(Q)}), \end{aligned} \tag{1}$$

[3] Such a formula is a consequence of the isogeny theorem [78, Theorem 4, p. 302] (see also [84, Theorem 3.6.4]) and duplication formula [78, Equation A, p. 332].

where $\star$ denotes the component-wise multiplication and

$$\mathcal{H}(x,y,z,t) := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix}.$$

2.2 Zero-knowledge Succinct Non-interactive Arguments of Knowledge

Let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a relation and let $\mathcal{L}(\mathcal{R}) := \{x : \exists w \text{ s.t. } (x,w) \in \mathcal{R}\}$ be its language. In the random oracle model, a *zero-knowledge succinct non-interactive argument of knowledge* for $\mathcal{R}$ is a pair $\Pi = (\mathsf{Prove}, \mathsf{Ver})$ of PPT algorithms with oracle access to a random oracle O, satisfying:

- Completeness. For every $(x,w) \in \mathcal{R}$,

$$\Pr\left[\mathsf{Ver}^{\mathsf{O}}(x,\pi) = 1 \;\middle|\; \pi \leftarrow \mathsf{Prove}^{\mathsf{O}}(x,w)\right] = 1.$$

- Argument of knowledge. There exist a negligible function ϵ and a PPT extractor $\mathcal{E}$ such that for every PPT prover $\tilde{P}$ and every instance x, the inequality holds $\epsilon(\lambda,|x|) \geq \mathsf{Adv}^{\mathsf{AoK}}_{\Pi}(\tilde{P})$ where

$$\mathsf{Adv}^{\mathsf{AoK}}_{\Pi}(\tilde{P}) := \left| \Pr_{w \leftarrow \mathcal{E}^{\tilde{P}}(x,1^\lambda)}[(x,w) \in \mathcal{R}] - \Pr_{\pi \leftarrow \tilde{P}^{\mathsf{O}}(x)}[\mathsf{Ver}^{\mathsf{O}}(x,\pi) = 1] \right|.$$

- Zero knowledge. There exists a PPT simulator $\mathcal{S}$ and a negligible function ϵ such that for every $(x,w) \in \mathcal{R}$ and every PPT distinguisher $\mathcal{D}$, the inequality holds $\epsilon(\lambda,|x|) \geq \mathsf{Adv}^{\mathsf{ZK}}_{\Pi}(\mathcal{D})$ where

$$\mathsf{Adv}^{\mathsf{ZK}}_{\Pi}(\mathcal{D}) := \left| \Pr_{\pi \leftarrow \mathsf{Prove}^{\mathsf{O}}(x,w)}[\mathcal{D}^{\mathsf{O}}(x,\pi) = 1] - \Pr_{\pi \leftarrow \mathcal{S}^{\mathsf{O}}(x)}[\mathcal{D}^{\mathsf{O}}(x,\pi) = 1] \right|.$$

- Succinctness: A proof system $(\mathsf{Prove}, \mathsf{Ver})$ for a relation $\mathcal{R}$ is *succinct* if, for any $(x,w) \in \mathcal{R}$ and corresponding proof $\pi \leftarrow \mathsf{Prove}^{\mathsf{O}}(x,w)$, the proof size is polylogarithmic in the relevant input size. In particular, $|\pi| = \mathsf{poly}(\lambda, |x|, \log|w|)$.

2.3 Rank-1 Constraint Systems

We recall the definition of rank-1 constraint systems (R1CS), a standard arithmetic representation used as the input language of many general-purpose zk-SNARKs. An R1CS is parameterized by $n, m \in \mathbb{N}$ and a prime power q, and consists of instance–witness pairs $((A,B,C,v),w)$, where $A,B,C \in \mathbb{F}_q^{m \times (n+1)}$ and v,w are vectors over $\mathbb{F}_q$. Let $z := (1,v,w) \in \mathbb{F}_q^{n+1}$, where the leading 1 accounts for constant terms. The witness w satisfies the instance (A,B,C,v) if $Az \circ Bz = Cz$, where $\circ$ denotes the coordinate-wise (Hadamard) product. Conceptually, A,B,C encode m quadratic constraints on the variables in z: the vector v collects the public (auxiliary) input, while w contains the secret input

Table 1: Summary of the $\mathbb{F}_p$-constraint costs for one-dimensional isogeny tasks.

	4^ℓ-Path	Eval	3^ℓ-Path	Eval	N-torsion mask
[28] ACNS'23	22ℓ	–	–	–	–
[57] Crypto'25	14ℓ	–	11ℓ	–	–
[69] ASIACRYPT'25	10ℓ	–	–	–	–
Ours	10ℓ	15ℓ	10ℓ	12ℓ	$11\lceil \log_2 N \rceil$

as well as intermediate variables arising from the computation. As is standard in isogeny-based cryptography, we work over the quadratic extension field $\mathbb{F}_{p^2}$ for a prime p. Following [28], we account for the cost of expressing $\mathbb{F}_{p^2}$-constraints over the base field $\mathbb{F}_p$: using Karatsuba technique, one $\mathbb{F}_{p^2}$ constraint can be implemented with 3 constraints over $\mathbb{F}_p$, whereas an $\mathbb{F}_{p^2}$ squaring constraint can be converted to only 2 base-field constraint. We use this conversion throughout our constraint-count calculations. Since encodings of standard elliptic-curve relations—such as point scalar multiplication—are well established [15, 44, 29, 59], we do not repeat them here. Throughout, we work over $\mathbb{F}_{p^2}$ and model the Montgomery-curve group law by assuming that a point addition and a point doubling cost 5 and 8 $\mathbb{F}_p$-constraints for the x-coordinate, respectively, and take extra 3 for verifying the y-coordinate. In contrast, most existing pairing encodings are optimized for curves used in practice (e.g., BLS12) rather than Montgomery curves. We therefore estimate pairing costs by following the same step-by-step accounting approach as [83]. Concretely, for an N-torsion pairing where $E[N]$ is accessible over $\mathbb{F}_{p^2}$, we estimate the Miller-loop constraint cost over $\mathbb{F}_p$ as $19 \cdot (L-1) + 14 \cdot (n-1)$, where $L := \lfloor \log_2 N \rfloor + 1$ is the bitlength of N and n is the Hamming weight of its binary expansion.

3 Dimension one Relations

This section presents our optimized R1CS encoding for 1-dimensional isogeny relations starting from the isogeny path relation and to the evaluation relations. Different from the formulas considered in the literature [28, 57, 69], we show that the classical Vélu formula can be optimized as efficient as for proving the isogeny path relation. In this way, importantly, our formulation can be extended to support evaluation relation easily. Our result is summarized in Table 1.

3.1 4-Isogeny Relation

From [31], we have the following Kummer variety arithmetic (x,z-coordinate) for 4-isogeny computation. Let $E : By^2 = Cx^3 + Ax^2 + Cx$ be a Montgomery curve with projective coefficients $(A : C) \in \mathbb{P}^1$.

Let $\varphi : E \to E'$ be a 4-isogeny whose kernel is generated by a point P with x-coordinate $(X_4 : Z_4) \in \mathbb{P}^1$ on E.

Codomain Computation. $(A' : C') = \big(2(2X_4^4 - Z_4^4) : Z_4^4\big)$.

Evaluation. For $Q = (X : Z) \in \mathbb{P}^1$ on E, the x-only evaluation of φ is

$$(X' : Z') = \Big(X\left(2X_4Z_4Z - X(X_4^2 + Z_4^2)\right)(X_4X - Z_4Z)^2 : \\ Z\left(2X_4Z_4X - Z(X_4^2 + Z_4^2)\right)(Z_4X - X_4Z)^2\Big).$$

3.1.1 4-Isogeny Codomain Relation.

We begin by optimizing the codomain computation relation in this subsection.

Optimization for Codomain by Taking Z_4=1. We can take $Z_4 = 1$ to minimize the cost. In this case, C' is naturally 1. This reduces to 2 squaring constraints.

$$X_4 \cdot X_4 = s_1, \qquad s_1 \cdot s_1 = A'/4 + 1/2.$$

Concretely, the witness is $z = (1, X_4, s_1, A'/4 + 1/2)$ for the public matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Remark that we have $A'/4 + 1/2 = X_4^4$. The formulation not only verifies the curve but also assists in the verifications later.

Order-4 Test. While verifying the image curve seems cheap (two squarings), this efficiency rely on the implicit promise that the input point has order 4. Without a way to ensure the order, a malicious prover can feed an invalid point and pass the check trivially.

There are two natural ways to certify that P has order 4. The first is to compute $[2]P$ and verify that its y-coordinate is zero; on Montgomery curves, every rational 2-torsion point satisfies $y = 0$. The second is to test whether the 4-division polynomial vanishes at P.

We show that the division-polynomial test can be made cheaper and leave the first approach in Section 4. At first glance, the double-and-check method looks more economical because the 4-division polynomial has relatively high degree. Nevertheless, with careful algebraic simplifications, evaluating the division polynomial becomes practical and outperforms the $[2]P$-and-y=0 check. We give the details below.

Order-4 Test: 4-Division Polynomial Test.

Recall that for a point with abscissa X we have $\psi_4(X) = 0$ if and only if the point of order dividing 4. Concretely, in our case, $\psi(x) = 4y(x-1)(x+1)(x^4 + 2Ax^3 + 6x^2 + 2Ax + 1)$. As 2-torsion points vanish at y it suffices to show $(x-1)(x+1)(x^4 + 2Ax^3 + 6x^2 + 2Ax + 1)$ vanishes at P for the order 4 test.

We first compute the multiplication AX_4. Recall that X_4^4, X_4^2 have been computed in the image curve computation. Then, we show

$$
\begin{aligned}
& (2AX_4 + X_4^2 + 5)(X_4^4 - 1) = 4X_4^2 - 4, \quad \text{then} \qquad (2)\\
& 0 = (2AX_4 + X_4^2 + 5)(X_4^4 - 1) - (4X_4^2 - 4)\\
\iff \quad & 0 = (X_4^2 - 1)\big[(2AX_4 + X_4^2 + 5)(X_4^2 + 1) - 4\big]\\
\iff \quad & 0 = (X_4^2 - 1)\big(X_4^4 + 2AX_4^3 + 6X_4^2 + 2AX_4 + 1\big).
\end{aligned}
$$

Since Eq. (2) can be represented by one $\mathbb{F}_{p^2}$ constraint, together with the computation of AX_4 our optimized order 4 test using only 6 constraints over $\mathbb{F}_p$. Hence, to prove a 4-isogeny relation, it takes 10 constraints over $\mathbb{F}_p$.

3.1.2 4-Isogeny Evaluation Relation.

We show our optimization on the evaluation relation in this subsection. We reuse the same trick first by taking the z-coordinate to be 1 to simplify the complexity.

Evaluation by Taking Z_4, Z=1.

$$
(X' : Z') = \Big(X\left(2X_4 - X(X_4^2+1)\right)(X_4X-1)^2 : \big(2X_4X - (X_4^2+1)\big)(X - X_4)^2\Big).
$$

Since $X_4^2 = s_1$ have been computed, we have the optimized procedure could be as follows:

(Row 1)	$X_4 \cdot X$	$= m_1$
(Row 2)	$(m_1 - 1) \cdot (m_1 - 1)$	$= s_2$
(Row 3)	$X \cdot X$	$= s_3$
(Row 4)	$(-s_2 + 1 - s_3) \cdot s_2$	$= X'$
(Row 5)	$(2m_1 - s_1 - 1) \cdot (s_3 - 2m_1 + s_1)$	$= Z'$
(Row 6)	$X'/Z' \cdot Z'$	$= X'$

In fact we can use the following arithmetic trick to optimize it further by moving one multiplication to one squaring which now is 3 multiplication and 3 squaring.

(Row 1)	$(X + X_4) \cdot (X + X_4)$	$= s_2'$	$= (X + X_4)^2$
(Row 2)	$(X - X_4) \cdot (X - X_4)$	$= s_3'$	$= (X - X_4)^2$
(Row 3)	$(\frac{s_2' - s_3'}{4} - 1) \cdot (\frac{s_2' - s_3'}{4} - 1)$	$= s_2$	$= (X_4X - 1)^2$
(Row 4)	$(1 - s_2 - \frac{s_2' + s_3' - 2s_1}{2}) \cdot s_2$	$= X'$	
(Row 5)	$(\frac{s_2' - s_3'}{2} - s_1 - 1) \cdot s_3'$	$= Z'$	
(Row 6)	$X'/Z' \cdot Z'$	$= X'$	

3.2 3-Isogeny Relation.

Similarly, from [31], we have the following Kummer variety arithmetic (x,z-coordinate) for 3-isogeny computation. We also take the formula from [30] into account. While [30] is faster in general, the form adapted and optimized above is better for the proof system here. Let $\varphi : E \to E'$ be a 3-isogeny whose kernel is generated by a point P with x-coordinate $(X_3 : Z_3) \in \mathbb{P}^1$ on E.

Codomain Computation. Using the projective 3-division relation, the codomain curve coefficients are

$$(A' : C') = \big((AX_3Z_3 + 6(Z_3^2 - X_3^2))X_3 : CZ_3^3\big), \tag{3}$$
$$= \big(Z_3^4 + 18X_3^2Z_3^2 - 27X_3^4 : 4X_3^3Z_3^3\big). \tag{4}$$

Evaluation. For an arbitrary input point $Q = (X : Z) \in \mathbb{P}^1$ on E, the x-only evaluation of φ is given by (cf. Eq. (3) in the reference)

$$(X' : Z') = \Big(X\,(X_3X - Z_3Z)^2 : Z\,(Z_3X - X_3Z)^2\Big).$$

Codomain by Taking $C = Z_3 = 1$. Specializing the kernel coordinate to $Z_3 = 1$ (standard cost saving) gives

$$(A' : C') = \big((AX_3 + 6(1 - X_3^2))X_3 : 1\big) = \big(1 + 18X_3^2 - 27X_3^4 : 4X_3\big).$$

We choose the former one with the following procedure:

$$X_3 \cdot X_3 = s_1, \qquad (A - 6X_3) \cdot s_1 = A' - 6X_3.$$

We have the following constraint system for the witness $z = (1, A,\ X_3,\ s_1,\ A')$ with 5 $\mathbb{F}_p$-constraints. As R1CS matrices we may write

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -6 & 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -6 & 0 & 1 \end{pmatrix}.$$

Order-3 Test. Similarly, there are two approaches to certify the order of P. The first is to show $X_{[2]P} = X_3$. Since by setting $Z_3 = 1$, P is not the point at infinity, the equation implies P is of order 3. The other is to use the 3-division polynomial. The latter approach gives a better performance. The 3-division polynomial is $\psi_3(x) = 3x^4 + 4Ax^3 + 6x^2 - 1$. Since X_3^2 and $X_3^3 - AX_3^2$ have been computed, we can first compute X^4 from squaring X^2. Then, with one constraint $(X_3^3 - AX_3^2)X_3 = -2X_3^4 - 6X_3^2 + 1$ we can verify the order, which takes 5 constraints over $\mathbb{F}_p$. In summary, it takes 10 $\mathbb{F}_p$ constraints in total to prove the 3-isogeny relation.

3.2.1 3-Isogeny Evaluation.

Similarly we handle the evaluation first by taking $Z_3, Z = 1$. For Q affine with $x(Q) = X$, and kernel generator X_3, the projective formula simplifies to

$$X' = X\,(X_3X - 1)^2, \qquad Z' = (X - X_3)^2.$$

We enforce $x' = X'/Z'$, and thereby $x'Z' = X'$.

We use the same technique developed in our 4-isogeny evaluation optimization above by trading one multiplication for one extra square, which saves 1 constraint over $\mathbb{F}_p$. Hence, a 3-isogeny evaluation takes 12 constraints over $\mathbb{F}_p$ as follows.

$$\begin{aligned}
&\text{(Row 1)} && (X + X_3) \cdot (X + X_3) && = s_+ && = (X + X_3)^2,\\
&\text{(Row 2)} && (X - X_3) \cdot (X - X_3) && = Z' && = (X - X_3)^2,\\
&\text{(Row 3)} && \left(\tfrac{s_+ - Z'}{4} - 1\right) \cdot \left(\tfrac{s_+ - Z'}{4} - 1\right) && = s_1 && = (X_3 X - 1)^2,\\
&\text{(Row 4)} && X \cdot s_1 && = X',\\
&\text{(Row 5)} && x' \cdot Z' && = X'.
\end{aligned}$$

3.3 On Masking Torsion Points

While scalar-multiplication constraints are relatively well studied in the classical literature, our isogeny setting admits a few additional techniques reducing the cost significantly.

Doubling Constraints Reduction by Reordering. Consider a statement $(P, P') \in E \times E'$ with witness (m, ϕ) such that $P' = [m]\phi(P)$. A natural approach is to introduce an intermediate point $\tilde{P}'$ and prove $\tilde{P}' = \phi(P)$ together with $[m]\tilde{P}' = P'$. Instead, we recommend introducing the intermediate point on the *domain* curve and proving

$$\tilde{P} = [m]P \qquad \text{and} \qquad P' = \phi(\tilde{P}).$$

The advantage comes from the way scalar multiplication is encoded. In typical R1CS-style encodings, the scalar m is represented by bits $m = \sum_{i=0}^{\ell-1} m_i 2^i$, with $(m_0, \ldots, m_{\ell-1}) \in \{0,1\}^\ell$, and the relation $[m]Q$ is verified using a double-and-add recurrence. Since the constraint system cannot express branching natively, it effectively has to enforce a dummy recurrence of the form $Q_i = [2]Q_{i-1} + [m_i]Q$, and therefore pays for *both* doubling and conditional addition at each step.

By switching to $\tilde{P} = [m]P$ and $P' = \phi(\tilde{P})$, we can include the pre-doubled points $[2]P, [4]P, \ldots$ as part of the *statement*. Then the recurrence can be rewritten as

$$P_i = P_{i-1} + [m_i]\,[2^i]P, \qquad P_1 = [m_1]P,$$

which eliminates the need to prove the doubling relation inside the circuit. Concretely, this reduces the scalar-multiplication cost by around 50%.

Working over $\mathbb{F}_{p^2}$, we count constraints over $\mathbb{F}_p$ using the standard Karatsuba lifting. Montgomery-curve addition costs eight $\mathbb{F}_p$-constraints [59]; enforcing bitness of each m_i costs one $\mathbb{F}_p$-constraint via $m_i(m_i - 1) = 0$; and the conditional addition requires two additional $\mathbb{F}_p$-constraints. Consequently, the reordered $[N]$-scalar-multiplication check costs $(8 + 2 + 1)\lceil \log_2(N) \rceil = 11\lceil \log_2(N) \rceil$ $\mathbb{F}_p$-constraints.

Dual masking in POKE-style relations. A second trick applies to the dual masking relation appearing in POKE encryption. There, the statement consists of (P, Q, P', Q') with witness (ϕ, m) such that $[m]\phi(P) = P'$ and $[m^{-1}]\phi(Q) = Q'$. A naive encoding would represent both m and m^{-1} as bit strings and additionally prove that they are inverses modulo the subgroup order. This is inconvenient in an $\mathbb{F}_p$-constraint system, since the inverse relation is defined modulo the group order rather than over the base field.

We propose two ways to avoid this issue. The first applies when $\{P, Q\}$ is a basis of the relevant subgroup. In this case, one may verify $[m]\phi(P) = P'$ and $[w]\phi(Q) = Q'$ separately, and then use the pairing equation $e(P', Q') = e(\phi(P), \phi(Q))$ to enforce $mw = 1$ modulo the subgroup order. This avoids expressing the modular inverse relation directly inside the constraint system. A cleaner and more general approach is to eliminate the inverse scalar altogether. Since $[m^{-1}]\phi(Q) = Q'$ is equivalent to $\phi(Q) = [m]Q'$, we introduce an intermediate point $\tilde{Q}'$ on the codomain curve and prove $\tilde{Q}' = [m]Q'$ and $\phi(Q) = \tilde{Q}'$. Thus both scalar-multiplication checks use the same bit decomposition of m, and no inverse relation has to be represented inside the circuit. Moreover, the first relation can be handled as above by proving $\tilde{P} = [m]P$ and $P' = \phi(\tilde{P})$; similarly, the second scalar multiplication can use the pre-doubled points $[2^i]Q'$ as part of the statement.

4 Alternative Formulas

This sections presents some alternative formulas for encoding for reference, which are less efficient under our scope.

Order-4 Test: Doubling and Y-coordinate Test. We use the doubling arithmetic in [32]: the x-formula is

$$(X_{[2]P_4}/Z_{[2]P_4} : 1) = \Big((X_4^2 - 1)^2/4(X_4^4 + X_4) : 1\big).$$

Concretely, the witness is $z = (1, X_4, s_1, X_4^4, X_{[2]P_4}/Z_{[2]P_4})$ for the public matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 & 4 & 0 & 4 & 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & 0 & -2 & 1 & 1 \end{pmatrix}$$

To show $x^3 + Ax^2 + x = 0$, we consider the following representation.

Concretely, the witness is $z = (1, A, x, s_2)$ for the public matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

This approach takes 2 multiplications and 1 squaring over $\mathbb{F}_{p^2}$. It takes 8 constraints in total by lifting to $\mathbb{F}_p$, which takes more constraints than the one considered in the context.

5 Construction for Dimension two

In this section, we describe how to construct a R1CS for the following relation.[4]

$$\mathcal{R}_{<2^n} := \{(E_1, E_2), \varphi \mid \varphi : E_1 \to E_2,\ E_1 \not\cong E_2,\ \exists q \ s.t.\ \deg(\varphi) = q(2^n - q)\ 2 \not| q\}$$

Our construction relies on Kani's Lemma [61], popularized in isogeny-based cryptography via the SIDH attacks [24, 74, 85]. Indeed, an isogeny $\varphi : E_1 \to E_2$ of degree $q(2^n - q)$ can be represented via a $(2^n, 2^n)$-isogeny $\Phi : E_1 \times E_2 \to E_3 \times E_4$. To be more specific, the isogeny Φ can be represented via the isogeny matrix $\Phi = \begin{pmatrix} \varphi_1 & -\widehat{\varphi}_2 \\ \psi_1 & \widehat{\psi}_2 \end{pmatrix} : E_1 \times E_2 \to E_3 \times E_4$. where the isogenies φ_i and ψ_i satisfy $\varphi = \varphi_2 \circ \varphi_1 = \psi_2 \circ \psi_1$ and $\deg(\psi_1) = \deg(\varphi_2) = 2^n - \deg(\varphi_1) = 2^n - \deg(\psi_2) = 2^n - q$. In particular $\ker(\Phi) = \{([2^n - q]\widehat{\varphi}_1(P), [q]\varphi_2(P)) \mid P \in E_3[2^n]\}$.

The isogeny Φ can be seen as a chain of $(2,2)$-isogenies, *i.e.*, $\Phi = \Phi_n \circ \ldots \circ \Phi_1 : E_1 \times E_2 \to E_3 \times E_4$, where each Φ_i is a $(2,2)$-isogeny. The best way to compute Φ is to use the so-called *theta-model* described in [35]. Following the same reasoning as in [35, §4.1], we will assume that Φ has been computed utilizing the *four-torsion above the kernel*, namely, the subgroup generated by $([2^n - q]\widehat{\varphi}_1(P), [q]\varphi_2(P))$ and $([2^n - q]\widehat{\varphi}_1(Q), [q]\varphi_2(Q))$, where (P, Q) is a basis of $E_3[2^{n+2}]$. This choice will give us several advantages that we will make explicit in what follows.

The rest of the section is organized as follows. We first explain how to encode each single $(2,2)$-isogeny into a R1CS and then we describe how to deal with the isogeny Φ_1 and the Φ_n. Finally, we explain how to expand our construction to also evaluate isogenies; namely we give a R1CS for the relation

$$\mathcal{R}_q^{\mathsf{eval}} := \{(E, P, E', P'), \varphi \mid ((E, E'), \varphi) \in \mathcal{R}_{<2^n}, \deg(\varphi) = q, \varphi(P) = P'\},$$

when the 2^n-torsion of E is defined over $\mathbb{F}_{p^2}$

5.1 Encoding $(2,2)$-isogenies in a R1CS.

As explained in Section 2.1.2, principally polarized abelian surfaces (PPASes) can be represented by *theta-null points*, *i.e.*, a tuple $(a : b : c : d) \in \mathbb{P}^3(\mathbb{F}_{p^2})$. Following the approach in [64], which is derived from Equation (1), $(2,2)$-isogenies can be described combining the following operators.

- $\mathcal{S}(x, y, z, t) := (x^2, y^2, z^2, t^2)$,
- $\mathcal{H}(x, y, z, t) := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix}$,
- $\mathcal{T}_{s_1 s_2 s_3}(x, y, z, t) := \left(x, (-1)^{s_1}\sqrt{yx}, (-1)^{s_2}\sqrt{zx}, (-1)^{s_3}\sqrt{tx}\right)$, for $s_i \in \{0, 1\}$.

We observe that each $s_i \in \{0, 1\}$ will correspond to a valid isogeny. The observation is important. In particular, since our goal is to verify $(2,2)$-isogenies, we may rewrite $\mathcal{T}_{s_1 s_2 s_3}(x, y, z, t)$ as $\left(\sqrt{x}, (-1)^{s_1}\sqrt{y}, (-1)^{s_2}\sqrt{z}, (-1)^{s_3}\sqrt{t}\right)$ rather

[4] We remark that, in $\mathcal{R}_{<2^n}$, the degree q is not part of the statement. Given a statement (E_1, E_2), where E_1 and E_2 are not isomorphic, the (non-necessarily unique) witness $\varphi : E_1 \to E_2$ is an isogeny of degree $q(2^n - q)$ "for some" odd $q < 2^n$.

than encoding a specified square root relation with a multiplication by ± 1. Indeed, theta-null points are projective, and every possible choice for the square-roots leads us to a valid $(2,2)$-isogenous theta-null point.

Given a theta-null point $(a : b : c : d)$, a $(2,2)$-isogenous theta-null point is computed as

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \overset{\mathcal{S}}{\longmapsto} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \overset{\mathcal{H}}{\longmapsto} \begin{pmatrix} A' \\ B' \\ C' \\ D' \end{pmatrix} \overset{\mathcal{T}_{s_1 s_2 s_3}}{\longmapsto} \begin{pmatrix} \tilde{a} \\ \tilde{b} \\ \tilde{c} \\ \tilde{d} \end{pmatrix} \overset{\mathcal{H}}{\longmapsto} \begin{pmatrix} a' \\ b' \\ c' \\ d' \end{pmatrix}, \tag{5}$$

for some $s_1, s_2, s_3 \in \{0,1\}$. This relation can be encoded in the following R1CS.

First, our key observation is that $\mathcal{H} \circ \mathcal{H} = 4\mathbf{I}_4$. This implies that

$$4 \cdot \mathcal{S}(a,b,c,d) = \mathcal{H}(A',B',C',D') \quad \text{and} \quad 4 \cdot (\tilde{a},\tilde{b},\tilde{c},\tilde{d}) = \mathcal{H}(a',b',c',d').$$

with a slight abuse of notation, we also denote with $\mathcal{H}$ the matrix underlying this operator. We define the following matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$:

$$\mathbf{A} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathcal{H} \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathcal{H} \end{pmatrix}, \ \mathbf{C} = \begin{pmatrix} 0 & \mathbf{0} & \mathcal{H} & \mathbf{0} \\ 0 & \mathbf{0} & 16\mathbf{I}_4 & \mathbf{0} \end{pmatrix}.$$

Such matrices encode the R1CS relation for the witness

$$z = (1, a, b, c, d, A', B', C', D', a', b', c', d')$$

Using the identity $\mathcal{H} \circ \mathcal{H} = 4\mathbf{I}_4$, the optimized constraints above can be expressed entirely in squaring form. Consequently, their cost is 14 constraints over $\mathbb{F}_p$. Compared with a straightforward encoding of Eq. (5) from left to right, this saves at least 50% of the constraints.

Remark 2. As explained in [43, §2.4], when computing Φ_1, we have that $\tilde{d} = 0$. This automatically leads to $D' = 0$, hence reducing two variables in this case and one constraint (i.e. the last row above).

5.2 Constructing theta-null points for elliptic products

Our application is to encode $(2^n, 2^n)$-isogenies $\Phi = \Phi_n \circ \ldots \circ \Phi_1 : E_1 \times E_2 \to E_3 \times E_4$ into a R1CS. Hence, we first have to deal with the problem of generating a valid theta-null point for the elliptic product $E_1 \times E_2$. Here, we assume that E_1 and E_2 are given in the Montgomery form, *i.e.*, $E_1 : y^2 = x^3 + A_1 x^2 + x$ and $E_2 : y^2 = x^3 + A_2 x^2 + x$.

The theta-null point describing $E_1 \times E_2$ must be *compatible* with $\ker(\Phi)[4]$; being compatible means that the theta-null point representing $E_1 \times E_2$ can be used to compute the isogeny Φ_1. Algorithms to construct such a compatible theta-null point are described in [35, §2.3] and [43]. However, since we are only interested in constructing a R1CS, we will only focus on explaining how $E_1 \times E_2$

are related to the compatible theta-null point; we denote such a point with $(a : b : c : d)$.

The first step is to adjust the theta-null point $(a : b : c : d)$ into the *canonical position*; here, with canonical position we refer to the *product principal polarization* in the sense of [35, Proposition 10]. As explained in [35, §4.1], there are ten possible matrices that send $(a : b : c : d)$ into a theta-null point in canonical position.[5] The matrix to use is determined by the position of the zero in dual theta coordinates [35, §4.1]. As explained in [43, §2.4], when using the four-torsion above as in our case, the zero coordinate is always in the last position. In practice, this means that $\begin{pmatrix} a' \\ b' \\ c' \\ d' \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$.

Remark 3. By checking that $(a' : b' : c' : d')$ is sent to a theta-null point in canonical position, we are also proving that $(a' : b' : c' : d')$ is not a theta-null point in canonical position. This means that the isogeny Φ_1 does not map onto an elliptic product. Since $E_1 \not\cong E_2$, this means that q must be odd [35, Appendix A, Case 3].

Once we have $(a' : b' : c' : d')$, we define

$$(a_1, b_1) := (a', c') \quad \text{and} \quad (a_2, b_2) := (a', b'). \tag{6}$$

Then, we have that $A'_i = -\frac{(a_i^2+b_i^2)^2-(a_i^2-b_i^2)^2}{(a_i^2+b_i^2)\cdot(a_i^2-b_i^2)}$. The following encoding takes 13 $\mathbb{F}_p$-constraints for each A'_i.

$$\begin{aligned}
&\text{(Row 1)} && a \cdot a && = x \\
&\text{(Row 2)} && b \cdot b && = y \\
&\text{(Row 3)} && x \cdot x && = x_2 = a^4 \\
&\text{(Row 4)} && y \cdot y && = y_2 = b^4 \\
&\text{(Row 5)} && (x+y) \cdot (x+y) && = s_2 = (x+y)^2 \\
&\text{(Row 6)} && A' \cdot (x_2 - y_2) && = (-2s_2 + 2x_2 + 2y_2)
\end{aligned}$$

Finally, the last step then consists of checking that the curve with equation $y^2 = x^3 + A'_i x^2 + x$ and the one with equation $y^2 = x^3 + A_i x^2 + x$ are isomorphic. This amounts to checking that their j-invariants coincide, *i.e.*, we need to check that $(A_i'^2 - 3)^3 \cdot (A_i^2 - 4) = (A_i^2 - 3)^3 \cdot (A_i'^2 - 4)$. It takes 20 constraints or 15

[5] Such matrices have been hardcoded in the library https://github.com/ThetaIsogenies/two-isogenies.

(optimized) constraints over $\mathbb{F}_p$ as follows.

$$\begin{aligned}
&\text{(Row 1)} && A_i \cdot A_i && = u && = A_i^2,\\
&\text{(Row 2)} && A'_i \cdot A'_i && = v && = A_i'^2,\\
&\text{(Row 3)} && (u-3)\cdot(u-3) && = p_2 && = (A_i^2-3)^2,\\
&\text{(Row 4)} && p_2 \cdot (u-3) && = p_3 && = (A_i^2-3)^3,\\
&\text{(Row 5)} && (v-3)\cdot(v-3) && = q_2 && = (A_i'^2-3)^2,\\
&\text{(Row 6)} && q_2 \cdot (v-3) && = q_3 && = (A_i'^2-3)^3,\\
&\text{(Row 7)} && q_3 \cdot (u-4) && = t && = (A_i'^2-3)^3(A_i^2-4),\\
&\text{(Row 8)} && p_3 \cdot (v-4) && = t && = (A_i^2-3)^3(A_i'^2-4).
\end{aligned}$$

$$\begin{aligned}
&\text{(Row 1)} && A \cdot A && = x && \\
&\text{(Row 2)} && A' \cdot A' && = x' && \\
&\text{(Row 3)} && x \cdot x' && = m_1 && \\
&\text{(Row 4)} && (x+x')\cdot(x+x') && = s_1 && \\
&\text{(Row 5)} && m_1 \cdot (x+x') && = m_2 && = (xx')(x+x')\\
&\text{(Row 6)} && (x-x')\cdot\big(m_2 - 4s_1 - 5m_1 + 36x + 36x' - 81\big) && = 0 &&
\end{aligned}$$

We face a similar problem when we compute the last isogeny Φ_n mapping onto $E_3 \times E_4$. After computing Φ_n, we are left with a theta-null point $(a'_n : b'_n : c'_n : d'_n)$. Following [34, Proposition 57], to get a theta-null point $(a_n : b_n : c_n : d_n)$ with respect to the product theta-structure, we have to apply the same matrix $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}$, *i.e.*, $\begin{pmatrix} a_n \\ b_n \\ c_n \\ d_n \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} a'_n \\ b'_n \\ c'_n \\ d'_n \end{pmatrix}$.

As explained in [35, §4.1], the matrix $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}$ sends the theta-null point $(a'_n : b'_n : c'_n : d'_n)$ to a theta-null point in the canonical product structure if and only if $a_n'^2 - b_n'^2 - c_n'^2 + d_n'^2 = 0$.[6] Checking this last equation ensures that the last theta-null point describes a product of two elliptic curves as desired. The following procedure takes 6 $\mathbb{F}_p$-constraints.

$$(a'_n + b'_n)\cdot(a'_n - b'_n) = t \qquad \text{and} \qquad (d'_n + c'_n)\cdot(d'_n - c'_n) = -t.$$

5.3 Evaluating (2, 2)-isogenies in a R1CS

In this section, we describe the evaluation map between two isogenous theta-null points $(a : b : c : d)$ and $(a' : b' : c' : d')$ underlying the PPASes X

[6] The best way to see this is to look at the function `compute_splitting_matrix` in https://github.com/ThetaIsogenies/two-isogenies/blob/main/Theta-SageMath/theta_isogenies/isomorphism.py. This function describes the connections between the hardcoded matrices and the zeros of dual theta coordinates.

and X', respectively; this algorithm is described in [35] and is a consequence of Equation (1). Define $(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}) := \mathcal{H}(a', b', c', d')$ and assume that $\tilde{a} \cdot \tilde{b} \cdot \tilde{c} \cdot \tilde{d} \neq 0$. Given a point $(x_0 : x_1 : x_2 : x_3)$ on X, we compute its evaluation as

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \overset{\mathcal{S}}{\longmapsto} \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} \overset{\mathcal{H}}{\longmapsto} \begin{pmatrix} X'_0 \\ X'_1 \\ X'_2 \\ X'_3 \end{pmatrix} \overset{\mathcal{S}ca\ell}{\longmapsto} \begin{pmatrix} \tilde{y}_0 \\ \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{pmatrix} \overset{\mathcal{H}}{\longmapsto} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}, \tag{7}$$

where $\mathcal{S}ca\ell(x, y, z, t) = (x/\tilde{a}, y/\tilde{b}, z/\tilde{c}, t/\tilde{d})$. We now explain how to encode this step in a R1CS.

Following a similar reasoning as above, we first observe that

$$4 \cdot \mathcal{S}(x_0, x_1, x_2, x_3) = \mathcal{H}(X'_0, X'_1, X'_2, X'_3) \quad \text{and} \quad 4 \cdot (\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \tilde{y}_3) = \mathcal{H}(y_0, y_1, y_2, y_3).$$

Additionally, we have that

$$(\tilde{a} \cdot \tilde{y}_0, \tilde{b} \cdot \tilde{y}_1, \tilde{c} \cdot \tilde{y}_2, \tilde{d} \cdot \tilde{y}_3) = (X'_0, X'_1, X'_2, X'_3).$$

We define the following matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$:

$$\mathbf{A} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathcal{H} & \mathbf{0} \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_4 \end{pmatrix}, \ \mathbf{C} = \begin{pmatrix} 0 & \mathbf{0} & \mathcal{H} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & 4\mathbf{I}_4 & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Such matrices encode the R1CS relation for the witness with 20 $\mathbb{F}_p$-constraints

$$z = (1, x_0, x_1, x_2, x_3, X'_0, X'_1, X'_2, X'_3, y_0, y_1, y_2, y_3, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}).$$

5.4 Evaluating the Gluing Isogeny

In practice, we will have to encode the evaluation $(2^n, 2^n)$-isogenies $\Phi = \Phi_n \circ \ldots \circ \Phi_1 : E_1 \times E_2 \to E_3 \times E_4$ into a R1CS, *i.e.*, for a given $(T_1, T_2) \in E_1 \times E_2$, we want to encode $\Phi(T_1, T_2)$. Similarly to Section 5.2, we first have to deal with the problem of representing (T_1, T_2) in theta coordinates. As above, we assume that E_1 and E_2 are given in the Montgomery form, *i.e.*, $E_1 : y^2 = x^3 + A_1 x^2 + x$ and $E_2 : y^2 = x^3 + A_2 x^2 + x$.

We now explain the relation between (T_1, T_2) and its representation (x_0, x_1, x_2, x_3) in theta coordinates. As explained in Section 5.2, by applying the transformation $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}$, we obtain a point (x'_0, x'_1, x'_2, x'_3)represented with respect to product theta structure; to be more precise

$$\begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Using the (a_i, b_i)'s from Equation (6), we can apply the conversion formulas between theta structures and Montgomery form [86, Ch.7, Appendix B]. We

define $(x_1, z_1) := (x'_0, x'_2)$ and $(x_2, z_2) := (x'_0, x'_1)$. Then, the point $(X'_i : Z'_i) = (a_i z_i + b_i x_i : a_i z_i - b_i x_i)$ is a point on the Kummer line underlying the curve $E'_i : y^2 = x^3 + A'_i x^2 + x$. Here identifying $(X'_i : Z'_i)$ takes 6 $\mathbb{F}_p$-constraints.

In general, the curves E_i and E'_i are only isomorphic, not identical. We now explain how to map the point $(X'_i : Z'_i)$ onto a point on E_i; this conversion algorithm is described in [1, Algorithm 8.9]. We first compute $\lambda_x = (2A_i^3 - 9A_i)(3 - A_i'^2)$ and $\lambda_z = (2A_i'^3 - 9A'_i)(3 - A_i^2)$. A straightforward encoding takes 8 $\mathbb{F}_p$-constraints for each. Then, the new point on the Kummer Line of E_i has coordinates $(\lambda_x(3X'_i + A'_i Z'_i) - \lambda_z A_i Z'_i : 3\lambda_z Z'_i)$. Hence, we need to check that x-coordinate of T_i coincides with $\frac{\lambda_x(3X'_i + A'_i Z'_i) - \lambda_z A_i Z'_i}{3\lambda_z Z'_i}$. The following encoding takes 15 $\mathbb{F}_p$-constraints for each T_i.

$$
\begin{aligned}
&\text{(Row 1)} && A'_i \cdot Z'_i && = s, && \\
&\text{(Row 2)} && \lambda_z \cdot Z'_i && = r, && \\
&\text{(Row 3)} && \lambda_x \cdot (3X'_i + s) && = u && = \lambda_x(3X'_i + A'_i Z'_i), \\
&\text{(Row 4)} && A_i \cdot r && = q && = \lambda_z A_i Z'_i, \\
&\text{(Row 5)} && X \cdot (3r) && = u - q && \iff X(3\lambda_z Z'_i) = \lambda_x(3X'_i + A'_i Z'_i) - \lambda_z A_i Z'_i.
\end{aligned}
$$

Now, we focus on encoding the evaluation $(y_0 : y_1 : y_2 : y_3)$ of the isogeny Φ_1 at the point $(x_0 : x_1 : x_2 : x_3)$ once we have the theta-null points $(a : b : c : d)$ and $(a' : b' : c' : d')$ for the domain and the codomain, respectively. Define $(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}) := \mathcal{H}(a', b', c', d')$. As explained in [43, §2.4], we have that $\tilde{d} = 0$, leaving us with the problem that the operator $\mathcal{S}ca\ell$ in Equation (7) is not defined on the last coordinate. In what follows, we apply the strategy in [35, §4] to counter this problem.

The tuple $(x_0 : x_1 : x_2 : x_3)$ represents the point $(T_1, T_2) \in E_1 \times E_2$ in theta coordinates. As explained in [35, §4], there exists a special point $(K_1, K_2) \in E_1 \times E_2$ that allows us to recover the last coordinate of the evaluation map, even if $\tilde{d} = 0$. Such a special point (K_1, K_2) can be represented in theta coordinates by $(1 : 0 : -1 : 0)$.[7] Define $(x_0^{K+T} : x_1^{K+T} : x_2^{K+T} : x_3^{K+T})$ to be theta coordinates of $(K_1, K_2) + (T_1, T_2)$, and define $(\tilde{y}_0, \tilde{y}_1, \tilde{y}_2, \tilde{y}_3)$ as $4\mathcal{H}(y_0, y_1, y_2, y_3)$. After normalising the theta coordinates appropriately, we have that

$$\mathcal{H} \circ \mathcal{S}(x_0, x_1, x_2, x_3) = (\tilde{a} \cdot \tilde{y}_0, \tilde{b} \cdot \tilde{y}_1, \tilde{c} \cdot \tilde{y}_2, 0), \tag{8}$$

and

$$\mathcal{H} \circ \mathcal{S}(x_0^{K+T}, x_1^{K+T}, x_2^{K+T}, x_3^{K+T}) = (\tilde{a} \cdot \tilde{y}_1, \tilde{b} \cdot \tilde{y}_0, \tilde{c} \cdot \tilde{y}_3, 0). \tag{9}$$

Utilising the same notation as in Section 5.3, we encode Equation (8) in a R1CS with 17 $\mathbb{F}_p$-constraints as follows. We define the following matrices $\mathbf{A}$, $\mathbf{B}$

[7] The point (K_1, K_2) is a point of 4-torsion that corresponds to T'_i in the notation of [35, §4]. From [43, Proposition 1], it follows that (K_1, K_2) is given as $(x : 0 : y : 0)$ in theta coordinates. After applying the evaluation formula, we get that $x^2 = y^2$, from which it follows that (K_1, K_2) is given as $(1 : 0 : \pm 1 : 0)$. Indeed, picking either choice will work since $\Phi_1(1 : 0 : 1 : 0) = \Phi_1(1 : 0 : -1 : 0)$.

and $\mathbf{C}$:

$$\mathbf{A} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathcal{H}_{(3)} & \mathbf{0} \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{pmatrix}, \ \mathbf{C} = \begin{pmatrix} 0 & \mathbf{0} & \mathcal{H}^{(3)} & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & 4\mathbf{I}_3 & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where $\mathcal{H}^{(3)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}$ and $\mathcal{H}_{(3)} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{pmatrix}$. Such matrices encode the R1CS relation for the witness $z = (1, x_0, x_1, x_2, x_3, X'_0, X'_1, X'_2, y_0, y_1, y_2, y_3, \tilde{a}, \tilde{b}, \tilde{c})$.

Also, after defining the $(X_0'^{K+T}, X_1'^{K+T}, X_2'^{K+T}, 0)$ for $(x_0^{K+T}, x_1^{K+T}, x_2^{K+T}, x_3^{K+T})$, we encode Equation (9) in a R1CS as follows with 11 $\mathbb{F}_p$-constraints as follows. We define the following matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$:

$$\mathbf{A} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} & 0 \\ 0 & \mathbf{0} & \mathbf{0} & h_4 & 0 \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 0 & 2\mathbf{I}_4 & \mathbf{0} & \mathbf{0} & 0 \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{pmatrix}, \ \mathbf{C} = \begin{pmatrix} 0 & \mathbf{0} & \mathcal{H} & \mathbf{0} & 0 \\ 0 & \mathbf{0} & 4\mathbf{v} & \mathbf{0} & 0 \end{pmatrix},$$

where $\mathcal{H}^{(3)} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}$, $h_4 = (1, -1, -1, 1)$ and $\mathbf{v} = (0, 0, 1)$. Such matrices encode the R1CS relation for the witness

$$z = (1, x_0^{K+T}, x_1^{K+T}, x_2^{K+T}, x_3^{K+T}, X_0'^{K+T}, X_1'^{K+T}, X_2'^{K+T}, y_0, y_1, y_2, y_3, \tilde{c}).$$

The last step is to relate (x_0, x_1, x_2, x_3) and $(x_0^{K+T}, x_1^{K+T}, x_2^{K+T}, x_3^{K+T})$. We convert:

1. $(x_0 : x_1 : x_2 : x_3)$ to a point $T = (T_1, T_2) \in E_1 \times E_2$,
2. $(x_0^{K+T} : x_1^{K+T} : x_2^{K+T} : x_3^{K+T})$ to a point $S = (S_1, S_2) \in E_1 \times E_2$
3. $(1 : 0 : -1 : 0)$ to a point $K = (K_1, K_2) \in E_1 \times E_2$.

The pairs of points T_i, S_i, K_i are given as points on the Kummer line. We provide the y-coordinates of each of these points as part of the witness. We finally check that $T_i - S_i = K_i$. Here, we spend 14 $\mathbb{F}_p$-constraints to enforce well-formedness of the two y-coordinates, and an additional 8 $\mathbb{F}_p$-constraints to check the full-coordinate addition.

5.5 Evaluating the Splitting Isogeny

We now deal with the last step of evaluating the $(2^n, 2^n)$-isogeny, *i.e.*, if $\Phi = \Phi_n \circ \ldots \circ \Phi_1 : E_1 \times E_2 \to E_3 \times E_4$, we evaluate the isogeny Φ_n in theta coordinates. First, we need to recover an equation for the Montgomery form of E_3 and E_4 from the codomain theta-null point of Φ_n; define this to be $(a'_n : b'_n : c'_n : d'_n)$. As explained in Section 5.2, we need to recover the theta-null point $(a_n : b_n : c_n : d_n)$ with respect to the product theta structure. This is given as

$$\begin{pmatrix} a_n \\ b_n \\ c_n \\ d_n \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} a'_n \\ b'_n \\ c'_n \\ d'_n \end{pmatrix}.$$

After getting the theta-null point $(a_n : b_n : c_n : d_n)$ compatible with the product theta structure, we compute the Montgomery invariants as in Section 5.2. Let us define

$$(a_3, b_3) := (a_n : b_n) \quad \text{and} \quad (a_4, b_4) := (b_n : d_n). \tag{10}$$

We observe that, as initially remarked in [10, Remark 12], we can always ensure that E_3 is the curve corresponding to the theta-null point $(a_3 : b_3)$ and E_4 is the curve corresponding to the theta-null point $(a_4 : b_4)$. Then, for $i = 3, 4$, the Montgomery invariant A_i for the curve $E_i : y^2 = x^3 + Ax^2 + X$ is $A_i = -\frac{(a_i^2+b_i^2)^2-(a_i^2-b_i^2)^2}{(a_i^2+b_i^2)\cdot(a_i^2-b_i^2)}$.

Suppose that we have a point $(x_0 : x_1 : x_2 : x_3)$ in theta coordinates for $E_3 \times E_4$, our goal is to recover its Montgomery coordinates on both E_3 and E_4. This can be done as above. We first apply the transformation $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}$ to (x_0, x_1, x_2, x_3) and obtain a point (x'_0, x'_1, x'_2, x'_3). We then use the conversion formulas between the theta model and Montgomery form on the points $(x_3, z_3) := (x'_0, x'_1)$ and $(x_4, z_4) := (x'_1, x'_3)$.

We finally convert from (x_3, z_3) and (x_4, z_4) to points on the Kummer line of E_3 and E_4 as follows. The pair (x_3, y_3) is sent to the point $(X_i : Z_i) = (a_i z_i + b_i x_i : a_i z_i - b_i x_i)$ on the Kummer line underlying the curve $E_i : y^2 = x^3 + A_i x^2 + x$. Similarly, it takes 6 $\mathbb{F}_p$-constraints.

5.6 Constructing a proof for $\mathcal{R}_q^{\mathsf{eval}}$

We now gather what we described above to construct a proof for a $(E, P, E', P') \in \mathcal{L}_{\mathcal{R}_q^{\mathsf{eval}}}$. First, we need to check that the witness φ for proving $(E, P, E', P') \in \mathcal{L}_{\mathcal{R}_q^{\mathsf{eval}}}$ has degree $q(2^n - q)$. Here, we assume that the 2^n-torsion of E is defined over $\mathbb{F}_{p^2}$. Let (R_1, R_2) be a basis of $E[2^n]$. Using the evaluation techniques above, we evaluate Φ, the isogeny encoding φ, at $(R_1, \mathcal{O}_{E'})$ and $(R_2, \mathcal{O}_{E'})$; define $(R'_1, *)$ and $(R'_2, *)$ to be their evaluation. Our goal is to encode the pairing $e_{2^n}(R'_1, R'_2)$ and then check that it is equal to $e_{2^n}(R_1, R_2)^q$. However, we do encounter a problem since we are only able to compute $\pm R'_i$ since we will be working on Kummer lines.

The problem of picking the same sign for $\pm R'_1$ and $\pm R'_2$ is countered as follows. We also evaluate Φ at $(R_1 - R_2, \mathcal{O}_{E'})$; we define $(R'_{1,2}, *)$ to be such an evaluation. We then encode the y-coordinates of such points as part of a witness in a R1CS. We then check that $R'_1 - R'_2 = R'_{1,2}$. Here, we spend 14 $\mathbb{F}_p$-constraints to enforce well-formedness of the two y-coordinates, and an additional 8 $\mathbb{F}_p$-constraints to check the full-coordinate addition.

Once we have a proof for the fact that we picked the correct signs, we encode the computation of $e_{2^n}(R'_1, R'_2)$ in a R1CS and check that it is equal to $e_{2^n}(R_1, R_2)^q$. We highlight that $e_{2^n}(R_1, R_2)^q$ is a public value.

Once we are convinced that the degree is correct, we have to compute a proof for $\begin{pmatrix} q-2^n & 0 \\ 0 & q \end{pmatrix} \cdot \Phi \begin{pmatrix} P \\ \mathcal{O}_{E_2} \end{pmatrix} = \Phi \begin{pmatrix} \mathcal{O}_{E_1} \\ P' \end{pmatrix}$. The proof requires twice the constraints for the evaluations presented before, which is roughly $40n + 148$. Since q is public we can reuse the methods in Section 3.3 without incurring more constraints,

Remark 4. Suppose that we have that the point P has order q and is rational (or defined over $\mathbb{F}_{p^4}$). Additionally, suppose that $P' = \mathcal{O}_{E_2}$. In this case, we can skip the check on the degree of the isogeny φ and we only have to prove

Table 2: Summary of the $\mathbb{F}_p$-constraint costs of our constructions for various isogeny tasks in 2-dimensional cases with the overhead required to support the corresponding functionality.

	2-D Isogeny of	Eval	q-Degree w. Pairing	q-Degree w. Torsion
Constraints	$14n + 84$	$40n + 148$	$17n$	0

that $\begin{pmatrix} q-2^n & 0 \\ 0 & q \end{pmatrix} \cdot \Phi \begin{pmatrix} P \\ \mathcal{O}_{E_2} \end{pmatrix} = \begin{pmatrix} \mathcal{O}_{E_3} \\ \mathcal{O}_{E_4} \end{pmatrix}$. This is significantly more efficient than the general evaluation routine. We will use this specific case for the application in Section 6.1.

5.7 Summary for 2-dimensional Isogeny

We summarize our constraint costs in Table 2. When the relevant torsion-point evaluations are provided, we can avoid encoding the pairing relation in R1CS and instead verify the pairing directly, which is the case for our application to DeuringVRF later.

6 Cryptographic applications

In this section, we briefly mention some concrete cryptographic applications of the above-mentioned tools.

6.1 Isogeny-based Signature Schemes with better Security.

Here we take DeuringVRF for instance. DeuringVRF (a.k.a. DeuringVUF) [68] is the first efficient hash-and-sign-paradigm-based signature in isogeny-based cryptography. Its security relies on the hardness of computing a large prime-degree isogeny without access to an endomorphism-ring trapdoor, which is captured by the *one-more evaluation* problem.

We give a simplified overview of DeuringVRF. Let $\Pi_{\mathsf{DVRF}} = (\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Verify})$. The public parameter includes a large prime p, a starting curve $E_0/\mathbb{F}_{p^2}$ with known endomorphism ring, and a cryptographically large prime q. A public key pk consists of a curve E together with an accessible q-torsion basis $\{P, Q\} \subseteq E[q]$; in practice, $P, Q \in E(\mathbb{F}_{p^4})$. The secret key sk is (a representation of) the endomorphism ring of E, which enables efficient evaluation of the required q-isogenies.

To evaluate (i.e., sign) on an input $x \in \{1, \ldots, q-1\}$, the signer uses the trapdoor to construct an isogeny $\phi_x : E \to E_x$ whose kernel is generated by $P + xQ$, and outputs the codomain curve E'. For verifiability, the signer additionally provides a proof that ϕ_x indeed has codomain E' and maps $P + xQ$ to the point at infinity. Concretely, this proof is given via the kernel (U_3, V_3) and the auxiliary curve E_3 of a suitable two-dimensional isogeny that contains ϕ_x as a component, allowing the verifier to check the claimed kernel relation. We denote this relation as R and $((x, E_3, \mathsf{pk}), (E_x, U_3, V_3)) \in R$.

In the full construction, an injective (or collision-resistant) hash maps messages to elements of $E[q]$ to determine the kernel; this detail will not affect our application. Therefore, it is natural to view the underlying hardness as an instance of the *one-more* evaluation assumption. We show how to apply our proof system to remove the *one-more* assumption, at the cost of an additional proof as overhead.

Algorithm 1: DeuringVRF Compiler

Input: Message x, the public key pk, NIZK $\Pi = (\mathsf{Prove}, \mathsf{Ver})$.
Output: (x, E_3, π) where E_3 is the new output and π is the new proof.

1 $E_x, \pi' \leftarrow \mathsf{Eval}(x)$ // $\pi' = (E_3, U_3, V_3)$, `old proof.`
2 $\pi \leftarrow \mathsf{Prove}((x, E_3, \mathsf{pk}), (E_x, U_3, V_3))$ // $((x, E_3, \mathsf{pk}), (E_x, U_3, V_3)) \in R$
3 **return** (x, E_3, π)

Problem 5 (Pseudorandomness Problem). Let $\mathsf{pk} \leftarrow \mathsf{KeyGen}$, and let E denote the public-key curve in pk. A PPT adversary $\mathcal{A}$ is given oracle access to $\mathcal{C}_b$, where the oracle samples a hidden bit $b \leftarrow\!\$ \{0,1\}$ once at the beginning of the experiment.

On each query $x \in \{1, \ldots, q-1\}$, the oracle responds as follows:

- If $b = 0$, it samples a curve E' by performing a random isogeny walk from E statistically uniform into the isogeny graph, and returns E'. We denote the distribution as $E' \leftarrow \mathsf{Walk}(E)$.
- If $b = 1$, it runs $(E_x, (E', U, V)) \leftarrow \mathsf{Eval}(x)$ and returns the evaluation output curve E'.

At the end of the experiment, $\mathcal{A}$ outputs a bit b' as its guess for b.

We define the advantage of $\mathcal{A}$ in Problem 5 as

$$\mathsf{Adv}^{\mathsf{PP}}_{\Pi_{\mathsf{DVRF}}}(\mathcal{A}) := \left| \Pr\left[b' = b\right] - \tfrac{1}{2} \right|, \quad \text{where } b \leftarrow\!\$ \{0,1\},\ b' \leftarrow \mathcal{A}^{\mathcal{C}_b}.$$

Theorem 6. *If* NIZK *is proof of knowledge and zero-knowledge, then DeuringVRF compiled from above is UF-CMA if Problem 5 is hard.*

Proof. To see the result, we use a hybrid argument by introducing a sequence of games $\mathsf{Game}_0, \ldots, \mathsf{Game}_2$ and modifying them gradually. In each game, we apply the extractor of Π to the adversary $\mathcal{A}$. Let Adv_i denote the probability that extraction succeeds in Game_i.

Game_0*: real experiment.* Game_0 is the real UF-CMA experiment. If extraction fails with non-negligible probability, we can construct an adversary $\mathcal{B}_0$ that breaks the argument-of-knowledge property of Π. Concretely, $\mathcal{B}_0$ runs $\mathcal{A}$ internally and answers its oracle queries honestly; when $\mathcal{A}$ outputs a final proof, $\mathcal{B}_0$ forwards it to the extractor for Π. Therefore, $\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}(\mathcal{A}) - \mathsf{Adv}_0 \leq \mathsf{Adv}^{\mathsf{AoK}}(\mathcal{B}_0)$.

Game_1*: simulated proofs.* Game_1 is identical to Game_0 except that all proofs are generated using the simulator $\mathcal{S}$ of Π rather than the real prover. This change can only be noticed if $\mathcal{A}$ distinguishes simulated proofs from real ones, i.e., breaks the zero-knowledge property of Π.

We define $\mathcal{B}_2$ against the zero-knowledgeness of Π as follows. $\mathcal{B}_2$ sets up all public parameters of DeuringVRF and generates the public key. Upon a signing request on m, $\mathcal{B}_2$ computes the corresponding statement (and witness, when needed) and forwards the statement to its ZK challenger, receiving back a proof that is either real or simulated depending on the challenge bit. It then forwards the received proof to $\mathcal{A}$.

At the end, $\mathcal{B}_2$ outputs a random bit if extraction succeeds, and 0 otherwise. Hence, $\mathsf{Adv}_1 - \mathsf{Adv}_0 \leq Q_s \cdot \mathsf{Adv}^{\mathsf{ZK}}(\mathcal{B}_2)$.

Game_2*: replacing the codomain curve.* Game_2 is the same as Game_1 except that, for each evaluation (signing) query, instead of running Eval to compute the output curve E', the challenger samples E' by $E' \leftarrow \mathsf{Walk}(E)$. All proofs are still generated via $\mathcal{S}$.

The only way $\mathcal{A}$ can notice this change is by distinguishing whether the returned curve E' comes from a genuine evaluation $\mathsf{Eval}(\mathsf{sk}, \cdot)$ or from the replacement distribution. We therefore build a distinguisher $\mathcal{B}_3$ for the corresponding distinguishing problem as follows.

$\mathcal{B}_3$ is given access to a challenge procedure that, on each query, returns a curve distributed either as in Game_1 (real E') or as in Game_2 (replaced E'), depending on a hidden challenge bit. $\mathcal{B}_3$ runs $\mathcal{A}$ internally and answers all of $\mathcal{A}$'s evaluation queries by: (i) obtaining E' from its challenge procedure, and (ii) using $\mathcal{S}$ to generate a proof consistent with the returned E' and the claimed kernel relation. Finally, $\mathcal{B}_3$ outputs a random bit iff $\mathcal{A}$'s output behavior is consistent with the Game_1-world rather than the Game_2-world (e.g., extraction succeeds), and 0 otherwise. By the hybrid reasoning, this implies $\mathsf{Adv}_2 - \mathsf{Adv}_1 \leq \mathsf{Adv}^{\mathsf{PP}}(\mathcal{B}_3)$.

Now we can use the adversary to break the evaluation problem by constructing $\mathcal{B}_4$ as follows. Concretely, upon receiving the challenge public key pk with E_{pk} and the q-torsion basis $\{P, Q\}$, $\mathcal{B}_4$ forwards them to $\mathcal{A}$. For each signing query of m, $\mathcal{B}_4$ generates a random curve $E' \leftarrow \mathsf{Walk}(E_{\mathsf{pk}})$ and runs $\mathcal{S}$ to generate a proof π. At the end of the experiment, after receiving (x, E', π), $\mathcal{B}_4$ invokes the extractor $\mathcal{E}$ of Π. If $\mathcal{B}_4$ obtains a 2-dimensional isogeny containing an isogeny mapping from E, E_x with kernel $P + [x]Q$, then outputs 1. If extraction fails, then $\mathcal{B}_4$ returns 0.

Hence, $\mathsf{Adv}_2 \leq \mathsf{Adv}^{\mathsf{PP}}(\mathcal{B}_4)$ and thereby

$$\mathsf{Adv}^{\mathsf{UFCMA}} \leq \mathsf{Adv}^{\mathsf{AoK}}(\mathcal{B}_0) + Q_s \cdot \mathsf{Adv}^{\mathsf{ZK}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{PP}}(\mathcal{B}_3) + \mathsf{Adv}^{\mathsf{PP}}(\mathcal{B}_4).$$

Compared to the original version, where the prover must provide the kernel of a two-dimensional isogeny, our compiled version reveals significantly less information. Indeed, knowing such a kernel typically enables the verifier to evaluate the corresponding isogeny on arbitrary torsion points beyond $P + [x]Q$. In contrast, in the compiled version the verifier only learns that the prover knows

an isogeny with the specified kernel, which minimized the information learned by the verifier.

Remark 7 (Other Applications.). A similar compiler can be instantiated for PRISM [8] to relax its underlying (one-more) assumption. In that case, however, the overhead is significantly larger, since PRISM's verification involves a pairing computation in order to certify the degree of the isogeny. As mentioned in Section 2.3 while pairing computations can in principle be encoded within zkSNARKs, doing so typically incurs a large amount of constraints overhead [44].

Along the same lines, one may apply our approach to the SQISign2D family [10, 80] by combining the two-dimensional isogeny proof with a 2-isogeny proof. This can eliminate the need for additional isogeny oracles (or hinted assumptions) and instead rely solely on the underlying isogeny problem. The resulting overhead is expected to be lower than in DeuringVRF, since no proof of evaluation is required. In theory, one may also apply our proof system to classical SQISign [40] to obtain a cleaner zero-knowledge guarantee. In SQISign, zero-knowledge currently hinges on a KLPT-dependent distinguishing problem, which is not statistical as observed in [22]. However, the corresponding responses involve several large-degree torsion components (from a few hundreds up to around a thousand), which would likely make the proof substantially more expensive. This also motivates the question of whether the KLPT algorithm can be improved so that the response isogeny has a much smoother degree (e.g., a power of 2).

At a glance, these applications trade efficiency for cleaner assumptions, which is a bit costly. We believe it is an important first step toward developing more advanced isogeny-based cryptosystems in the future.

6.2 Key Validation in POKÉ

Most non-oriented isogeny-based public-key encryption schemes remain vulnerable to adaptive attacks in the spirit of the GPST attack [52], originally mounted against SIDH [60]. This issue appears recursively in several SIDH-derived settings [48, 49, 51] and persists even in recent proposals [11, 77], as further confirmed by follow-up analyses [77, 71]. At a high level, the attack exploits the fact that an adversary can submit malformed ciphertexts, or even malformed public keys, to a party holding a static secret key and observe whether decryption or the handshake procedure succeeds. This success/failure behavior acts as an oracle that leaks information about the static secret key; by adaptively repeating the interaction, the adversary can eventually recover the entire key.

A standard countermeasure is to apply the Fujisaki–Okamoto transform [50]. In the isogeny setting, however, this approach requires the decapsulating party to re-derive the encryption randomness and check ciphertext consistency. This makes the resulting mechanism inherently asymmetric for static-key use: the randomness of one party must be recoverable and verified, and therefore the approach does not directly support a setting in which both parties simultaneously keep static secret keys. An alternative, proposed in [37], is to require both parties to provide a proof of knowledge of the randomness used to generate their public keys. This allows both sides to use static secret keys while ruling out maliciously generated public keys.

Using the zero-knowledge proofs developed in this work for both the isogeny relation and the evaluation relation, we can efficiently instantiate such proofs of knowledge for the schemes of [11, 77].

7 Conclusion and Future Work

This work takes a step toward making zkSNARKs practically usable for a broader set of isogeny relations beyond the classical *isogeny path knowledge* language. Motivated by the gap between modern isogeny-based constructions and what generic proof systems can efficiently support, we systematically develop zkSNARK-friendly encodings in the R1CS model for relations that naturally arise in contemporary isogeny-based protocols.

We illustrate how these encodings can serve as compilation tools for cryptographic applications. In particular, they enable replacing certain protocol-level assumptions (e.g., one-more evaluation) or auxiliary oracles by explicit zero-knowledge proofs of the corresponding isogeny relations, at the price of extra constraints. We believe it is an important first step toward developing more advanced isogeny-based cryptosystems in the future.

While this work focuses on R1CS to emphasize broad compatibility across zkSNARK frameworks, the derived relations are multivariate polynomial in nature. A systematic translation into other proof-system representations (e.g., relation-tailored encodings) may yield further efficiency gains and deserves a dedicated study. We leave the exploration of this direction as an important avenue for future work.

Overall, these results provide a first library of zkSNARK-friendly isogeny gadgets that go beyond path proofs, and we hope they will serve as a useful stepping stone toward more expressive and modular isogeny-based protocols with native zero-knowledge support.

References

1. Aardal, M.A., Adj, G., Aranha, D.F., Basso, A., Canales Martínez, I.A., Chávez-Saab, J., Santos, M.C., Dartois, P., De Feo, L., Duparc, M., Eriksen, J.K., Fouotsa, T.B., Filho, D.L.G., Hess, B., Kohel, D., Leroux, A., Longa, P., Maino, L., Meyer, M., Nakagawa, K., Onuki, H., Panny, L., Patranabis, S., Petit, C., Pope, G., Reijnders, K., Robert, D., Rodríguez Henríquez, F., Schaeffler, S., Wesolowski, B.: SQIsign. Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures 24
2. Alagic, G., Bros, M., Ciadoux, P., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Silberg, H., Smith-Tone, D., Waller, N.: Status report on the fourth round of the nist post-quantum cryptography standardization process. Tech. Rep. NIST IR 8545, National Institute of Standards and Technology (2025). https://doi.org/10.6028/NIST.IR.8545 2
3. Albrecht, M.R., Davidson, A., Deo, A., Smart, N.P.: Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. In: Garay, J. (ed.) PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 12711, pp. 261–289. Springer,

Cham, Switzerland, Virtual Event (May 10–13, 2021). https://doi.org/10.1007/978-3-030-75248-4_10 2
4. Albrecht, M.R., Fenzi, G., Lapiha, O., Nguyen, N.K.: SLAP: Succinct lattice-based polynomial commitments from standard assumptions. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part VII. Lecture Notes in Computer Science, vol. 14657, pp. 90–119. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58754-2_4 2
5. Allombert, B., Biasse, J.F., Eriksen, J.K., Kutas, P., Leonardi, C., Page, A., Scheidler, R., Bagi, M.T.: Faster SCALLOP from non-prime conductor suborders in medium sized quadratic fields. In: Jager, T., Pan, J. (eds.) PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part III. Lecture Notes in Computer Science, vol. 15676, pp. 333–363. Springer, Cham, Switzerland, Røros, Norway (May 12–15, 2025). https://doi.org/10.1007/978-3-031-91826-1_11 2
6. Arora, S., Safra, S.: Probabilistic checking of proofs; A new characterization of NP. In: 33rd Annual Symposium on Foundations of Computer Science. pp. 2–13. IEEE Computer Society Press, Pittsburgh, PA, USA (Oct 24–27, 1992). https://doi.org/10.1109/SFCS.1992.267824 2
7. Baldimtsi, F., Cheng, J., Goyal, R., Yadav, A.: Non-interactive blind signatures: Post-quantum and stronger security. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part II. Lecture Notes in Computer Science, vol. 15485, pp. 70–104. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0888-1_3 2
8. Basso, A., Borin, G., Castryck, W., Corte-Real Santos, M., Invernizzi, R., Leroux, A., Maino, L., Vercauteren, F., Wesolowski, B.: PRISM: Simple and compact identification and signatures from large prime degree isogenies. In: Jager, T., Pan, J. (eds.) PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part III. Lecture Notes in Computer Science, vol. 15676, pp. 300–332. Springer, Cham, Switzerland, Røros, Norway (May 12–15, 2025). https://doi.org/10.1007/978-3-031-91826-1_10 2, 3, 7, 30
9. Basso, A., Codogni, G., Connolly, D., De Feo, L., Fouotsa, T.B., Lido, G.M., Morrison, T., Panny, L., Patranabis, S., Wesolowski, B.: Supersingular curves you can trust. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part II. Lecture Notes in Computer Science, vol. 14005, pp. 405–437. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30617-4_14 2, 4, 5
10. Basso, A., Dartois, P., De Feo, L., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-West - the fast, the small, and the safer. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 15486, pp. 339–370. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_11 2, 26, 30, 40
11. Basso, A., Maino, L.: POKÉ: A compact and efficient PKE from higher-dimensional isogenies. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part II. Lecture Notes in Computer Science, vol. 15602, pp. 94–123. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91124-8_4 3, 7, 8, 30, 31, 40
12. Basso, A., Maino, L., Pope, G.: FESTA: Fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part VII. Lecture Notes in Computer Science, vol. 14444, pp. 98–126. Springer,

Singapore, Singapore, Guangzhou, China (Dec 4–8, 2023). https://doi.org/10.1007/978-981-99-8739-9_4 2, 4, 7
13. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) Advances in Cryptology – EUROCRYPT 2003. Lecture Notes in Computer Science, vol. 2656, pp. 614–629. Springer Berlin Heidelberg, Germany, Warsaw, Poland (May 4–8, 2003). https://doi.org/10.1007/3-540-39200-9_38 4
14. Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for R1CS. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, Part I. Lecture Notes in Computer Science, vol. 11476, pp. 103–128. Springer, Cham, Switzerland, Darmstadt, Germany (May 19–23, 2019). https://doi.org/10.1007/978-3-030-17653-2_4 2
15. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Scalable zero knowledge via cycles of elliptic curves. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology – CRYPTO 2014, Part II. Lecture Notes in Computer Science, vol. 8617, pp. 276–294. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2014). https://doi.org/10.1007/978-3-662-44381-1_16 13
16. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part II. Lecture Notes in Computer Science, vol. 13276, pp. 95–126. Springer, Cham, Switzerland, Trondheim, Norway (May 30 – Jun 3, 2022). https://doi.org/10.1007/978-3-031-07085-3_4 2
17. Beullens, W., Dodgson, L., Faller, S.H., Hesse, J.: The 2Hash OPRF framework and efficient post-quantum instantiations. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part VIII. Lecture Notes in Computer Science, vol. 15608, pp. 332–362. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91101-9_12 2
18. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, Part II. Lecture Notes in Computer Science, vol. 12492, pp. 464–492. Springer, Cham, Switzerland, Daejeon, South Korea (Dec 7–11, 2020). https://doi.org/10.1007/978-3-030-64834-3_16 2
19. Beullens, W., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Lattice-based blind signatures: Short, efficient, and round-optimal. In: Meng, W., Jensen, C.D., Cremers, C., Kirda, E. (eds.) ACM CCS 2023: 30th Conference on Computer and Communications Security. pp. 16–29. ACM Press, Copenhagen, Denmark (Nov 26–30, 2023). https://doi.org/10.1145/3576915.3616613 2
20. Beullens, W., Seiler, G.: LaBRADOR: Compact proofs for R1CS from module-SIS. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, Part V. Lecture Notes in Computer Science, vol. 14085, pp. 518–548. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 20–24, 2023). https://doi.org/10.1007/978-3-031-38554-4_17 2
21. Bootle, J., Lyubashevsky, V., Nguyen, N.K., Sorniotti, A.: A framework for practical anonymous credentials from lattices. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, Part II. Lecture Notes in Computer Science, vol. 14082, pp. 384–417. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 20–24, 2023). https://doi.org/10.1007/978-3-031-38545-2_13 2

22. Borin, G., Lai, Y.F., Leroux, A.: Erebor and durian: Full anonymous ring signatures from quaternions and isogenies. IACR Communications in Cryptology (CiC) **1**(4), 4 (2024). https://doi.org/10.62056/ava3zivrzn 4, 30
23. Branco, P., Choudhuri, A.R., Döttling, N., Jain, A., Malavolta, G., Srinivasan, A.: Black-box non-interactive zero knowledge from vector trapdoor hash. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 15604, pp. 64–92. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91134-7_3 2
24. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 423–447. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_15 4, 19
25. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology – ASIACRYPT 2018, Part III. Lecture Notes in Computer Science, vol. 11274, pp. 395–427. Springer, Cham, Switzerland, Brisbane, Queensland, Australia (Dec 2–6, 2018). https://doi.org/10.1007/978-3-030-03332-3_15 2
26. Chávez-Saab, J., Rodríguez-Henríquez, F., Tibouchi, M.: Verifiable isogeny walks: Towards an isogeny-based postquantum VDF. In: AlTawy, R., Hülsing, A. (eds.) SAC 2021: 28th Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 13203, pp. 441–460. Springer, Cham, Switzerland, Virtual Event (Sep 29 – Oct 1, 2022). https://doi.org/10.1007/978-3-030-99277-4_21 2
27. Chen, M., Leroux, A., Panny, L.: SCALLOP-HD: Group action from 2-dimensional isogenies. In: Tang, Q., Teague, V. (eds.) PKC 2024: 27th International Conference on Theory and Practice of Public Key Cryptography, Part III. Lecture Notes in Computer Science, vol. 14603, pp. 190–216. Springer, Cham, Switzerland, Sydney, NSW, Australia (Apr 15–17, 2024). https://doi.org/10.1007/978-3-031-57725-3_7 2
28. Cong, K., Lai, Y.F., Levin, S.: Efficient isogeny proofs using generic techniques. In: Tibouchi, M., Wang, X. (eds.) ACNS 2023: 21st International Conference on Applied Cryptography and Network Security, Part II. Lecture Notes in Computer Science, vol. 13906, pp. 248–275. Springer, Cham, Switzerland, Kyoto, Japan (Jun 19–22, 2023). https://doi.org/10.1007/978-3-031-33491-7_10 2, 5, 6, 7, 13, 40
29. arkworks contributors: `arkworks` zksnark ecosystem (2022), https://arkworks.rs 13
30. Costello, C., Hisil, H.: A simple and compact algorithm for SIDH with arbitrary degree isogenies. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, Part II. Lecture Notes in Computer Science, vol. 10625, pp. 303–329. Springer, Cham, Switzerland, Hong Kong, China (Dec 3–7, 2017). https://doi.org/10.1007/978-3-319-70697-9_11 6, 16
31. Costello, C., Longa, P., Naehrig, M.: Efficient algorithms for supersingular isogeny Diffie-Hellman. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 572–601. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53018-4_21 2, 6, 7, 13, 16
32. Costello, C., Smith, B.: Montgomery curves and their arithmetic - the case of large characteristic fields. Journal of Cryptographic Engineering **8**(3), 227–240 (Sep 2018). https://doi.org/10.1007/s13389-017-0157-6 9, 18

33. Dartois, P., Eriksen, J.K., Fouotsa, T.B., Merdy, A.H.L., Invernizzi, R., Robert, D., Rueger, R., Vercauteren, F., Wesolowski, B.: PEGASIS: Practical effective class group action using 4-dimensional isogenies. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology – CRYPTO 2025, Part I. Lecture Notes in Computer Science, vol. 16000, pp. 67–99. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 17–21, 2025). https://doi.org/10.1007/978-3-032-01855-7_3 2
34. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: New dimensions in cryptography. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part I. Lecture Notes in Computer Science, vol. 14651, pp. 3–32. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58716-0_1 2, 4, 22
35. Dartois, P., Maino, L., Pope, G., Robert, D.: An algorithmic approach to (2, 2)-isogenies in the theta model and applications to isogeny-based cryptography. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 15486, pp. 304–338. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_10 3, 7, 8, 9, 11, 19, 20, 21, 22, 23, 24
36. De Feo, L.: Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017) 9
37. De Feo, L., Dobson, S., Galbraith, S.D., Zobernig, L.: SIDH proof of knowledge. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology – ASIACRYPT 2022, Part II. Lecture Notes in Computer Science, vol. 13792, pp. 310–339. Springer, Cham, Switzerland, Taipei, Taiwan (Dec 5–9, 2022). https://doi.org/10.1007/978-3-031-22966-4_11 2, 4, 5, 8, 30
38. De Feo, L., Fouotsa, T.B., Kutas, P., Leroux, A., Merz, S.P., Panny, L., Wesolowski, B.: SCALLOP: Scaling the CSI-FiSh. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 13940, pp. 345–375. Springer, Cham, Switzerland, Atlanta, GA, USA (May 7–10, 2023). https://doi.org/10.1007/978-3-031-31368-4_13 2
39. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 64–93. Springer, Cham, Switzerland, Daejeon, South Korea (Dec 7–11, 2020). https://doi.org/10.1007/978-3-030-64837-4_3 2, 4
40. De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the Deuring correspondence - towards practical and secure SQISign signatures. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 659–690. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_23 30
41. De Feo, L., Meyer, M.: Threshold schemes from isogeny assumptions. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 12111, pp. 187–212. Springer, Cham, Switzerland, Edinburgh, UK (May 4–7, 2020). https://doi.org/10.1007/978-3-030-45388-6_7 2
42. Delpech de Saint Guilhem, C., Pedersen, R.: New proof systems and an OPRF from CSIDH. In: Tang, Q., Teague, V. (eds.) PKC 2024: 27th International Conference on Theory and Practice of Public Key Cryptography, Part III. Lecture Notes in Computer Science, vol. 14603, pp. 217–251. Springer, Cham, Switzerland, Sydney,

NSW, Australia (Apr 15–17, 2024). https://doi.org/10.1007/978-3-031-57725-3_82
43. Duparc, M.: Superglue: Fast formulae for (2,2)-gluing isogenies. In: Hanaoka, G., Yang, B.Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 372–400. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_12 8, 9, 20, 21, 24
44. El Housni, Y.: Pairings in rank-1 constraint systems. In: Tibouchi, M., Wang, X. (eds.) ACNS 2023: 21st International Conference on Applied Cryptography and Network Security, Part I. Lecture Notes in Computer Science, vol. 13905, pp. 339–362. Springer, Cham, Switzerland, Kyoto, Japan (Jun 19–22, 2023). https://doi.org/10.1007/978-3-031-33488-7_13 13, 30
45. Esgin, M.F., Steinfeld, R., Zhao, R.K.: MatRiCT$^+$: More efficient post-quantum private blockchain payments. In: 2022 IEEE Symposium on Security and Privacy. pp. 1281–1298. IEEE Computer Society Press, San Francisco, CA, USA (May 22–26, 2022). https://doi.org/10.1109/SP46214.2022.9833655 2
46. Fischlin, M.: Round-optimal composable blind signatures in the common reference string model. In: Dwork, C. (ed.) Advances in Cryptology – CRYPTO 2006. Lecture Notes in Computer Science, vol. 4117, pp. 60–77. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2006). https://doi.org/10.1007/11818175_4 4
47. Fouotsa, T.B., Moriya, T., Petit, C.: M-SIDH and MD-SIDH: Countering SIDH attacks by masking information. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 282–309. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_10 2, 4
48. Fouotsa, T.B., Petit, C.: SHealS and HealS: Isogeny-based PKEs from a key validation method for SIDH. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021, Part IV. Lecture Notes in Computer Science, vol. 13093, pp. 279–307. Springer, Cham, Switzerland, Singapore (Dec 6–10, 2021). https://doi.org/10.1007/978-3-030-92068-5_10 30
49. Fouotsa, T.B., Petit, C.: A new adaptive attack on SIDH. In: Galbraith, S.D. (ed.) Topics in Cryptology – CT-RSA 2022. Lecture Notes in Computer Science, vol. 13161, pp. 322–344. Springer, Cham, Switzerland, Virtual Event (Mar 1–2, 2022). https://doi.org/10.1007/978-3-030-95312-6_14 30
50. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) Advances in Cryptology – CRYPTO'99. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 1999). https://doi.org/10.1007/3-540-48405-1_34 30
51. Galbraith, S.D., Lai, Y.F.: Attack on SHealS and HealS: The second wave of GPST. In: Cheon, J.H., Johansson, T. (eds.) Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022. pp. 399–421. Springer, Cham, Switzerland, Virtual Event (Sep 28–30, 2022). https://doi.org/10.1007/978-3-031-17234-2_19 4, 30
52. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the security of supersingular isogeny cryptosystems. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 63–91. Springer Berlin Heidelberg, Germany, Hanoi, Vietnam (Dec 4–8, 2016). https://doi.org/10.1007/978-3-662-53887-6_3 4, 8, 30

53. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 626–645. Springer Berlin Heidelberg, Germany, Athens, Greece (May 26–30, 2013). https://doi.org/10.1007/978-3-642-38348-9_37 6
54. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge, UK (2004). https://doi.org/10.1017/CBO9780511721656 4
55. Hanzlik, L.: Non-interactive blind signatures for random messages. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 722–752. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_25 4, 6
56. Hanzlik, L., Lai, Y.F., Mula, M., Paracucchi, E., Slamanig, D., Tang, G.: Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum group actions. In: Hanaoka, G., Yang, B.Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 35–69. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_2 2
57. den Hollander, T., Kleine, S., Mula, M., Slamanig, D., Spindler, S.A.: More efficient isogeny proofs of knowledge via canonical modular polynomials. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology – CRYPTO 2025, Part I. Lecture Notes in Computer Science, vol. 16000, pp. 131–166. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 17–21, 2025). https://doi.org/10.1007/978-3-032-01855-7_5 2, 6, 7, 13
58. den Hollander, T., Levin, S., Mula, M., Pedersen, R., Slamanig, D., Spindler, S.A.: SPRINT: New isogeny proofs of knowledge and isogeny-based signatures. Cryptology ePrint Archive, Paper 2026/364 (2026), https://eprint.iacr.org/2026/364 2, 4, 5
59. Hopwood, D.E., Bowe, S., Hornby, T., Wilcox, N.: Zcash protocol specification, version 2025.6.3-51-gb25176 [overwinter+sapling]. https://zips.z.cash/protocol/sapling.pdf (Feb 2026), electric Coin Company 13, 17
60. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011. pp. 19–34. Springer Berlin Heidelberg, Germany, Tapei, Taiwan (Nov 29 – Dec 2 2011). https://doi.org/10.1007/978-3-642-25405-5_2 4, 6, 30
61. Kani, E.: The existence of curves of genus two with elliptic differentials. Journal of Number Theory **64**(1), 130–161 (1997). https://doi.org/10.1006/jnth.1997.2105 7, 19
62. Katsumata, S., Lai, Y.F., LeGrow, J.T., Qin, L.: CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023, Part III. Lecture Notes in Computer Science, vol. 14083, pp. 729–761. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 20–24, 2023). https://doi.org/10.1007/978-3-031-38548-3_24 2
63. Kim, H., Hong, S., Kim, S., Lee, S.: INKE: Fast isogeny-based PKE using intermediate curves. Cryptology ePrint Archive, Report 2025/1458 (2025), https://eprint.iacr.org/2025/1458 7
64. 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: 28th International Conference on Theory and Practice of Public Key Cryptography, Part III. Lecture Notes in Computer Science, vol. 15676, pp. 265–299. Springer, Cham, Switzerland, Røros, Norway (May 12–15, 2025). https://doi.org/10.1007/978-3-031-91826-1_9 3, 9, 19
65. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM Journal on Computing **35**(1), 170–188 (2005) 2
66. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In: 8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013). vol. 22, pp. 20–34. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2013) 2
67. Lai, Y.F., Galbraith, S.D., Delpech de Saint Guilhem, C.: Compact, efficient and UC-secure isogeny-based oblivious transfer. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021, Part I. Lecture Notes in Computer Science, vol. 12696, pp. 213–241. Springer, Cham, Switzerland, Zagreb, Croatia (Oct 17–21, 2021). https://doi.org/10.1007/978-3-030-77870-5_8 2
68. Leroux, A.: Verifiable random function from the deuring correspondence and higher dimensional isogenies. In: Fehr, S., Fouque, P.A. (eds.) Advances in Cryptology – EUROCRYPT 2025, Part VII. Lecture Notes in Computer Science, vol. 15607, pp. 167–194. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91098-2_7 2, 3, 7, 27
69. Levin, S., Pedersen, R.: Faster proofs and VRFs from isogenies. Cryptology ePrint Archive, Report 2024/1626 (2024), https://eprint.iacr.org/2024/1626 7, 13
70. Levin, S., Pedersen, R.: Faster proofs and VRFs from isogenies. In: Hanaoka, G., Yang, B.Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 307–340. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_10 2, 6
71. Lim, D., Ti, Y.B.: Adaptive attack on static POKÉ keys. Cryptology ePrint Archive, Report 2025/1541 (2025), https://eprint.iacr.org/2025/1541 4, 8, 30
72. Lund, C., Fortnow, L., Karloff, H.J., Nisan, N.: Algebraic methods for interactive proof systems. In: 31st Annual Symposium on Foundations of Computer Science. pp. 2–10. IEEE Computer Society Press, St. Louis, MO, USA (Oct 22–24, 1990). https://doi.org/10.1109/FSCS.1990.89518 2
73. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022, Part II. Lecture Notes in Computer Science, vol. 13508, pp. 71–101. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 15–18, 2022). https://doi.org/10.1007/978-3-031-15979-4_3 2
74. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 448–471. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_16 4, 19
75. Mokrani, Y., Jao, D.: Zero-knowledge proofs for SIDH variants with masked degree or torsion. Cryptology ePrint Archive, Report 2024/056 (2024), https://eprint.iacr.org/2024/056 2, 4, 5, 8
76. Moriya, T., Onuki, H., Xu, M., Zhou, G.: Adaptive attacks against FESTA without input validation or constant-time implementation. In: Saarinen, M.J., Smith-Tone, D. (eds.) Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II. pp. 3–19. Springer, Cham, Switzerland, Oxford, UK (Jun 12–14, 2024). https://doi.org/10.1007/978-3-031-62746-0_1 4

77. Moriya, T., Stopar, M.: LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram. In: Hanaoka, G., Yang, B.Y. (eds.) Advances in Cryptology – ASIACRYPT 2025, Part IV. Lecture Notes in Computer Science, vol. 16248, pp. 275–306. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_9 30, 31
78. Mumford, D.: On the Equations Defining Abelian Varieties. I. Inventiones Mathematicae **1** (12 1966). https://doi.org/10.1007/BF01389737 11
79. Nakagawa, K., Onuki, H.: QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part V. Lecture Notes in Computer Science, vol. 14924, pp. 75–106. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68388-6_4 2, 4, 7
80. Nakagawa, K., Onuki, H., Castryck, W., Chen, M., Invernizzi, R., Lorenzon, G., Vercauteren, F.: SQIsign2D-East: A new signature scheme using 2-dimensional isogenies. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. Lecture Notes in Computer Science, vol. 15486, pp. 272–303. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_9 2, 30
81. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd Annual ACM Symposium on Theory of Computing. pp. 427–437. ACM Press, Baltimore, MD, USA (May 14–16, 1990). https://doi.org/10.1145/100216.100273 4
82. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: Nearly practical verifiable computation. In: 2013 IEEE Symposium on Security and Privacy. pp. 238–252. IEEE Computer Society Press, Berkeley, CA, USA (May 19–22, 2013). https://doi.org/10.1109/SP.2013.47 6
83. Reijnders, K.: Effective pairings in isogeny-based cryptography. In: Aly, A., Tibouchi, M. (eds.) Progress in Cryptology - LATINCRYPT 2023: 8th International Conference on Cryptology and Information Security in Latin America. Lecture Notes in Computer Science, vol. 14168, pp. 109–128. Springer, Cham, Switzerland, Quito, Ecuador (Oct 3–6, 2023). https://doi.org/10.1007/978-3-031-44469-2_6 13
84. Robert, D.: Fonctions thêta et applications à la cryptographie. Ph.D. thesis, Université Henry Poincaré - Nancy 1 (2010) 11
85. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 472–503. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_17 4, 19
86. Robert, D.: Some notes on algorithms for abelian varieties. Cryptology ePrint Archive, Report 2024/406 (2024), https://eprint.iacr.org/2024/406 23
87. Setty, S.: Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020, Part III. Lecture Notes in Computer Science, vol. 12172, pp. 704–737. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 17–21, 2020). https://doi.org/10.1007/978-3-030-56877-1_25 6
88. Setty, S., Thaler, J., Wahby, R.: Customizable constraint systems for succinct arguments. Cryptology ePrint Archive, Report 2023/552 (2023), https://eprint.iacr.org/2023/552 6
89. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009) 9, 10
90. Thaler, J.: Proofs, arguments, and zero-knowledge. https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.pdf (2022) 6

A Generated Proof-Size and Runtime Reference Data

This appendix reports generated Aurora reference data for proof size and runtime used the codes in [28]. The experiments were generated on a 12th Gen Intel(R) Core(TM) i7-12700H CPU @ 2.30 GHz. The proof-size field is recorded in bytes in the generated data; throughout this section, we report proof sizes in kB by dividing the byte value by 1000.

The parameter families used in the generated data are

$$\texttt{p251x} = 2^{128}\cdot 3^{76}-1, \qquad \texttt{poke1} = 2^{129}\cdot 3^{164}\cdot 5^{18}-1, \qquad \texttt{p441x} = 2^{218}\cdot 3^{138}\cdot 37+1.$$

where `p251x` and `poke1` are the primes used in SQIsign and POKE respectively [10, 11].

The plots below are not expected to be monotone in the number of constraints. When m, n cross certain power-of-two boundaries, the evaluation domain size may increase; at the same time, the FRI parameter optimizer may select a different set of parameters, such as fewer query repetitions or a different localization schedule. As a result, a proof for a larger constraint system can sometimes be smaller than a proof for a smaller one. Optimizing the figures is out of the scope of this work.

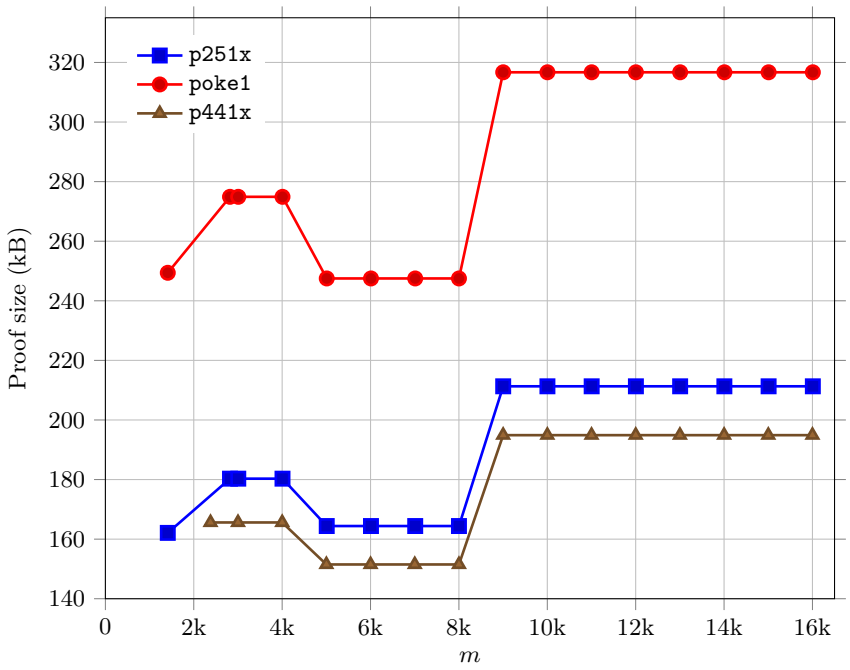


Fig. 1: Generated Aurora proof-size reference data for the three primes. Proof sizes are reported in kB by dividing the generated byte values by 1000. Note that the date points are not necessarily monotone because concrete Aurora/FRI parameter choices may change when the constraint dimensions cross implementation-relevant thresholds.

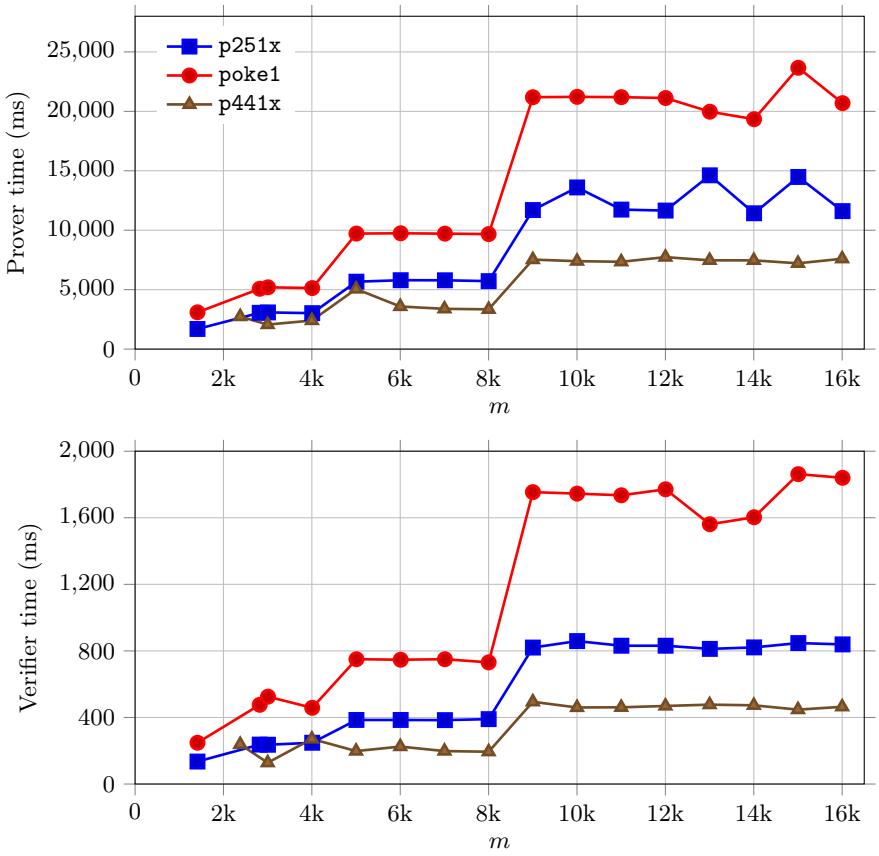


Fig. 2: Generated Aurora runtime reference data for the three parameter families. Prover and verifier times are reported in milliseconds. The `p441x` point at $m = 16000$ is approximate, following the recorded data.