A Post-Quantum Digital Signature Scheme Based on Supersingular Isogenies

Youngho Yoo[1], Reza Azarderakhsh[3], Amir Jalali[3],
David Jao[1,2], and Vladimir Soukharev[4]

[1] University of Waterloo, {yh2yoo,djao}@uwaterloo.ca
[2] evolutionQ, Inc., david.jao@evolutionq.com
[3] Florida Atlantic University, {razarderakhsh,ajalali2016}@fau.edu
[4] InfoSec Global, Inc., Vladimir.Soukharev@infosecglobal.com

Abstract. We present the first general-purpose digital signature scheme based on supersingular elliptic curve isogenies secure against quantum adversaries in the quantum random oracle model with small key sizes. This scheme is an application of Unruh's construction of non-interactive zero-knowledge proofs to an interactive zero-knowledge proof proposed by De Feo, Jao, and Plût. We implement our proposed scheme on an x86-64 PC platform as well as an ARM-powered device. We exploit the state-of-the-art techniques to speed up the computations for general C and assembly. Finally, we provide timing results for real world applications.

Keywords: Digital signatures, Isogenies, Post-quantum cryptography

1 Introduction

The security of most public-key cryptosystems in use today are based on the intractability of certain mathematical problems, namely integer factorization and discrete logarithms. However, large-scale quantum computers will be able to efficiently solve both of these problems, posing a serious threat to modern cryptography. Post-quantum cryptography is the study of classical cryptosystems that remain secure against quantum adversaries. There are several candidate approaches for building post-quantum cryptographic primitives: lattice-based, code-based, hash-based, and multivariate cryptography. Recently, cryptosystems based on supersingular elliptic curve isogenies were proposed by De Feo, Jao, and Plût [12], who gave protocols for key exchange, zero-knowledge proof of identity, and public key encryption. With small key sizes and efficient implementations [8, 17], isogenies provide a strong candidate for post-quantum key establishment.

Various isogeny-based authentication schemes have been proposed as well, such as strong designated verifier signatures [20], undeniable signatures [16], and undeniable blind signatures [19]. However, it was not known whether isogeny-based cryptography could support general authentication. In this paper, we show that this is indeed possible by constructing the first digital signature scheme based on isogenies which is strongly unforgeable under chosen message attack in the quantum random oracle model.

Our signature scheme is obtained by applying a generic transformation to the zero-knowledge proof of identity proposed in [12]. Classically, obtaining a secure digital signature from an interactive zero-knowledge proof can be achieved by applying the Fiat-Shamir transform [13]. However, its classical security proof requires certain techniques such as rewinding and reprogramming the random oracle which do not necessarily apply in the quantum setting. Quantum rewinding is possible in some restricted cases [23, 25], but it has been shown to be insecure in general [1]. Further, since random oracles model hash functions which, in a real world implementation, could be evaluated in superposition by a quantum adversary, we require *quantum* random oracles which can be queried in a superposition of possibly exponentially many states. This makes it difficult to observe an adversary's queries as measuring the input disturbs the state.

Unruh [24] recently proposed a transformation which remedies these problems to produce a secure signature in the quantum random oracle model. Its overhead is generally much larger than Fiat-Shamir – in some cases exponentially large, making the scheme impractical. Fortunately, applying it to the isogeny-based zero-knowledge proof incurs only twice as much computation as the Fiat-Shamir transform, producing a workable quantum-safe digital signature scheme with small key sizes.

Our Contributions.

- We construct the first general-purpose digital signature scheme based on supersingular elliptic curve isogenies, and prove its security in the quantum random oracle model.
- We analyze implementation aspects of our scheme and compare parameter sizes with various post-quantum signature schemes, showing that our scheme achieves very small key sizes.
- We provide source code[5] as well as performance results on x86-64 platforms and on ARM devices with assembly-optimized arithmetic.

Related Work. Independently of us, Galbraith, Petit, and Silva recently published a preprint containing two isogeny-based digital signature schemes [14]. Their second scheme, based on endomorphism rings, is completely unrelated to our work. Their first scheme, based on the De Feo, Jao, and Plût identification scheme, is conceptually identical to our scheme, but they present significant space optimizations to reduce the signature size down to $12\lambda^2$ bits (or $6\lambda^2$ if non-repudiation is not required), compared to our signature size of $69\lambda^2$ bits. However, we note that their signature size is for classical security level λ and as of this writing their posted preprint contains no signature sizes for post-quantum security, whereas our signature sizes are given in terms of post-quantum security. Moreover, their scheme may be slower, since they use a time-space tradeoff to achieve such small signature sizes. The performance of their scheme is not immediately clear, since they provide no implementation results. In this work, by contrast, we provide a complete implementation of our scheme, as well as performance results on multiple platforms and source code for reference.

[5] Source code is available at `https://github.com/yhyoo93/isogenysignature`

Outline. The rest of the paper is organized as follows. In Section 2, we give a brief preliminary on isogeny-based cryptography and describe the interactive zero-knowledge proof which will be used to construct our scheme. In Section 3, we describe Unruh's construction. We construct our isogeny-based digital signature scheme and analyze its algorithmic aspects and parameter sizes in Section 4, and give security proofs in Section 5. Performance results are provided in Section 6.

2 Isogeny-Based Cryptography

We consider elliptic curves over a finite field $\mathbb{F}_q$. An *isogeny* $\phi\colon E_1 \to E_2$ is a surjective rational map between elliptic curves which preserves the point at infinity $\mathcal{O}$. Isogenies are necessarily group homomorphisms $E_1(\mathbb{F}_q) \to E_2(\mathbb{F}_q)$ and can be identified with their kernels. This gives a one-to-one correspondence between isogenies and subgroups of the curve. Two curves E_1 and E_2 over $\mathbb{F}_q$ are isogenous if and only if $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$ [22]. The degree of an isogeny is its degree as a rational map. For separable isogenies, as are all isogenies in this paper, the degree is equal to the size of the kernel.

Every isogeny $\phi\colon E_1 \to E_2$ with degree d has a unique dual isogeny $\hat{\phi}\colon E_2 \to E_1$ of the same degree such that $\hat{\phi} \circ \phi\colon E_1 \to E_1$ is the multiplication map $P \mapsto [d]P$. The set of isogenies mapping a curve E to itself forms a ring under pointwise addition and composition, called the endomorphism ring. A curve E is *supersingular* if its endomorphism ring is isomorphic to an order in a quaternion algebra, and *ordinary* otherwise. All supersingular elliptic curves over finite fields of characteristic p are isomorphic to curves defined over $\mathbb{F}_{p^2}$.

The ℓ-torsion group of E is defined as $E[\ell] = \{P \in E(\overline{\mathbb{F}}_{p^2})\colon [\ell]P = \mathcal{O}\}$. If ℓ is coprime to p, then $E[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^2$, thus an ℓ-torsion group is generated by two elements of order ℓ.

2.1 Zero-Knowledge Proof of Identity

We use primes of the form $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$ where ℓ_A, ℓ_B are small primes (typically 2 and 3) with roughly $\ell_A^{e_A} \approx \ell_B^{e_B}$, and f is a small cofactor to ensure p is prime. The public parameters consist of a prime $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$, a supersingular curve $E(\mathbb{F}_{p^2})$ of order $(\ell_A^{e_A} \ell_B^{e_B} f)^2$, and generators P_B, Q_B of the $\ell_B^{e_B}$-torsion subgroup $E[\ell_B^{e_B}]$.

The zero-knowledge proof takes place over the diagram in Figure 1. Peggy (the prover) has a secret point S generating the kernel of the isogeny $\phi\colon E \to E/\langle S \rangle$. Her private key is S (or any generator of $\langle S \rangle$) and her public key is the curve $E/\langle S \rangle$ and the images of the public generators $\phi(P_B), \phi(Q_B)$.

$$\begin{tikzcd}
E \arrow[r, "\phi"] \arrow[d, "\psi"'] & E/\langle S \rangle \arrow[d, "\psi'"] \\
E/\langle R \rangle \arrow[r, "\phi'"] & E/\langle R, S \rangle
\end{tikzcd}$$

Fig. 1. Each arrow is labelled by the isogeny and its kernel.

In order to prove her knowledge of $\langle S\rangle$ to Vic (the verifier), Peggy chooses a random point R of order $\ell_B^{e_B}$ defining an isogeny $\psi\colon E \to E/\langle R\rangle$. Note that

$$(E/\langle S\rangle)/\langle\phi(R)\rangle = E/\langle R, S\rangle = (E/\langle R\rangle)/\langle\psi(S)\rangle$$

In other words, the diagram in Figure 1 commutes.

Peggy computes the isogenies in the diagram and sends to Vic the two non-public curves. Vic sends her a challenge bit $b \in \{0,1\}$, and Peggy reveals some of the isogenies depending on b, which Vic then verifies.

More precisely, Peggy and Vic run the following protocol:

1. – Peggy chooses a random point R of order $\ell_B^{e_B}$.
   – She computes the isogeny $\psi\colon E \to E/\langle R\rangle$.
   – She computes the isogeny $\phi'\colon E/\langle R\rangle \to E/\langle R, S\rangle$ with kernel $\langle\psi(S)\rangle$ (alternatively the isogeny $\psi'\colon E/\langle S\rangle \to E/\langle R, S\rangle$ with kernel $\langle\phi(R)\rangle$)
   – She sends the commitment com $= (E_1, E_2)$ to Vic, where $E_1 = E/\langle R\rangle$ and $E_2 = E/\langle R, S\rangle$.
2. Vic randomly chooses a challenge bit ch $\in \{0,1\}$ and sends it to Peggy.
3. Peggy sends the response resp where
   – If ch $= 0$, then resp $= (R, \phi(R))$.
   – If ch $= 1$, then resp $= \psi(S)$.
4. – If ch $= 0$, Vic verifies that R and $\phi(R)$ have order $\ell_B^{e_B}$ and generate the kernels for the isogenies $E \to E_1$ and $E/\langle S\rangle \to E_2$ respectively.
   – If ch $= 1$, Vic verifies that $\psi(S)$ has order $\ell_A^{e_A}$ and generates the kernel for the isogeny $E_1 \to E_2$.

$$\begin{tikzcd}
E \arrow[r, dashed, "\phi"] \arrow[d, thick, "\psi"'] & E/\langle S\rangle \arrow[d, thick, "\psi'"] \\
E/\langle R\rangle \arrow[r, dashed, "\phi'"] & E/\langle R, S\rangle
\end{tikzcd}$$

$b = 0$

$$\begin{tikzcd}
E \arrow[r, dashed, "\phi"] \arrow[d, dashed, "\psi"'] & E/\langle S\rangle \arrow[d, dashed, "\psi'"] \\
E/\langle R\rangle \arrow[r, thick, "\phi'"] & E/\langle R, S\rangle
\end{tikzcd}$$

$b = 1$

Fig. 2. Hidden isogenies are indicated by dashed lines. Bolded lines indicate the isogenies revealed by Peggy on challenge b. In either case, the revealed isogenies do not leak information about the secret isogeny ϕ.

To achieve λ bits of security, the prime p should be roughly 6λ bits (see Section 5) and this protocol should be run λ times. If Vic successfully verifies all λ rounds of the protocol, then Peggy has proved her identity (knowledge of the private key S) to Vic. Otherwise, Vic rejects.

3 Unruh's Construction

Unruh's construction [24] transforms an interactive zero-knowledge proof system into a non-interactive one. The construction satisfies *online extractability* which allows us to extract the witness (private key) from a successful adversary without rewinding. It also avoids the problem of determining the query inputs of the quantum random oracle by including its outputs in the proof (signature) and "inverting" them in the security proof. See [24] for the full security proof.

We fix a binary relation R. A statement x holds if there exists w such that $(x, w) \in R$. In this case, we call w a *witness* to x. In a proof system, a prover P tries to prove a statement x to a verifier V (in other words, to convince V that P knows a witness w to x). We assume that all parties have access to a quantum random oracle H which can be queried in superposition.

3.1 Sigma Protocols

A *sigma protocol* $\Sigma = ((P^1, P^2), V)$ is an interactive proof system consisting of three messages in order: a commitment $\mathrm{com} = P^1(x, w)$ made by the prover, a challenge ch chosen uniformly at random by the verifier, and the response $\mathrm{resp} = P^2(x, w, \mathrm{com}, \mathrm{ch})$ computed by the prover based on the challenge. Then V outputs $V(x, \mathrm{com}, \mathrm{ch}, \mathrm{resp})$, indicating whether they accept or reject the proof.

Let $\Sigma = (P, V)$ be a sigma protocol where $P = (P^1, P^2)$. We define the following properties of sigma protocols (from [24, §2.2]):

Completeness: If P knows a witness w to the statement x, then V accepts.

Special soundness: There exists a polynomial time extractor E_Σ such that, given any pair of valid interactions $(\mathrm{com}, \mathrm{ch}, \mathrm{resp})$ and $(\mathrm{com}, \mathrm{ch}', \mathrm{resp}')$ with $\mathrm{ch} \neq \mathrm{ch}'$ that V accepts, E_Σ can compute a witness w such that $(x, w) \in R$.

Honest-verifier zero-knowledge (HVZK): There is a polynomial time simulator S_Σ with outputs of the form $(\mathrm{com}, \mathrm{ch}, \mathrm{resp})$ that are indistinguishable from valid interactions between a prover and an honest verifier by any quantum polynomial time algorithm.

Note that the isogeny-based zero-knowledge proof of identity from the previous section is a sigma protocol. We will show in Section 5 that it satisfies all three properties listed above.

3.2 Non-interactive Proof Systems

A *non-interactive proof system* consists of two algorithms: a prover $P(x, w)$ outputting a proof π of the statement x (which has witness w), and a verifier $V(x, \pi)$ outputting whether it accepts or rejects the proof π of x.

For a non-interactive proof system (P, V), we define the following properties (from [24, §2.1]):

Completeness: If $(x, w) \in R$, then V accepts the proof $\pi = P(x, w)$.

Zero-knowledge (NIZK): There exists a polynomial time simulator S such that, given the ability to program the random oracle, S can output proofs indistinguishable from those produced by P by any quantum polynomial time algorithm.

The simulator is modeled by two algorithms $S = (S_{\mathrm{init}}, S_P)$, where S_{init} outputs an initial circuit H simulating a quantum random oracle, and S_P is a stateful algorithm which may reprogram H and produce proofs using H.

Simulation-sound online-extractability: (with respect to a simulator $S = (S_{\text{init}}, S_P)$) There exists a polynomial time extractor E such that, if a quantum polynomial time algorithms $\mathcal{A}$ with quantum access to $H \leftarrow S_{\text{init}}$ and classical access to the prover S_P outputs a new valid proof of a statement x, then E can compute (extract) a witness w of x.

Remark 1. Granting $\mathcal{A}$ classical access to the simulated prover S_P is analogous to granting the adversary access to a classical signing oracle in a chosen message attack in the context of signatures. We could allow $\mathcal{A}$ to have *quantum* access to S_P, corresponding to a *quantum* chosen message attack as defined in [6]. We do not know whether Unruh's construction remains secure under this relaxation.

3.3 Unruh's Construction

Unruh's construction transforms a sigma protocol Σ into a non-interactive proof system (P_{OE}, V_{OE}) so that, if Σ satisfies completeness, special soundness, and HVZK, then the result is a complete NIZK proof system with simulation-sound online extractability.

Suppose we have a sigma protocol $\Sigma = (P_\Sigma, V_\Sigma)$ with $P_\Sigma = (P^1_\Sigma, P^2_\Sigma)$, where there are c possible challenges in the challenge domain N_{ch} and the parties want to run the protocol t times, where t depends on the security parameter λ (in our signature scheme we will have $N_{ch} = \{0, 1\}$, $c = 2$, and $t = 2\lambda$). Let G, H be quantum random oracles, where G has the same domain and range. We define a non-interactive proof system (P_{OE}, V_{OE}) where P_{OE} and V_{OE} are given by Algorithms 1 and 2 respectively.

The idea is to simulate the interaction in Σ by setting the challenge $J = J_1 \| \ldots \| J_t$ as the output of the random function H. However, instead of evaluating H on the commitments $(\text{com}_i)_i$ alone as in the Fiat-Shamir transform, we also include the hashes $h_{i,j} = G(\text{resp}_{i,j})$ of the responses $\text{resp}_{i,j}$ to each possible challenge $\text{ch}_{i,j}$, for each commitment com_i. Then the produced proof consists of the commitments, an ordering of all possible challenges, hashed responses to the corresponding challenges, and the responses to the challenges given by $J_1 \| \ldots \| J_t$. The verifier can then take the data to reproduce $J_1 \| \ldots \| J_t$, check that the data was produced properly, and verify the responses $(\text{resp}_{i,J_i})_i$ for each round of Σ.

The main theorem of [24] proves that this construction is secure in the quantum oracle model. Its proof is based on the fact that the random oracle G is indistinguishable from a random permutation, and replaces G with an efficiently invertible function (a random polynomial of high degree) which is unnoticeable by any quantum polynomial time adversary. This allows the hashes to be inverted to obtain the hidden responses in the adversary's forged proof.

Theorem 1 ([24, Corollary 19]). *If Σ satisfies completeness, special soundness, and HVZK, then (P_{OE}, V_{OE}) is a complete non-interactive zero-knowledge proof system with simulation-sound online extractability in the quantum random oracle model.*

Algorithm 1 Prover: P_{OE} on input (x, w)

```
// Create $t \cdot c$ proofs and hash each response
for $i = 1$ to $t$ do
    $\mathrm{com}_i \leftarrow P_\Sigma^1(x, w)$
    for $j = 1$ to $c$ do
        $\mathrm{ch}_{i,j} \leftarrow_R N_{ch} \setminus \{\mathrm{ch}_{i,1}, \ldots, \mathrm{ch}_{i,j-1}\}$
        $\mathrm{resp}_{i,j} \leftarrow P_\Sigma^2(x, w, \mathrm{com}_i, \mathrm{ch}_{i,j})$
        $h_{i,j} \leftarrow G(\mathrm{resp}_{i,j})$

// Get challenge by hashing
$J_1 \| \ldots \| J_t \leftarrow H(x, (\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j})$

// Return proof
return $\pi \leftarrow ((\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j}, (\mathrm{resp}_{i,J_i})_i)$
```

Algorithm 2 Verifier: V_{OE} on input (x, π), where $\pi = ((\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j}, (\mathrm{resp}_{i,J_i})_i)$

```
// Compute the challenge hash
$J_1 \| \ldots \| J_t \leftarrow H(x, (\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j})$

for $i = 1$ to $t$ do
    check $\mathrm{ch}_{i,1}, \ldots, \mathrm{ch}_{i,m}$ pairwise distinct
    check $h_{i,J_i} = G(\mathrm{resp}_i)$
    check $V_\Sigma(x, \mathrm{com}_i, \mathrm{ch}_{i,J_i}, \mathrm{resp}_i) = 1$
if all checks succeed then
    return 1
```

3.4 Signatures from Non-interactive Zero-Knowledge Proofs

A *digital signature* scheme consists of three algorithms:

- Keygen(λ): takes a security parameter λ and outputs a key pair $(\mathrm{pk}, \mathrm{sk})$.
- Sign(sk, m): signs the message m using sk, outputting a signature σ.
- Verify(pk, m, σ): takes the public key of the claimed signer and verifies the signature σ on the message m.

A digital signature scheme is *strongly unforgeable under chosen message attack (SUF-CMA)* if, for any quantum polynomial time adversary $\mathcal{A}$ with classical access to the signing oracle $\mathrm{sig} \colon m \mapsto \mathrm{Sign}(\mathrm{sk}, m)$, $\mathcal{A}$ cannot produce a new valid message-signature pair with non-negligible probability.

Suppose we have a function Keygen generating a public-private key pair $(\mathrm{pk}, \mathrm{sk})$ such that no quantum polynomial-time algorithm can recover a valid sk from pk with non-negligible probability. A proof of identity can be viewed as proving the statement $x = \mathrm{pk}$ with witness $w = \mathrm{sk}$, where $(x, w) \in R$ if and only if (x, w) is a valid key pair that can be generated by Keygen.

In this sense, a digital signature is basically a non-interactive zero-knowledge proof of identity, except that we need to incorporate a specific message into each proof (signature). This is done by including the message as a part of the statement $x = (\mathrm{pk}, m)$, and the relation R ignores the message m; ie. $((\mathrm{pk}, m), w) \in R$

if and only if (pk, w) is a valid key pair. Thus, from a NIZK proof of identity (P, V), we obtain a digital signature scheme $\mathcal{DS} = (\mathrm{Keygen}, \mathrm{Sign}, \mathrm{Verify})$ where $\mathrm{Sign}(\mathrm{sk}, m) = P((\mathrm{pk}, m), \mathrm{sk})$ and $\mathrm{Verify}(\mathrm{pk}, m, \sigma) = V((\mathrm{pk}, m), \sigma)$.

Theorem 2 ([24, Theorem 23]). *If (P, V) is a NIZK proof of identity satisfying simulation-sound online-extractability, then the signature scheme $\mathcal{DS}$ above is SUF-CMA in the quantum random oracle model.*

Proof (sketch). Since (P, V) is zero-knowledge, there is a polynomial time simulator that can indistinguishably simulate proofs (signatures) by reprogramming the random oracle. If an adversary can forge a new valid message-signature pair by querying the simulator, then by simulation-sound online-extractability, we can efficiently extract a witness sk. □

4 Isogeny-Based Digital Signature

We propose our isogeny-based digital signature scheme based on the results from previous sections. Let Σ denote the isogeny-based zero-knowledge proof of identity described in Section 2.1. Applying Unruh's construction to Σ, we obtain a non-interactive proof of identity (P_{OE}, V_{OE}), from which we get a digital signature scheme:

Public Parameters. We have the same public parameters as in Σ: a prime $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$, a supersingular curve E of cardinality $(\ell_A^{e_A} \ell_B^{e_B})^2$ over $\mathbb{F}_{p^2}$, and generators (P_B, Q_B) of the torsion group $E[\ell_B^{e_B}]$.

Key Generation. To generate keys, select a random point S of order $\ell_A^{e_A}$, compute the isogeny $\phi\colon E \to E/\langle S \rangle$, and output the key pair $(\mathrm{pk}, \mathrm{sk})$ where $\mathrm{pk} = (E/\langle S \rangle, \phi(P_B), \phi(Q_B))$ and $\mathrm{sk} = S$.

Signing. To sign a message m, set $\mathrm{Sign}(\mathrm{sk}, m) = P_{OE}((\mathrm{pk}, m), \mathrm{sk})$.

Verification. To verify the signature σ of message m, set $\mathrm{Verify}(\mathrm{pk}, m, \sigma) = V_{OE}((\mathrm{pk}, m), \sigma)$.

Algorithms 3, 4, and 5 give explicit descriptions of $(\mathrm{Keygen}, \mathrm{Sign}, \mathrm{Verify})$.

4.1 Algorithmic Aspects

We describe some of the lower-level algorithmic aspects of our signature scheme. Full details can be found in [12, 8]. For efficiency in our implementation, we mainly follow [8] for their algorithms and representations of parameters.

Sampling Torsion Points. Let P, Q be fixed generators for the torsion group $E[\ell^e]$. To sample a point R of order ℓ^e, we choose $m, n \in \mathbb{Z}/\ell^e\mathbb{Z}$, not both divisible by ℓ, and compute $R = [m]P + [n]Q$. Since R and $[k]R$ generate the same subgroup $\langle R \rangle = \langle [k]R \rangle$ for any k not divisible by ℓ, we can replace R by $P + [m^{-1}n]Q$ or $[mn^{-1}]P + Q$, depending on which coefficient is coprime to ℓ.

For simplicity, we ignore the coefficient of P as in [8] where it is shown that, for certain pairs of generators P, Q related by distortion maps, each value of $n \in \{1, 2, \ldots, \ell^{e-1} - 1\}$ gives a point $R = P + [\ell n]Q$ of full order ℓ^e generating distinct subgroups. Note that this procedure samples from $\ell^{e-1} - 1$ possible subgroups.

Algorithm 3 Keygen(λ)

```
Pick a random point $S$ of order $\ell_A^{e_A}$
Compute the isogeny $\phi\colon E \to E/\langle S\rangle$
$\mathrm{pk} \leftarrow (E/\langle S\rangle, \phi(P_B), \phi(Q_B))$
$\mathrm{sk} \leftarrow S$
return $(\mathrm{pk}, \mathrm{sk})$
```

Algorithm 4 Sign(sk, m)

```
for $i = 1$ to $2\lambda$ do
    Pick a random point $R$ of order $\ell_B^{e_B}$
    Compute the isogeny $\psi\colon E \to E/\langle R\rangle$
    Compute either $\phi'\colon E/\langle R\rangle \to E/\langle R, S\rangle$ or $\psi'\colon E/\langle S\rangle \to E/\langle R, S\rangle$
    $(E_1, E_2) \leftarrow (E/\langle R\rangle, E/\langle R, S\rangle)$
    $\mathrm{com}_i \leftarrow (E_1, E_2)$
    $\mathrm{ch}_{i,0} \leftarrow_R \{0, 1\}$
    $(\mathrm{resp}_{i,0}, \mathrm{resp}_{i,1}) \leftarrow ((R, \phi(R)), \psi(S))$
    if $\mathrm{ch}_{i,0} = 1$ then
        $\mathrm{swap}(\mathrm{resp}_{i,0}, \mathrm{resp}_{i,1})$
    $h_{i,j} \leftarrow G(\mathrm{resp}_{i,j})$
$J_1 \| \ldots \| J_{2\lambda} \leftarrow H(\mathrm{pk}, m, (\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j})$
return $\sigma \leftarrow ((\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j}, (\mathrm{resp}_{i,J_i})_i)$
```

Algorithm 5 Verify(pk, m, σ)

```
$J_1 \| \ldots \| J_{2\lambda} \leftarrow H(m, x, (\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j})$
for $i = 1$ to $2\lambda$ do
    check $h_{i,J_i} = G(\mathrm{resp}_{i,J_i})$
    if $\mathrm{ch}_{i,J_i} = 0$ then
        Parse $(R, \phi(R)) \leftarrow \mathrm{resp}_{i,J_i}$
        check $R, \phi(R)$ have order $\ell_B^{e_B}$
        check $R$ generates the kernel of the isogeny $E \to E_1$
        check $\phi(R)$ generates the kernel of the isogeny $E/\langle S\rangle \to E_2$
    else
        Parse $\psi(S) \leftarrow \mathrm{resp}_{i,J_i}$
        check $\psi(S)$ has order $\ell_A^{e_A}$
        check $\psi(S)$ generates the kernel of the isogeny $E_1 \to E_2$
if all checks succeed then
    return 1
```

Computing Isogenies. Isogenies of degree ℓ^e can be computed by composing e isogenies of degree ℓ. Isogeny computation is by far the most expensive process in isogeny-based systems. Detailed analysis on optimizing isogeny computation can be found in [8, 12].

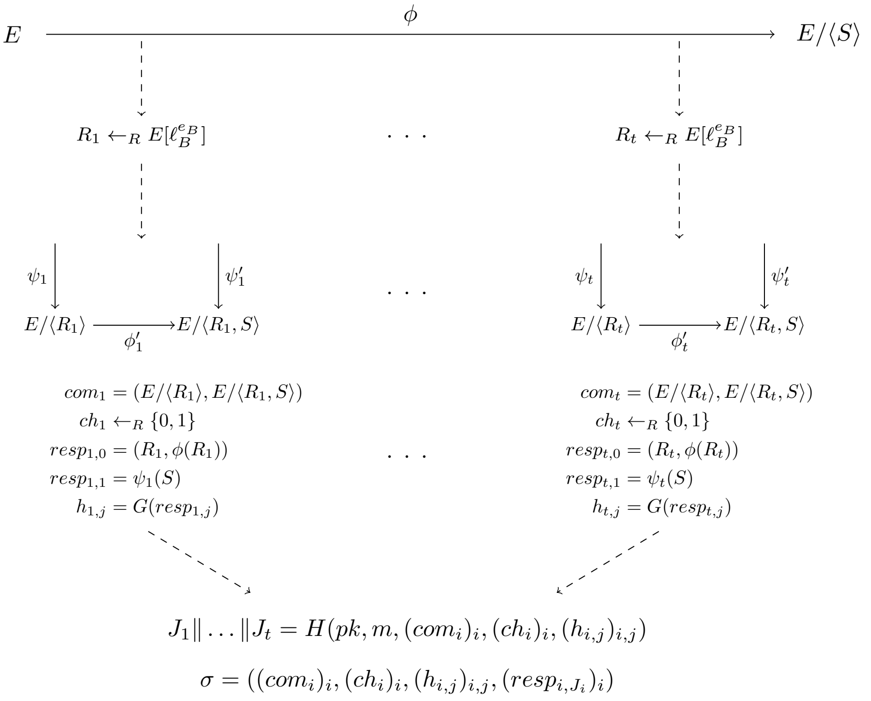
E
ϕ
$E/\langle S\rangle$
$R_1 \leftarrow_R E[\ell_B^{e_B}]$
$\cdots$
$R_t \leftarrow_R E[\ell_B^{e_B}]$
ψ_1
ψ_1'
$E/\langle R_1\rangle$
$E/\langle R_1, S\rangle$
ϕ_1'
ψ_t
ψ_t'
$E/\langle R_t\rangle$
$E/\langle R_t, S\rangle$
ϕ_t'
$com_1 = (E/\langle R_1\rangle, E/\langle R_1, S\rangle)$
$ch_1 \leftarrow_R \{0,1\}$
$resp_{1,0} = (R_1, \phi(R_1))$
$resp_{1,1} = \psi_1(S)$
$h_{1,j} = G(resp_{1,j})$
$com_t = (E/\langle R_t\rangle, E/\langle R_t, S\rangle)$
$ch_t \leftarrow_R \{0,1\}$
$resp_{t,0} = (R_t, \phi(R_t))$
$resp_{t,1} = \psi_t(S)$
$h_{t,j} = G(resp_{t,j})$
$J_1\|\ldots\|J_t = H(pk, m, (com_i)_i, (ch_i)_i, (h_{i,j})_{i,j})$
$\sigma = ((com_i)_i, (ch_i)_i, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$

Fig. 3. An illustration of the signing algorithm running t rounds of the isogeny-based zero-knowledge proof. For each ZKP round, the signer chooses a random full-order $\ell_B^{e_B}$-torsion point R and computes the relevant data in the ZKP and hashes of the responses (note that these can run in parallel and be precomputed before the message m is known). The collective data is then hashed together with the message to obtain the challenge bits $J_1\|\ldots\|J_t$. The signature σ contains the data necessary for the verifier to compute $J_1\|\ldots\|J_t$, and the responses to the challenges..

Representing of Curves and Points. We use projective coordinates for both points and curve coefficients as in [8] to reduce the number of field inversions. The curves in our system are isomorphic to Montgomery curves which have the form $E_{(A,B)} : By^2 = x^3 + Ax^2 + x$. The Kummer line on a Montgomery curve, which identifies each point $(X : Y : Z)$ with its inverse $(X : -Y : Z)$, has efficient point arithmetic and allows us to disregard the Y coordinate in our computations. This allows us to represent points by just one field element X/Z in $\mathbb{F}_{p^2}$. However, to compute linear combinations we require an additional x-coordinate of $P - Q$ to perform *differential addition.* We thus include the x-coordinate of $\phi(P_B - Q_B)$ as part of the public key. Isogeny computations are unaffected because a point R and its inverse $-R$ generate the same subgroup.

In the Montgomery form, it turns out that there are only two isomorphism classes of Montgomery curves for a given coefficient value A, and they have the same Kummer line. So the B coefficient also does not affect our computations, and curves can also be represented by one field element for their A-coordinate.

4.2 Parameter Sizes

Recall that our primes have the form $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ with roughly $\ell_A^{e_A} \approx \ell_B^{e_B}$. Note that we require primes of bitlength 6λ in order to achieve λ bits of post-quantum security (see §5), so we have $\ell_A^{e_A} \approx \ell_B^{e_B} \approx 2^{3\lambda}$.

Since all supersingular curves are defined over $\mathbb{F}_{p^2}$, each field element requires 12λ bits. Our curves are represented in Montgomery form $By^2 = x^3 + Ax^2 + x$ where the A-coefficient suffices for isogeny computations. Similarly, a point on the Kummer line can be represented by their X-coordinate. In both cases, we need one field element, requiring 12λ bits.

Compression. Azarderakhsh et al. [2] showed that torsion points can be compressed by representing them by their coefficients with respect to a deterministically generated basis (computing 2-dimensional discrete log is polynomial-time for smooth curves). Their implementation was however very slow. Recent work by Costello et al. [7] proposed new algorithms accelerating the previous work by more than an order of magnitude and further reduce public key sizes. Their improved compression algorithm runs roughly as fast as a round of the ZKP protocol.

A torsion point used to generate a subgroup can be represented by one coefficient since we can always normalize the coefficient of one generator. Compressing two generators of a torsion group requires three coefficients to keep track of their relation when computing linear combinations. Each coefficient requires roughly 3λ bits.

We can apply the compression to our signature scheme in two ways: first to the public key and second to the responses $\psi(S)$ for the rounds where $ch = 1$. The private key and the other responses $(R, \phi(R))$ are generated using a 3λ-bit coefficient and as such do not require additional computation for compression.

Public Keys. The public key has the form $\mathrm{pk} = (a, x(P_B), x(Q_B), x(P_B - Q_B))$, where a denotes the A-coefficient of the public curve $E/\langle S\rangle$. These four field elements require 48λ bits of storage.

We can compress the public key significantly by compressing the torsion basis $(\phi(P_B), \phi(Q_B))$, requiring three 3λ-bit coefficients. Moreover, the X-coordinate of $\phi(P_B - Q_B)$ is no longer required since the full coordinates of $\phi(P_B)$ and $\phi(Q_B)$ can be recovered from their compressed coefficients. Thus the compressed public key requires 12λ bits for the curve and 9λ bits for the generators, for a total of 21λ bits.

Private Keys. The private key S can be stored as a single coefficient n with respect to a $\ell_A^{e_A}$-torsion basis P_A, Q_A (ie. $S = P_A + [n]Q_A$), requiring 3λ bits.

Signatures. The signature contains $(\mathrm{com}_i, \mathrm{ch}_{i,j}, h_{i,j}, \mathrm{resp}_{i,J_i})$ for each round i of the ZKP protocol. Each commitment contains two curves (E_1, E_2), each requiring one field element. We need one bit to indicate the first challenge bit $\mathrm{ch}_{i,0}$. We do not need to send $\mathrm{ch}_{i,1}$ since $\mathrm{ch}_{i,1} = 1 - \mathrm{ch}_{i,0}$. The hash $h_{i,j} = G(\mathrm{resp}_{i,j})$ should have bitlength 3λ (this will be justified in §5.2). Note that we do not need to send h_{i,J_i} since it can be computed from resp_{i,J_i}.

The response has a different length depending on the challenge bit J_i. If $J_i = 0$, the response $(R, \phi(R))$ can be represented by their coefficients with respect to the public bases at no additional computational cost, requiring only 3λ bits. If $J_i = 1$, the response $\psi(S)$ requires 12λ bits as a field element. With compression, $\psi(S)$ can be represented in 3λ bits.

In total, each round of the ZKP requires roughly $24\lambda + 1 + 3\lambda + \frac{3\lambda+12\lambda}{2} \approx 34.5\lambda$ bits on average without compression, and roughly 30λ bits on average with compression. Although λ rounds of the ZKP sufficed for λ bits of post-quantum security, the signature requires 2λ rounds of the ZKP protocol due to the challenge hash being vulnerable to Grover's algorithm [15] (see §5.3). So the entire signature has size roughly $69\lambda^2$ ($60\lambda^2$ compressed) bits on average.

For instance, to achieve 128 bits of post-quantum security, our signature scheme requires $48\lambda = 6144$ bits (768 bytes) for the public key (336 bytes compressed), $3\lambda = 384$ bits (48 bytes) for the private key, and $69\lambda^2 = 1,130,496$ bits (141,312 bytes) for the signature (122,880 bytes compressed) on average.

Comparison. We compare our parameter sizes with various post-quantum signature schemes: the stateless hash-based signature SPHINCS-256 [4], a code-based signature based on Niederreiter's variant of the McEliece cryptosystem [5, 9], a lattice-based signature BLISS [11], a recent ring-LWE-based signature TESLA# [3], and the multivariate polynomial-based Rainbow signature [10, 18].

Table 1. Comparison of parameter sizes (in bytes) with various post-quantum signature schemes at the quantum 128-bit security level.

Scheme	Public-key size	Private-key size	Signature size
Hash-based	1,056	1,088	41,000
Code-based	192,192	1,400,288	370
Lattice-based	7,168	2,048	5,120
Ring-LWE-based	7,168	4,608	3,488
Multivariate-based	99,100	74,000	424
Isogeny-based	768	48	141,312
Compressed	336	48	122,880

It is clear from Table 1 that our isogeny-based signature achieves very small key sizes relative to the other post-quantum signature schemes. We note that the variants of the Merkle signature scheme can achieve smaller (32 byte) key sizes at the same security level, but require state management. We expect future works in isogenies to improve upon signature sizes and performance to produce more practical signatures with still compact keys.

5 Security

The security of isogeny-based cryptosystems are based on the following problems (from [12, §5]), which are believed to be intractable even for quantum computers.

Computational Supersingular Isogeny (CSSI) problem: Let $\phi_A : E_0 \to E_A$ be an isogeny whose kernel is $\langle R_A \rangle$ where R_A is a random point with order $\ell_A^{e_A}$. Given $E_A, \phi_A(P_B), \phi_A(Q_B)$, find a generator of $\langle R_A \rangle$.

Decisional Supersingular Product (DSSP) problem: Let $\phi\colon E_0 \to E_3$ be an isogeny of degree $\ell_A^{e_A}$. Given (E_1, E_2, ϕ') sampled with probability 1/2 from one or the other of the following distributions, determine which distribution it is from.

- A random point R of order $\ell_B^{e_B}$ is chosen and $E_1 = E_0/\langle R\rangle$, $E_2 = E_3/\langle\phi(R)\rangle$, and $\phi'\colon E_1 \to E_2$ is an isogeny of degree $\ell_A^{e_A}$.
- E_1 is chosen randomly among curves of the same cardinality as E_0, and $\phi'\colon E_1 \to E_2$ is a random isogeny of degree $\ell_A^{e_A}$

The best known attack for the CSSI problem involves claw-finding algorithms using quantum walks [21] and takes $O(p^{1/6})$ time, which is optimal for a black-box claw attack [26]. Therefore it is believed that a prime with bitlength 6λ achieves λ bits of post-quantum security.

5.1 Security of the Zero-Knowledge Proof

It is proven in [12, §6.2] that our isogeny-based zero-knowledge proof of identity from §2.1 satisfies completeness, soundness, and honest-verifier zero-knowledge under the assumption that the CSSI and DSSP problems are hard. However, Unruh's construction requires *special soundness.*

Theorem 3 ([12, Theorem 6.3]). *The isogeny-based zero-knowledge proof of identity satisfies completeness, special soundness, and HVZK.*

Proof. We only prove special soundness. Suppose we are given two valid transcripts $(\mathrm{com}, 0, \mathrm{resp}_0)$ and $(\mathrm{com}, 1, \mathrm{resp}_1)$, where $\mathrm{com} = (E_1, E_2)$. Then we can use $\mathrm{resp}_0 = (R, \phi(R))$ to compute the isogeny $\psi\colon E \to E/\langle R\rangle$. Since $\mathrm{resp}_1 = \psi(S)$ is a generator of the kernel of ϕ', we can take the dual isogeny $\hat{\psi}\colon E/\langle R\rangle \to E$, and compute $\hat{\psi}(\mathrm{resp}_1)$, a generator for $\langle S\rangle$. □

$$\begin{tikzcd}
E \arrow[r, dashed, "\phi"] \arrow[d, "\psi"'] & E/\langle S\rangle \arrow[d, "\psi'"] \\
E/\langle R\rangle \arrow[r, "\phi'"'] & E/\langle R, S\rangle
\end{tikzcd}$$

Fig. 4. If ψ and ϕ' are both known, then we can recover the secret subgroup $\langle S\rangle$.

5.2 Security of the Signature

Theorem 2 implies that our isogeny-based signature scheme obtained in §4 is SUF-CMA. However, one important detail in Unruh's proof is that the quantum random oracle G must have the same domain and range for both response types, so that one can substitute G with a random polynomial and invert hashes in the security proof. In §4.2, we described compression techniques giving us a few variants of our signature scheme with a space-time tradeoff (we could compress the public key, the responses, or both), and we also took G to be a random oracle outputting hashes of bitlength $k \approx 3\lambda$. While Unruh's proof applies directly to our compressed signatures, it is invalid in our uncompressed signature scheme where the responses can have bitlength k or $4k$. In this case, the only way to apply Unruh's construction directly is to pad the shorter responses to $4k$ bits. G should then output hashes of bitlength $4k$ so that the domain and range of G are both equal to $\{0,1\}^{4k}$, increasing signature sizes by roughly $18\lambda^2$ bits.

We show by an ad-hoc argument that compression is not necessary—the uncompressed signature scheme remains secure when G outputs hashes of bitlength $k \approx 3\lambda$. Let $\mathcal{DS}_u$ denote the uncompressed signature scheme and $\mathcal{DS}_c$ denote the scheme where the responses $\psi(S)$ are compressed.

Theorem 4. *$\mathcal{DS}_c$ is SUF-CMA in the quantum random oracle model.*

Proof. Since all responses are represented by bitstrings of length k, the security of $\mathcal{DS}_c$ follows from Theorem 2. □

Theorem 5. *$\mathcal{DS}_u$ is SUF-CMA in the quantum random oracle model.*

Proof. Suppose there exists a quantum polynomial-time adversary $\mathcal{A}$ breaking the SUF-CMA security of $\mathcal{DS}_u$. We show that, given a classical signing oracle to an instance of $\mathcal{DS}_c$ with quantum random oracle $G_c \colon \{0,1\}^k \to \{0,1\}^k$, we can forge a new valid message-signature pair for $\mathcal{DS}_c$ using $\mathcal{A}$.

Suppose we are given the public key pk and a signing oracle to an instance of $\mathcal{DS}_c$ with quantum random oracles G_c and H. Let C_0, C_1 denote the set of possible responses to the challenge $ch = 0, 1$ respectively in $\mathcal{DS}_c$. Note that both sets have cardinality roughly 2^k and consist of k-bitstrings. We create an instance of $\mathcal{DS}_u$ with the same setup, except the quantum random oracle G_u is to be defined as follows.

Let U_0, U_1 denote the set of possible responses to the challenge $ch = 0, 1$ respectively in $\mathcal{DS}_u$. Then we have $C_0 = U_0$ and $|C_1| = |U_1|$, but the elements of U_1 are $4k$-bitstrings. Let $\mathcal{C} \colon U_1 \to C_1$ eenote the compression map taking the field representation of a point $\psi(S)$ in U_1 to its compressed coefficient representation in C_1. Then $\mathcal{C}$ is a bijection that can be computed efficiently both ways since the compression map is injective and its inverse just computes the linear combination. Let $G'_u \colon \{0,1\}^{4k} \to \{0,1\}^k$ be a quantum random oracle such that $G'_u(z \| x) = G_c(x)$ for all $x \in \{0,1\}^k$, where z denotes the all-zeros string of length $3k$. Define $G_u \colon \{0,1\}^{4k} \to \{0,1\}^k$ where

$$G_u(x) = \begin{cases} G'_u(z \| \mathcal{C}(x)) & \text{if } x \in U_1 \\ G'_u(\mathcal{C}^{-1}(y)) & \text{if } x = z \| y \text{ where } y \in C_1 \\ G'_u(x) & \text{otherwise} \end{cases}$$

Since G_u just permutes the inputs according to the bijection $\mathcal{C}$ (with MSB zero-padding) before applying the quantum random oracle G'_u, it follows that G_u is indistinguishable from G'_u. Hence $\mathcal{A}$ can break $\mathcal{DS}_u$ when instantiated with G_u.

We give $\mathcal{A}$ the same public key pk with quantum random oracles G_u and H. When $\mathcal{A}$ makes a signing query on a message m, we relay it to the $\mathcal{DS}_c$ signing oracle to get back a signature

$$\sigma = ((\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j}, (\mathrm{resp}_{i,J_i})_i)$$

where $J_1 \| \dots \| J_t = H(\mathrm{pk}, m, (\mathrm{com}_i)_i, (\mathrm{ch}_{i,j})_{i,j}, (h_{i,j})_{i,j})$ and $h_{i,j} = G_c(\mathrm{resp}_{i,j})$. We simply decompress all responses resp_{i,J_i} in σ where $\mathrm{ch}_{i,J_i} = 1$, and give this modified σ to $\mathcal{A}$. Since $G_u(\mathcal{C}^{-1}(y)) = G'_u(z\|y) = G_c(y)$ for all $y \in C_1$, and $G_u(x) = G_c(x)$ for all $x \in C_0$ (with MSB zero-padding of input), it follows that the $h_{i,j}$'s are still valid hashes in $\mathcal{DS}_u$ with G_u. Hence the modified σ is a valid signature for m in $\mathcal{DS}_u$.

Therefore we can answer $\mathcal{A}$'s signing oracle queries so that $\mathcal{A}$ can forge a new valid message-signature pair (m, σ) in $\mathcal{DS}_u$. By similar reasoning, we can then re-compress the new signature without recalculating the hashes to obtain a valid message-signature pair for $\mathcal{DS}_c$, contradicting Theorem 4. □

5.3 Number of Rounds

To achieve λ bits of security, the protocol must be run at least $t = 2\lambda$ times, since a quantum adversary can choose arbitrary bits $J_1 \| \dots \| J_t$, compute simulated proofs using $J_1 \| \dots \| J_t$ as challenge, then perform a pre-image search on H using Grover's algorithm [15] to find a message m that will give the required hash. A faster collision attack does not seem to apply since an adversary must know the challenge bits beforehand in order for their simulated proofs to be verifiable with non-negligible probability. Thus to achieve λ bits of security against quantum attacks, our signature scheme runs the zero-knowledge proof $t = 2\lambda$ times.

We have seen that, in the underlying zero-knowledge proof, revealing responses to both challenges $b = 0, 1$ will allow anyone to compute the secret isogeny. Consequently, it is crucial that our signature scheme does not use the same commitment twice. We show that this happens with negligible probability.

Recall that $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1 \approx 2^{6\lambda}$ with $\ell_A^{e_A} \approx \ell_B^{e_B} \approx 2^{3\lambda}$. There are roughly $\ell_B^{e_B - 1} - 1 \approx 2^{3\lambda}$ distinct cyclic subgroups of $E[\ell_B^{e_B}]$ from which the commitments are chosen randomly. The zero-knowledge protocol is run 2λ times for each signature, so if we sign 2^s messages, we would select $2^{s+1}\lambda$ cyclic subgroups of $E[\ell_B^{e_B}]$ at random. An upper bound on the probability that we will select the same subgroup at least twice is given by the Birthday bound:

$$\frac{2^{s+1}\lambda(2^{s+1}\lambda - 1)}{2 \cdot 2^{3\lambda}} \leq \frac{2^{2s+2}\lambda^2}{2^{3\lambda+1}} \leq \frac{\lambda^2}{2^{\lambda-1}}$$

for $s \leq \lambda$, which is negligible in λ.

6 Implementations

For maximum performance, we implemented the uncompressed signature scheme by modifying the Supersingular Isogeny Diffie-Hellman (SIDH) library published by Costello, Longa, and Naehrig [8]. The SIDH implementation uses fixed public parameters: the prime $p = 2^{372} \cdot 3^{239} - 1$, the curve $E_0 : y^2 = x^3 + x$, and generators P_B, Q_B related by a distortion map. The prime p has bitlength 751, providing 186 bits of classical security and 124 bits of quantum security.

6.1 Performance

Performance tests of the uncompressed signature scheme were run on an Intel Xeon E5-2637 v3 3.5 GHz Haswell processor running CentOS v6.8, compiled with GCC v4.4.7. We also present timing results on the high-performance ARM Cortex-A57 processor in both C and an optimized arithmetic library on ASM [17]. The Juno platform provides a combination of Cortex-A57 and Cortex-A53 cores for ARMv8 big.LITTLE technology. However, our software is only benchmarked on a single high-performance Cortex-A57 core to get the most performance-oriented results. The software is compiled with Linaro GCC v4.9.4 on a single core 1.1GHz ARM Cortex-A57 running OpenEmbedded Linux v4.5.0.

The signing and verifying algorithms are easily parallelizable with linear speedup, since the computations required for each round of the ZKP protocol is independent. We have implemented parallelization for the PC platform. The timing results are summarized in Table 2.

Table 2. Performance results (in 10^6 clock cycles) on Intel Xeon E5-2637 v3 3.5 GHz.

Platform	Threads	Keygen	Signing	Verifying
PC	1	63	28,776	19,679
	2	-	14,474	10,042
	4	-	7,449	5,536
ARM (C)	-	1,656	767,928	493,797
ARM (ASM)	-	123	57,092	36,757

As noted before, the computing costs in the signing algorithm are incurred almost entirely in the ZKP rounds which can be precomputed offline. With precomputation, the signing algorithm simply needs to evaluate a hash function on the data and output the appropriate responses for the signature.

7 Conclusion

We present and implement a stateless quantum-resistant digital signature scheme based on supersingular elliptic curve isogenies with very small key sizes, useful for post-quantum applications with strict key size requirements. Combined with previous works, these results show that isogenies can provide the full range of public-key cryptographic primitives including key establishment, encryption, and digital signatures. Though our results are promising, further improvements are still needed to bring isogeny-based signatures truly into the realm of practicality.

Acknowledgments. We thank Steven Galbraith for helpful comments on an earlier version of this paper, and the anonymous reviewers for their constructive feedback. This work was partially supported by NSF grant no. CNS-1464118, NIST award 60NANB16D246, the CryptoWorks21 NSERC CREATE Training Program in Building a Workforce for the Cryptographic Infrastructure of the 21st Century, and InfoSec Global, Inc.

References

1. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014. pp. 474–483 (2014)
2. Azarderakhsh, R., Jao, D., Kalach, K., Koziel, B., Leonardi, C.: Key compression for isogeny-based cryptosystems. In: Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography. pp. 1–10. AsiaPKC '16, ACM, New York, NY, USA (2016)
3. Barreto, P.S.L.M., Longa, P., Naehrig, M., Ricardini, J.E., Zanon, G.: Sharper ring-lwe signatures. Cryptology ePrint Archive, Report 2016/1026 (2016)
4. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: Sphincs: Practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology — EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. pp. 368–397. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
5. Bernstein, D.J., Lange, T., Peters, C.: Attacking and defending the mceliece cryptosystem. In: Buchmann, J., Ding, J. (eds.) Post-Quantum Cryptography: Second International Workshop, PQCrypto 2008 Cincinnati, OH, USA, October 17-19, 2008 Proceedings. pp. 31–46. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
6. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology — CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II. pp. 361–379. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Costello, C., Jao, D., Longa, P., Naehrig, M., Renes, J., Urbanik, D.: Efficient compression of SIDH public keys. Cryptology ePrint Archive, Report 2016/963 (2016)
8. Costello, C., Longa, P., Naehrig, M.: Efficient Algorithms for Supersingular Isogeny Diffie-Hellman. In: Advances in Cryptology — CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. pp. 572–601. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
9. Courtois, N.T., Finiasz, M., Sendrier, N.: How to achieve a mceliece-based digital signature scheme. In: Boyd, C. (ed.) Advances in Cryptology — ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings. pp. 157–174. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
10. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005. Proceedings. pp. 164–175. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
11. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology — CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 40–56. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

12. Feo, L.D., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology 8(3) (Jan 2014)
13. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Proceedings on Advances in cryptology—CRYPTO '86. pp. 186–194. Springer-Verlag, London, UK, UK (1987)
14. Galbraith, S.D., Petit, C., Silva, J.: Signature schemes based on supersingular isogeny problems. Cryptology ePrint Archive, Report 2016/1154 (2016)
15. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. pp. 212–219. STOC '96, ACM, New York, NY, USA (1996)
16. Jao, D., Soukharev, V.: Isogeny-based quantum-resistant undeniable signatures. Post-Quantum Cryptography Lecture Notes in Computer Science pp. 160–179 (2014)
17. Koziel, B., Jalali, A., Azarderakhsh, R., Jao, D., Kermani, M.M.: NEON-SIDH: Efficient Implementation of Supersingular Isogeny Diffie-Hellman Key Exchange Protocol on ARM. In: Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings. pp. 88–103 (2016)
18. Petzoldt, A., Bulygin, S., Buchmann, J.: Selecting parameters for the rainbow signature scheme. In: Sendrier, N. (ed.) Post-Quantum Cryptography: Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings. pp. 218–240. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
19. Seshadri, S.M., Chandrasekaran, V.: Isogeny-based quantum-resistant undeniable blind signature scheme. Cryptology ePrint Archive, Report 2016/148 (2016)
20. Sun, X., Tian, H., Wang, Y.: Toward quantum-resistant strong designated verifier signature from isogenies. 2012 Fourth International Conference on Intelligent Networking and Collaborative Systems (2012)
21. Tani, S.: Claw finding algorithms using quantum walk. Theor. Comput. Sci. 410(50), 5285–5297 (2009)
22. Tate, J.: Endomorphisms of abelian varieties over finite fields. Inventiones Mathematicae 2(2), 134144 (1966)
23. Unruh, D.: Quantum proofs of knowledge. In: Advances in Cryptology — EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. pp. 135–152. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
24. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Advances in Cryptology - EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. pp. 755–784. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
25. Watrous, J.: Zero-knowledge against quantum attacks. SIAM Journal on Computing 39(1), 25–58 (2009)
26. Zhang, S.: Promised and distributed quantum search. In: Wang, L. (ed.) Computing and Combinatorics: 11th Annual International Conference, COCOON 2005 Kunming, China, August 16–19, 2005 Proceedings. pp. 430–439. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)