# ECLIPSE: Strongly Unforgeable Isogeny Signatures from the Prime-Degree Variant of PRISM

Dustin Ray, Anchorage Digital

September 2026

**Abstract**

ECLIPSE is the prime-degree signature construction the PRISM authors describe beside their salted scheme, with that salt, implemented. It exists because a two-dimensional isogeny signature carries an auxiliary isogeny the hash does not bind: the SQIsign round-3 specification says SQIsign cannot be strongly unforgeable for that reason, and PRISM's prime-degree variant has no auxiliary isogeny but was published without parameters, code or a running verifier. This paper is the first implementation, on the SQIsign round-3 primes: a verifier that evaluates the paper's criterion through one isogeny chain and a pull-back, a canonical wire format whose strict decoder recovers one coefficient from two Weil pairings, randomisable keys, two independent implementations that accept each other's signatures, and one key in both roles: a MIKE key pair at the level I prime is an ECLIPSE key pair, the public curve re-encoded and the signing key derived from the MIKE secret alone, with MIKE's code untouched. The only post-quantum precedent is CSIDH with CSI-FiSh, a group action; this is the first shared key in the SQIsign family. Signatures are 206 bytes at NIST level I, within 15 bytes of SQIsign at every matched state. In one session against the SQIsign reference's assembly build, signing is 2.6 times faster and verification costs twice as much, 1.6 times with the public key prepared once. Strong unforgeability is proved in the random oracle model under the PRISM authors' Problem 4, and the verifier's soundness is proved.

## 1 Introduction

A signature scheme is strongly unforgeable when no one holding only the public key can produce a second valid signature on a message already signed [ADR02]. A chain needs this wherever a signature is an identifier: a transaction named by the hash of its bytes changes identity when any signature byte changes, and a signature anyone can rewrite is a malleability every protocol downstream must defend against.

Isogeny signatures that send their response through a two-dimensional representation have lacked the property. The SQIsign round-3 specification says why: the representation embeds the response together with an auxiliary isogeny the hash does not bind, so "two-dimensional representations are inherently not unique" and "SQIsign cannot achieve strong unforgeability" [AAA$^+$26, Section 8.4.2]. The mechanism, swapping a small-degree factor of the auxiliary isogeny for another, is written out in [Ray26a, Section 6].

PRISM is hash-and-sign: the message is hashed to a degree and the signature is one isogeny of that degree from the public curve [BBC$^+$25, BBC$^+$26a]. Its salted variant is existentially unforgeable [BBC$^+$26a, Section 6, Theorem 2] with C [BBC$^+$26b] and SageMath [IVB26] code. Its Section 5 variant, with a prime degree $2^{2e-1} - s^2$ and a verifier that embeds the signature isogeny alone, no auxiliary isogeny and no pairing, was published without parameters, code or a running verifier [BBC$^+$26a, Sections 5 and 7].

This paper implements that variant and calls it ECLIPSE[1]. It contributes:

1. to our knowledge the first implementation, on the SQIsign round-3 primes, in Rust [Ray26c], and a SageMath model of it (Sections 4 and 7);
2. a verifier that evaluates the paper's criterion with one $(2, 2)$-isogeny chain, a fixed change of theta structure and a pull-back of three points, and its soundness lemma: an accepted signature carries an isogeny of the hashed degree with the transmitted torsion images (Section 4, Appendix A);
3. strong unforgeability proved in the random oracle model under Problem 4 of [BBC+26a] (Section 7);
4. parameters from the salted proof's digest term, an exact count of the hash target and the Miller–Rabin model re-run for the prime degree (Section 5);
5. a canonical wire format with a strict decoder, one coefficient recovered from two Weil pairings, and a message-binding tag (Section 6);
6. one benchmark session against the SQIsign reference's assembly build and that port (Section 8);
7. randomisable keys with the key kept in the hash (Section 9);
8. one key, both roles: a MIKE key pair at the level I prime is an ECLIPSE key pair, the public curve re-encoded and the signing key derived once from the MIKE secret alone, with MIKE's code unchanged but for the parameter set (Section 10). A CSIDH curve is also a CSI-FiSh key [CLM+18, BKV19]; this is, to our knowledge, the first shared key in the SQIsign family, where the key is a walk rather than a class group action and the hard problem is the endomorphism ring, at MIKE's and ECLIPSE's speeds.

Section 2 is a glossary and Section 3 says what exists; Section 7 proves strong unforgeability and says what was tested; Section 11 places the sizes against SQIsign.

## 2 Terminology

Table 1 defines the terms the paper uses.

## 3 Related work

**PRISM and salt-PRISM** PRISM [BBC+25] introduces the identification scheme and the hash-and-sign signature from large prime-degree isogenies, with the two-dimensional representation of the response and EUF-CMA proofs. Its successor [BBC+26a] adds the salt with its bound and parameter derivation (Theorem 2, Equation 18) and, in Section 5, the prime-degree construction with a pairing-free verifier, a sketch of strong unforgeability under its Problem 4, an interactive assumption stronger than the one behind its existential unforgeability, as the iX3DH paper notes [FLMZ26, p. 19], and Remark 11, which says that the proof of Theorem 2 carries over to a salted version of that construction. Fouotsa's PRISMO, PRISM over a supersingular group action, is stated by its author not to be strongly unforgeable, the Frobenius conjugate of a signature being another valid signature [Fou25, Remark 22], the one published strong-unforgeability statement for a PRISM relative. Its parameters, benchmarks and C code are for salt-PRISM on the round-2 SQIsign

[1] Encoding-Canonical Large-prime Isogeny Signature Extension: Section 5 of [BBC+26a] with the salt of its Section 6, the salted version its Remark 11 names. The authors did not name it; we use ECLIPSE for the construction and for this instantiation.

Table 1: Terms used in this paper.

| term | meaning |
|---|---|
| canonical encoding | A byte format in which each object has exactly one accepted byte string: every freedom of the representation is fixed by a rule the decoder checks. |
| degree | The size of an isogeny's kernel; the number of points that map to zero. A degree-$d$ isogeny is "a path of length $d$". |
| endomorphism ring | All isogenies from a curve to itself. Knowing it is the secret; every scheme here rests on it being hard to compute for a random supersingular curve. |
| hash-and-sign | Hash the message to a target, then sign by producing the object the target names. In PRISM the target is a degree and the object an isogeny of that degree. |
| isogeny | A map between elliptic curves that respects their group law. It is determined by its kernel; "walking" from one curve to another means composing isogenies of small degree. |
| Kani embedding | Kani's lemma [Kan97]: an isogeny of degree $d$ between elliptic curves, together with a complement of degree $N - d$, sits inside an isogeny of degree $N$ between products of curves. When $N - d = s^2$ is a square the complement is the scalar $s$, and no second isogeny is needed. |
| KAT | Known-answer test: the keys and signatures the reference [The26] publishes for fixed seeds, which an implementation must reproduce byte for byte. |
| megacycle | One million processor cycles, read with `rdtsc`; the unit the SQIsign reference's benchmark [The26] reports and the unit of every cost in this paper. |
| non-interactive key exchange | Two parties derive a shared secret from each other's public keys alone, with no messages exchanged, as in Diffie–Hellman. |
| MIKE | A non-interactive key exchange whose public key is a supersingular curve at the end of a $2^e$-walk from a fixed curve and whose secret is the walk's scalar; its key agreement computes a dimension-four isogeny. Section 10 puts a MIKE key in both roles. |
| salt | Random bytes hashed with the message and carried in the signature, so that the same message hashes to a fresh target each time. |
| strict decoder | A decoder that accepts a byte string only if it is the canonical encoding of what it decodes to, so that decoding followed by encoding returns the input. |
| strong unforgeability | No one holding only the public key can produce a second valid signature on a message that was signed, in any byte string. |
| supersingular curve | An elliptic curve over $\mathbb{F}_{p^2}$ whose endomorphism ring is a maximal order in a quaternion algebra. All curves in this paper are supersingular. |
| theta model | The coordinate system the verifier's chain runs in; a point of a two-dimensional variety has four theta coordinates. |
| torsion | The points of a curve whose order divides a given number. The $2^t$-torsion is where the signature's action is written down; $2^t$ divides $p + 1$ on these primes, so those points are defined over $\mathbb{F}_{p^2}$. |

primes. Three code bases exist, the authors' C implementation [BBC+26b], a SageMath proof of concept of SQIsign version 2.0 and PRISM [IVB26], and the SageMath randomisable keys built on it [IC26]; all three are salt-PRISM at the round-2 primes, and none has the Section 5 construction, its hash Hsquares or a verifier for it.

**Around PRISM** Basso, Borin, Corte-Real Santos, Dartois, Invernizzi, Maino, Pedersen and Seck define signatures with randomisable keys for SQIsign and PRISM; their PRISM-RK adds an adaptation algorithm that moves a signature to a randomised key [BBC+26c]. Kim, Kim and Lee build a threshold signature on PRISM with a graph-based access structure [KKL26]. Cottaar, Galbraith, Maino and Trimoska analyse a PRISM whose primality test is too weak and give minimal Miller–Rabin round counts for the original and the salted version [CGMT26]; Nakagawa and Onuki give the small-extension-field attack on PRISM-id and a margin for the degree [NO26]. Both are re-run for the prime degree in Section 5.

**SQIsign** The round-3 specification [AAA+26] supplies the substrate this implementation ports: primes, field and curve arithmetic, the theta-model $(2,2)$-chains of Dartois, Maino, Pope and Robert [DMPR24], the quaternion layer, the deterministic bases and hints, and the tag of its Section 8.4.1. Its Section 8.4.2 is the statement quoted above; [Ray26a, Section 6] is the mechanism. The Rust port of [Ray26d] reproduces the reference's round-3 known-answer vectors, and the same substrate carries the dimension-4 SQIsign of [Ray26b], whose one-session method Section 8 shares (same C build, same benchmark settings, verification from bytes), reporting `criterion` means where that paper reports medians.

**What we did not find** We found no implementation of the Section 5 construction before this one. The three PRISM code bases above [BBC+26b, IVB26, IC26] were read on 2026-09-22 at the commits the bibliography records and contain the salted scheme only; a GitHub search for the construction's hash and degree form returned nothing else. An implementation we missed would change this paragraph and nothing else.

## 4 The construction

Fix a prime $p = c \cdot 2^e - 1$, a torsion exponent $e' \leq e - 2$ (Section 5), and write $t = e' + 2$. Keys are salt-PRISM's [BBC+26a, Section 3.3, Remark 2 and Section 7.2] with $2^t$ for $2^a$: the public key is the Montgomery coefficient of $E_{\mathrm{vk}}$ and a one-byte hint for the deterministic basis $(P_{\mathrm{vk}}, Q_{\mathrm{vk}})$ of $E_{\mathrm{vk}}[2^t]$; the secret key is $(E_{\mathrm{vk}}, I_{sk}, M_{sk})$, the ideal of $E_0 \to E_{\mathrm{vk}}$ and the matrix modulo $2^t$ that writes this basis in the images of the fixed basis of $E_0[2^t]$. Key generation is SQIsign's [AAA+26].

**What is taken and what is changed** The construction is Section 5 of [BBC+26a] with the salt of its Section 6, the salted version its Remark 11 names. Taken from there: the degree form $2^{2e'-1} - s^2$ and the hash Hsquares into Squares, the response as one isogeny from the public curve, the criterion that the top-left block of the embedding is $[s]$, the factorisation $\Phi_1 = \Psi_{e'} \circ \Phi_2 \circ \iota$, and the hash input $(E_{\mathrm{vk}}, msg, \mathrm{salt})$. Ours: the primes and the exponents $e' = a + 1$, $t = e' + 2$ (Section 5), the tag, the single chain with the constant $\omega$ and the pull-back in place of a second chain, the scalar check on three points of $E_{\mathrm{vk}}[2^t]$, the deterministic salt, and the wire format with its decoder (Section 6). Algorithms 1 and 2 mark each block with its source; Figure 2 is the data flow with each phase's share of the level I cost.

**The scheme on one page** Write $N = 2^{2e'-1}$. A public key is a curve $E_{\mathrm{vk}}$ with the deterministic basis $(P_{\mathrm{vk}}, Q_{\mathrm{vk}})$ of $E_{\mathrm{vk}}[2^t]$; a signature is a curve $E_{\mathrm{sig}}$, two points $P_{\mathrm{sig}}, Q_{\mathrm{sig}}$ of $E_{\mathrm{sig}}[2^t]$, a salt and a tag.

*The hash.* Hsquares maps $(E_{\mathrm{vk}}, msg, \mathrm{salt})$ to an odd $s < 2^{e'-1}$ with $d = N - s^2$ prime: the digest is $2n$ bits, $n = e' - 1$; $s = 2h + 1$ for $h$ the low $e' - 2$ bits; the salt is rejected unless $d$ passes trial division and $t$ Miller–Rabin rounds; the tag is $\lambda$ bits of the high half. The accepted $s$ form $\mathrm{Squares}_{e'}$ of [BBC+26a, Section 5], and the signer's work is one isogeny of degree $d$ from $E_{\mathrm{vk}}$.

*Kani's criterion.* Kani's lemma [BBC+26a, Theorem 1] with the coprime summands $s^2 + d = N$: an isogeny $\sigma\colon E_{\mathrm{vk}} \to E_{\mathrm{sig}}$ of degree $d$ is the bottom-left block of the $(N, N)$-endomorphism

$$F = \begin{pmatrix} [s] & -\hat{\sigma} \\ \sigma & [s] \end{pmatrix} : E_{\mathrm{vk}} \times E_{\mathrm{sig}} \to E_{\mathrm{vk}} \times E_{\mathrm{sig}}, \qquad \hat{F}F = [N],$$

and $F$ factors through the $2^{e'}$-torsion: $F = \hat{\Phi}_2 \circ \Phi_1$ with $\Phi_1$ the $(2^{e'}, 2^{e'})$-isogeny of kernel $\{([s]P, -\sigma(P)) : P \in E_{\mathrm{vk}}[2^{e'}]\}$ and $\Phi_2$ the $(2^{e'-1}, 2^{e'-1})$-isogeny of kernel $\{([s]P, \sigma(P)) : P \in E_{\mathrm{vk}}[2^{e'-1}]\}$, so that $\Phi_1 = \Psi_{e'} \circ \Phi_2 \circ \iota$ with $\Psi_{e'}$ one $(2,2)$-step and $\iota(P, Q) = (P, -Q)$ [BBC+26a, Section 5]. With $u = s^{-1} \bmod 2^t$ the kernel of $\Phi_1$ is generated by $(P_{\mathrm{vk}}, -P_{\mathrm{sig}})$ and $(Q_{\mathrm{vk}}, -Q_{\mathrm{sig}})$ for $P_{\mathrm{sig}} = [u]\sigma(P_{\mathrm{vk}})$, $Q_{\mathrm{sig}} = [u]\sigma(Q_{\mathrm{vk}})$, which is what the signature transmits. The criterion [BBC+26a, Section 5]: an $(N, N)$-endomorphism of $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$ embeds an isogeny of degree $d$ exactly when its top-left block is $[s]$, and that block is $[s]$ once it is $[s]$ on $E_{\mathrm{vk}}[N']$ for any $N'$ with $N'^2 > 2^{2e'} + 1$; we take $N' = 2^t$.

*The verifier.* Algorithm 2 evaluates $F$ on three points of $E_{\mathrm{vk}}[2^t]$ without a second chain (Figure 1): it runs the chain of $\Phi_1$, $e'$ steps from $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$, carrying $([u]T, 0)$ for $T \in \{P_{\mathrm{vk}}, Q_{\mathrm{vk}}, P_{\mathrm{vk}} - Q_{\mathrm{vk}}\}$ and recording the null point of every surface; moves the codomain $A_1$ of the chain to the codomain $A_2$ of its first $e' - 1$ steps by the constant change of theta structure $\omega$; and pulls the carried points back through the recorded steps, the dual $\hat{\Phi}_2$. Its three checks are, in order:

1. every step of the chain has a valid $(2,2)$-kernel, so that the chain is a chain of polarised isogenies with the transmitted kernel;
2. $\omega$ carries the null point of $A_1$ to the null point of $A_2$, so that the two codomains are one surface and $\hat{\Phi}_2 \circ \omega \circ \Phi_1$ is an endomorphism of $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$;
3. that endomorphism sends $([u]T, 0)$ to $\pm(T, T_{\mathrm{sig}})$ for the three $T$, $T_{\mathrm{sig}}$ the matching signature point, so that its top-left block is $[s]$ on $E_{\mathrm{vk}}[2^t]$ and its bottom-left block has degree $d$ and the transmitted images.

Before any of this the decoder has fixed the representation (Section 6) and the tag has been compared.

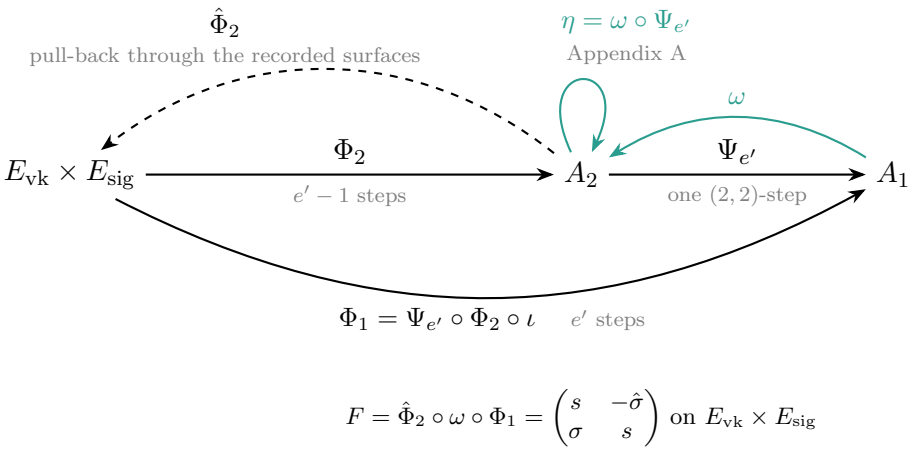



Figure 1: The two isogenies of the criterion, the map between their codomains, and the verifier's way back. $\Phi_2$ is the first $e' - 1$ steps of the chain of $\Phi_1$ and $\Psi_{e'}$ its last step. $\omega\colon A_1 \to A_2$ is the constant change of theta structure that identifies the two codomains; the loop $\eta = \omega \circ \Psi_{e'}$ is the endomorphism of $A_2$ from which Appendix A derives it. The dashed arrow is the pull-back of the three carried points through the surfaces the chain visited.

**Algorithm 1** Sign$(sk, msg)$

**Input:** $sk = (E_{\mathrm{vk}}, I_{sk}, M_{sk})$; a message $msg$.
**Output:** A signature of $2\,\mathrm{FP} + 1 + 3\lceil (e'+2)/8 \rceil + k/8 + \lambda/8$ bytes.

stream $\leftarrow$ SHAKE256("ECLIPSE-det-v1" $\|$ $sk$ $\|$ $msg$) ▷ ours; every random choice below is drawn from it
**repeat**
  salt $\leftarrow$ $k$ bits of the stream ▷ the salt of [BBC+26a, Figure 8]
  $H \leftarrow$ SHAKE256("ECLIPSE-hsq-v1" $\|$ $E_{\mathrm{vk}}$ $\|$ $msg$ $\|$ salt), squeezed to $2n$ bits, $n = e' - 1$
  $s \leftarrow 2h + 1$ for $h$ the low $e' - 2$ bits of $H$; tag $\leftarrow$ the first $\lambda$ bits of the high half ▷ Hsquares of [BBC+26a, Section 5]; the tag is ours
  $d \leftarrow 2^{2e'-1} - s^2$
**until** $d$ passes trial division and $t$ Miller–Rabin rounds with random bases
$I_{\mathrm{chall}} \leftarrow$ a left $O_0$-ideal of norm $d$ ▷ from here, GenIsogenyStrong [BBC+26a, Algorithm 2] at $2^t$; the sampler, one square root modulo $d$ with no rejection, is ours
$(\varphi, \varphi(P_0), \varphi(Q_0)) \leftarrow$ IdealToIsogeny$(I_{sk} \cap I_{\mathrm{chall}})$; $E_{\mathrm{sig}} \leftarrow$ the codomain of $\varphi$ ▷ one norm equation, one $(2^f, 2^f)$-chain
$u \leftarrow s^{-1} \bmod 2^t$; $(P_{\mathrm{sig}}, Q_{\mathrm{sig}}) \leftarrow [u]M_{sk}$ applied to $(\varphi(P_0), \varphi(Q_0))$, reduced to order $2^t$
**return** Encode$(E_{\mathrm{vk}}, E_{\mathrm{sig}}, P_{\mathrm{sig}}, Q_{\mathrm{sig}}, \mathrm{salt}, \mathrm{tag})$ ▷ canonical, Section 6

**Algorithm 2** Verify$(vk, msg, \sigma)$

**Input:** $vk = (A(E_{\mathrm{vk}}), \mathrm{hint})$; a message $msg$; a byte string $\sigma$.
**Output:** accept or reject.

1: $(E_{\mathrm{sig}}, P_{\mathrm{sig}}, Q_{\mathrm{sig}}, \mathrm{salt}, \mathrm{tag}) \leftarrow \mathrm{Decode}(vk, \sigma)$; **reject** on $\perp$ ▷ strict; the fourth coefficient from two Weil pairings
2: $(s, \mathrm{tag}') \leftarrow$ the hash of lines 4 and 5 of Algorithm 1 on $(E_{\mathrm{vk}}, msg, \mathrm{salt})$
3: **reject** unless $\mathrm{tag}' = \mathrm{tag}$ (compared in constant time) and $d = 2^{2e'-1} - s^2$ passes the $t$ rounds
4: $(P_{\mathrm{vk}}, Q_{\mathrm{vk}}) \leftarrow$ the basis of $E_{\mathrm{vk}}[2^t]$ from the hint; $u \leftarrow s^{-1} \bmod 2^t$
5: Run the $(2,2)$-chain of $e'$ steps from $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$ with kernel $\langle (P_{\mathrm{vk}}, -P_{\mathrm{sig}}), (Q_{\mathrm{vk}}, -Q_{\mathrm{sig}}) \rangle$, checking the kernel at every step, carrying $([u]T, 0)$ for $T \in \{P_{\mathrm{vk}}, Q_{\mathrm{vk}}, P_{\mathrm{vk}} - Q_{\mathrm{vk}}\}$ and recording the null point $\theta_j$ of every intermediate surface; **reject** if a step fails ▷ $\Phi_1$ of [BBC+26a, Section 5]
6: **reject** unless $\omega(\theta_{e'}) = \theta_{e'-1}$ ▷ ours, as are lines 7 to 9: the constant change of theta structure
7: **for** each carried point, with $q$ its image after $e'$ steps moved to $A_2$ by $\omega$ **do**
8: pull $q$ back through the $e' - 1$ recorded surfaces, $q \leftarrow H(q^2)/\theta_j$ at each; **reject** on a zero coordinate
9: **reject** unless the result is $\pm(T, T_{\mathrm{sig}})$ on the two Kummer lines, $T_{\mathrm{sig}}$ the matching signature point
10: **end for**
11: **return** accept

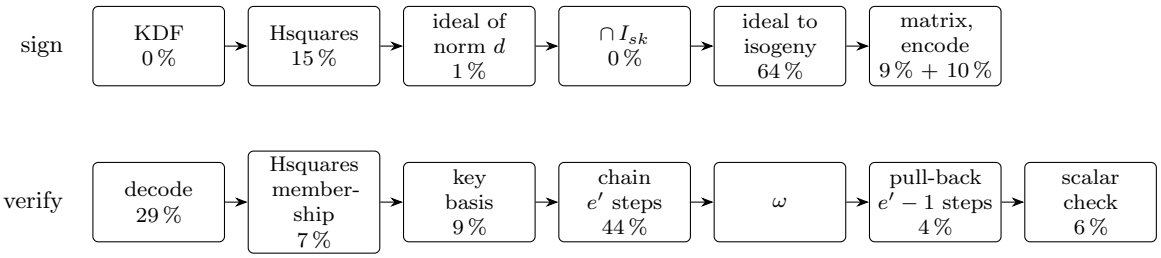



Figure 2: Signing (top) and verification (bottom) of ECLIPSE, one box per phase, with each box's share of the level I median of the phase profile of 2026-09-25 (signing 44.4 megacycles including encoding, verification 22.5 from bytes). The $\omega$ box and the dual of the gluing that ends the pull-back are below one percent each; the pairing recovery inside the decoder is 12 % of verification.

**Why it is a proof of the degree** Section 5 of [BBC+26a] tests the criterion of Section 4 as $\Psi_{e'} \circ \Phi_2([2^{e'-1}]T, 0) = \Phi_2([s]T, 0)$ on a basis of $E_{\mathrm{vk}}[N']$ with $N' = p + 1$, computing both sides as chains. Algorithm 2 evaluates the same condition with three changes: $\Phi_1$ is the chain of line 5 and $\Phi_2$ its first $e' - 1$ steps, computed with two theta structures that the constant $\omega$ identifies; $\hat{\Phi}_2$ is evaluated by pulling the carried points back through the recorded steps, not by a second chain; and $N' = 2^t$ meets the condition, so three points of $E_{\mathrm{vk}}[2^t]$, whose basis the hint already gives, replace a basis of $E_{\mathrm{vk}}[p + 1]$. Lemma 1 proves that an accepting run yields the degree-$d$ isogeny with the transmitted images.

**The pull-back** The dual of a $(2,2)$-step $f\colon A \to B$ at a point $Q$ of $B$ with dual theta coordinates $q$ is $\hat{f}(Q) = H(q^2)/\theta_A(0)$, $H$ the Hadamard transform and $\theta_A(0)$ the null point the forward chain recorded: the duplication formula on $A$ read through $\hat{f} \circ f = [2]$ [DMPR24]. So $\hat{\Phi}_2$ is one division

per recorded step and carried point, with no kernel and no codomain arithmetic; the last step is the dual of the gluing and lands in $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$. It costs a tenth of the forward chain.

**The constant $\omega$** In [BBC+26a, Section 5] the codomains of $\Phi_1$ and $\Phi_2$ are one surface. Computed as theta-model chains they are that surface with two theta structures, $A_1$ and $A_2$, and $\omega\colon A_1 \to A_2$ is the change between them: a constant, the Hadamard transform followed by the swap of the two middle coordinates (Figure 1; Appendix A derives it). The verifier applies it to the null point of $A_1$ and rejects unless the result is the null point of $A_2$.

**Soundness of the verifier** Theta coordinates are of level 2 throughout, so a point is known up to sign and a point of a product up to a sign on each factor; $H$ is the Hadamard transform and $S$ the coordinate-wise square [DMPR24, Section 2.1].

**Lemma 1 (Soundness of Algorithm 2)** *Let $vk = (E_{\mathrm{vk}}, \mathrm{hint})$ decode, with hint basis $(P_{\mathrm{vk}}, Q_{\mathrm{vk}})$ of $E_{\mathrm{vk}}[2^t]$, let $(E_{\mathrm{sig}}, P_{\mathrm{sig}}, Q_{\mathrm{sig}}, \mathrm{salt}, \mathrm{tag})$ be the decoding of a byte string under $vk$, and let $s$ be odd with $0 < s < 2^{e'-1}$, $d = 2^{2e'-1} - s^2$ and $u = s^{-1} \bmod 2^t$. If lines 5 to 9 of Algorithm 2 accept, there is an isogeny $\sigma\colon E_{\mathrm{vk}} \to E_{\mathrm{sig}}$ defined over $\mathbb{F}_{p^2}$, of degree $d$, and a sign $\varepsilon \in \{\pm 1\}$ with*

$$\sigma(P_{\mathrm{vk}}) = \varepsilon[s]P_{\mathrm{sig}}, \qquad \sigma(Q_{\mathrm{vk}}) = \varepsilon[s]Q_{\mathrm{sig}},$$

*that is, $(P_{\mathrm{sig}}, Q_{\mathrm{sig}}) = \pm[u]\sigma(P_{\mathrm{vk}}, Q_{\mathrm{vk}})$, the transmitted torsion images up to the common sign. The lemma does not use that $d$ is prime.*

The decoder supplies that $(P_{\mathrm{sig}}, Q_{\mathrm{sig}})$ is a basis of $E_{\mathrm{sig}}[2^t]$ over $\mathbb{F}_{p^2}$ (odd determinant, three points of order $2^t$) with $e_{2^t}(P_{\mathrm{sig}}, Q_{\mathrm{sig}}) = e_{2^t}(P_{\mathrm{vk}}, Q_{\mathrm{vk}})^{-1}$, the relation the omitted coefficient is recovered from. So the subgroup $K$ generated by $(P_{\mathrm{vk}}, -P_{\mathrm{sig}})$ and $(Q_{\mathrm{vk}}, -Q_{\mathrm{sig}})$ is isomorphic to $(\mathbb{Z}/2^t)^2$ and isotropic for the $2^t$-Weil pairing of the product polarisation, its $2^{e'}$-part $K_1 = [4]K$ is maximal isotropic in $(E_{\mathrm{vk}} \times E_{\mathrm{sig}})[2^{e'}]$, and $K$ is the two bits of torsion above $K_1$ the theta chain needs [DMPR24, Section 4.1].

*Proof. Step 1, the chain.* Line 5 runs the theta-model $(2,2)$-chain of [DMPR24, Section 4] from $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$ with the generators of $K$, checking each step's kernel. By the correctness of that chain (the gluing step from the product theta structure of [DMPR24, Proposition 3], then generic steps, each codomain null point from the 8-torsion above the step's kernel), a run that completes computes $(2,2)$-isogenies $E_{\mathrm{vk}} \times E_{\mathrm{sig}} \to A^{(1)} \to \cdots \to A^{(e')}$ between principally polarised abelian surfaces, each with the level-2 theta structure the chain gives it and the null point $\theta_j$ the verifier records; $\Phi_1 = \Psi_{e'} \circ \cdots \circ \Psi_1$ is a $(2^{e'}, 2^{e'})$-isogeny of kernel $K_1$ and $\Phi_2 = \Psi_{e'-1} \circ \cdots \circ \Psi_1$ a $(2^{e'-1}, 2^{e'-1})$-isogeny of kernel $[2]K_1$. Write $A_1 = A^{(e')}$, $A_2 = A^{(e'-1)}$.

*Step 2, $\omega$ and the null-point check.* A level-2 theta structure on a principally polarised abelian surface is an isomorphism of the Heisenberg group $G(2)$ with the theta group of $2\Theta$; a change of theta structure is composition with an automorphism of $G(2)$ fixing $k^*$, which induces a symplectic automorphism of $H(2) = K(2) \times \hat{K}(2)$ and acts on theta coordinates by a fixed matrix [DMPR24, Section 2.1], [Dar24, Theorem 12]. The constant $\omega$ of Appendix A is the Hadamard transform, the automorphism exchanging $K(2)$ and $\hat{K}(2)$, followed by the exchange of the two index bits of $K(2) = (\mathbb{Z}/2)^2$ with its dual; both are symplectic, so $\omega$ is the action of a symplectic automorphism $\psi$, and $\omega(\theta_{e'})$ is the null point of the structure $\Theta_1 \circ \psi$ on $A_1$. Line 6 accepts only if $\omega(\theta_{e'}) = \theta_{e'-1}$: $(A_1, \Theta_1 \circ \psi)$ and $(A_2, \Theta_2)$ have the same level-2 null point. We use the following standard fact as stated: *a level-2 theta null point determines the principally polarised abelian surface with its theta structure up to isomorphism over $\bar{\mathbb{F}}_p$.* The level-2 theta functions map the surface onto its Kummer

surface, the quotient by $\pm 1$, canonically once the theta structure is fixed [Mum66, pp. 297 and 304]; the null point fixes the image quartic with its sixteen nodes, the images of the 2-torsion, and the level-2 structure on it, and the surface is the double cover of the Kummer surface branched over the nodes with the pulled-back polarisation. The classical statement is over $\mathbb{C}$ [BL04, Chapter 10, Sections 10.2 and 10.3], [GD94]; Mumford's theory gives the embedding and the structure in characteristic not 2 [Mum66, Section 2]. Granting it, there is an isomorphism of principally polarised abelian surfaces $\alpha\colon A_1 \to A_2$ carrying $\Theta_1 \circ \psi$ to $\Theta_2$ and acting on theta coordinates as $\omega$ does, so line 7 evaluates $\alpha$ on the carried points.

*Step 3, the pull-back.* For one $(2,2)$-step $f\colon A \to B$ of the chain with recorded null point $\theta_A(0)$, equation (2) of [DMPR24, Section 2.2], the duplication formula read through the isogeny theorem, at $P = Q = Y$ with $f(Y) = X$ and $\hat{f} \circ f = [2]$ gives

$$\theta^A(\hat{f}(X)) = H\big(S(\tilde{\theta}^B(X))\big)/\theta_A(0)$$

coordinate-wise, $\tilde{\theta}^B$ the dual coordinates on $B$ and $\hat{f}$ the dual with respect to the principal polarisations: this is line 8 through the $e' - 1$ recorded generic steps in reverse order, and it computes $\hat{\Phi}_2$ on the carried points. The division needs every coordinate of $\theta_A(0)$ non-zero, which the generic steps guarantee [DMPR24, Remark 6]; line 8 rejects when any recorded null point has a zero coordinate, since a zero would mean an intermediate surface whose level-2 coordinates do not determine the point [DMPR24, Appendix A], and line 9 rejects when the gluing's adapted null point $\theta_M(0)$ has a zero coordinate, since the comparison below multiplies by $\theta_M(0)$ and needs $y \mapsto y \star \theta_M(0)$ injective. (Equation (2) reads $(\theta_i^A(P+Q))_i \star (\theta_i^A(P-Q))_i = H((\tilde{\theta}_i^B(f(P)))_i \star (\tilde{\theta}_i^B(f(Q)))_i)$; with $P = Q = Y$ the left side is $\theta^A(2Y) \star \theta^A(0) = \theta^A(\hat{f}(X)) \star \theta^A(0)$ and the right side $H(S(\tilde{\theta}^B(X)))$. In the implementation, `DualCache::pull_back` applies, per recorded step, the square, the product with the inverted null point and the Hadamard transform.) The last step back is the dual of the gluing $f_1\colon E_{\mathrm{vk}} \times E_{\mathrm{sig}} \to A^{(1)}$, whose domain carries the non-product theta structure the gluing adapted to its kernel through the change-of-basis matrix $M$ of [DMPR24, Section 4.1, "Gluing Isogeny"]: in those coordinates the product theta coordinates of a couple point $(T_1, T_2)$ are $M\,(x_1x_2 : x_1z_2 : z_1x_2 : z_1z_2)$, the Segre product of the two Kummer-line coordinates, and the same identity reads $\theta_M(\hat{f}_1(X)) \star \theta_M(0) = H(S(\tilde{\theta}^{A^{(1)}}(X)))$. The implementation (`gluing_dual_equals`) does not solve for $\hat{f}_1(X)$: line 9 checks the identity projectively against the expected couple point, rejecting when either side is the zero vector, and since the Segre embedding is injective on the product of Kummer lines the identity holds exactly when $\hat{f}_1(X) = (\pm T_1, \pm T_2)$. No splitting and no canonical form is needed, the codomain of the dual gluing being the domain itself [DMPR24, Section 4.1].

*Step 4, the composite.* Put $F = \hat{\Phi}_2 \circ \alpha \circ \Phi_1$ on $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$. With $\hat{\alpha} = \alpha^{-1}$,

$$\hat{F}F = \hat{\Phi}_1\,\alpha^{-1}\,\Phi_2\hat{\Phi}_2\,\alpha\,\Phi_1 = \hat{\Phi}_1\,[2^{e'-1}]\,\Phi_1 = [N], \qquad N = 2^{2e'-1},$$

so $F$ is an $(N,N)$-isogeny for the product polarisation. Write $F = \begin{pmatrix} a & b \\ c & \delta \end{pmatrix}$ with $a \in \mathrm{End}(E_{\mathrm{vk}})$ and $c\colon E_{\mathrm{vk}} \to E_{\mathrm{sig}}$; the top-left entry of $\hat{F}F = [N]$ is $\hat{a}a + \hat{c}c = [\deg a + \deg c] = [N]$. Line 9 accepts only if $F([u]T, 0) = (a([u]T), c([u]T)) = (\pm T, \pm T_{\mathrm{sig}})$ on the two Kummer lines for $T \in \{P_{\mathrm{vk}}, Q_{\mathrm{vk}}, P_{\mathrm{vk}} - Q_{\mathrm{vk}}\}$, $T_{\mathrm{sig}}$ the matching point of $\{P_{\mathrm{sig}}, Q_{\mathrm{sig}}, P_{\mathrm{sig}} - Q_{\mathrm{sig}}\}$, the signs independent. If $a([u]P_{\mathrm{vk}}) = \varepsilon_1 P_{\mathrm{vk}}$ and $a([u]Q_{\mathrm{vk}}) = \varepsilon_2 Q_{\mathrm{vk}}$ with $\varepsilon_1 \neq \varepsilon_2$, then $a([u](P_{\mathrm{vk}} - Q_{\mathrm{vk}})) = \pm(P_{\mathrm{vk}} + Q_{\mathrm{vk}})$ could equal $\pm(P_{\mathrm{vk}} - Q_{\mathrm{vk}})$ only if $2P_{\mathrm{vk}} = 0$ or $2Q_{\mathrm{vk}} = 0$, impossible at order $2^t$; so $a([u]T) = \varepsilon T$ with one sign, and since $su \equiv 1 \pmod{2^t}$, $a = \varepsilon[s]$ on $E_{\mathrm{vk}}[2^t]$. Then $a - \varepsilon[s] = [2^t]\beta$ for some $\beta \in \mathrm{End}(E_{\mathrm{vk}})$, and as $\deg a \leq N$ and the degree is a positive definite quadratic form, $\deg(a - \varepsilon[s]) \leq (\sqrt{N} + s)^2 < 4N = 2^{2e'+1}$, so

$\deg\beta < 2^{2e'+1}/2^{2t} = 1/8$: $\beta = 0$ and $a = \varepsilon[s]$. This is the criterion of [BBC+26a, Section 5] at $N' = 2^t$, and it gives $\deg c = N - s^2 = d$. The same sign argument on the basis $(P_{\mathrm{sig}}, Q_{\mathrm{sig}})$ of $E_{\mathrm{sig}}[2^t]$ gives $c([u]T) = \varepsilon' T_{\mathrm{sig}}$ with one sign $\varepsilon'$; set $\sigma = c$, so $\sigma(P_{\mathrm{vk}}) = \varepsilon'[s]P_{\mathrm{sig}}$ and $\sigma(Q_{\mathrm{vk}}) = \varepsilon'[s]Q_{\mathrm{sig}}$.

*Rationality.* Steps 2 to 4 produce $\sigma$ over $\bar{\mathbb{F}}_p$. A supersingular curve over $\mathbb{F}_{p^2}$ has Frobenius trace in $\{0, \pm p, \pm 2p\}$, and a rational basis of its $2^t$-torsion with $t \geq 2$ forces the group of rational points to be $(\mathbb{Z}/(p+1))^2$ or $(\mathbb{Z}/(p-1))^2$, that is Frobenius $[-p]$ or $[p]$; since $p \equiv 3 \pmod 4$ gives $v_2(p-1) = 1$, the second has no rational 4-torsion. So the hint bases of order $2^t$ over $\mathbb{F}_{p^2}$ put Frobenius equal to $[-p]$ on both $E_{\mathrm{vk}}$ and $E_{\mathrm{sig}}$, which excludes the quadratic twist, and for any isogeny $\sigma\colon E_{\mathrm{vk}} \to E_{\mathrm{sig}}$ over $\bar{\mathbb{F}}_p$ the conjugate $\sigma^{(p^2)}$ satisfies $\sigma^{(p^2)} \circ [-p] = [-p] \circ \sigma$, hence $\sigma^{(p^2)} = \sigma$: $\sigma$ is defined over $\mathbb{F}_{p^2}$. □

**Remarks on the lemma** Only Step 2 rests on a statement that is not a numbered result in [DMPR24], [AAA+25] or [Dar24]; the rest is [DMPR24, Sections 2 to 4], [Dar24, Theorem 12] and elementary. Completeness, that honest signatures are accepted, is the construction of Section 4 with the derivation of $\omega$ in Appendix A, and is tested in the artifact (Section 7); the lemma does not need it. Primality of $d$ is the hash's business: with $d$ composite the lemma still yields an isogeny of degree $d$, which is the reroute of Section 7 and why the round count of Section 5 matters.

**The scalar check on $E_{\mathrm{vk}}[2^t]$** $N = 2^t$ meets the condition of [BBC+26a, Section 5], and we take it over their $p + 1$ because its torsion is rational and three of its points are what the chain already carries: on Kummer lines a basis is the three $x$-coordinates of $P$, $Q$ and $P - Q$ of the hint basis. An accepted triple belongs to exactly one isogeny of degree $d$ (Lemma 4; [BBC+26a, Section 5] states the same for the $2^{e'}$-torsion). Multiplying the three carried points by $u = s^{-1} \bmod 2^t$ instead of the six images by $s$ halves the ladders; $F$ is $\mathbb{Z}$-linear and the points have order $2^t$, so the conditions agree.

**Determinism** By default the salt and every later random choice come from SHAKE256 of a domain string, the secret key and the message, the RFC 6979 pattern [Por13], so a signature is a function of key and message, which [BBC+26a, Remark 8 and Section 5.1] impose for their sketch; Theorem 3 does not need it. A randomised mode draws the salt from fresh entropy; the verifier cannot tell the modes apart, and in either mode a second signature on the same message is a fresh one with another salt, degree and isogeny.

## 5 Parameters

**The primes** The three primes are the SQIsign round-3 primes $p = 3 \cdot 2^{324} - 1$, $27 \cdot 2^{500} - 1$ and $17 \cdot 2^{664} - 1$ [AAA+26]. Key recovery in PRISM is the endomorphism ring problem of $E_{\mathrm{vk}}$, which is why PRISM takes SQIsign's primes [BBC+26a, Sections 4.4 and 7.1]; the round-3 submission sized them after Wesolowski's $p^{1/3+o(1)}$ algorithm [Wes26] under a memory-bounded cost model [AAA+26, Section 8.2.1.2], and nothing here changes that. Public keys are 83 / 129 / 169 bytes.

**The two shipped level I sets** The implementation ships two level I parameter sets, as compile-time types (Table 2). The default is the round-3 prime $3 \cdot 2^{324} - 1$ with every parameter above; MIKE (Section 10) runs on it unmodified with its fast level I walk length $e = 228$. That prime lies inside MIKE's own definition, "a prime of the shape $p = 2^f c - 1$ with $c$ small odd; a positive integer $e \leq f - 2$" [BDD+26, Section 4.4]; MIKE's parameter appendix lists the SQIsign primes with $e = 228$ or 322 as one of its three concrete choices [BDD+26, Appendix J.4], and asked on the

pqc-forum list on 2026-10-06, Damien Robert answered that "for your use case you could simply reuse the exact same primes as for SQIsign" [Rob26]. The second set, for holders of keys at MIKE's rigorous level I set, is ECLIPSE with the same scheme parameters on that set's prime $117 \cdot 2^{374} - 1$ (381 bits, seven limbs); its cost is in Section 8. MIKE's fast prime was built and measured and is not shipped (Appendix B). A set is a type, never a runtime choice: each needs reviewed field arithmetic and constants, and a key on one prime means nothing on another.

Table 2: The two shipped level I parameter sets. Every scheme parameter of Table 3 is the same on both.

| set | prime | bits | for | public key | signature |
|---|---|---|---|---|---|
| shared (default) | $3 \cdot 2^{324} - 1$ | 326 | ECLIPSE as published; MIKE on it with $e = 228$ | 83 B | 206 B |
| MIKE rigorous | $117 \cdot 2^{374} - 1$ | 381 | keys at MIKE's rigorous level I set | 97 B | 220 B |

**The digest** The digest length is salt-PRISM's: Theorem 2 gives $n \geq \lambda + \log_2 N_{\text{sign}}$, and with $N_{\text{sign}} = 2^{64}$ and $a = n$ this is $a = \lambda + 64 = 192/256/320$ [BBC+26a, Equation 18 and the text after it].

**The torsion exponent** $e' = a + 1$ Section 5 of [BBC+26a] uses the full exponent $e$ of $p$, because its own primes have $2\lambda \approx e$. We take $e' = a + 1 = 193/257/321$ instead, so the chain has $e'$ steps and each coefficient $\lceil (e' + 2)/8 \rceil$ bytes at any prime with enough torsion: the signature does not grow with the prime. The construction works for any $4 \leq e' \leq e - 2$ (the degree is prime with $2e' - 1$ bits, the kernel is rational, $s$ is odd, and the chains [DMPR24] need two bits of torsion above the kernel, which $e' + 2 \leq e$ gives). The lower bound is the digest: Hsquares draws from $[0, 2^{e'-1})$ and needs $e' - 1 \geq \lambda + 64 = a$, so $e' = a + 1$ puts the digest term where salt-PRISM's is. Nothing asks for more: an isogeny of degree $d$ from a curve of unknown endomorphism ring costs $\tilde{O}(\min(d^2, p^{1/2}))$, so the isogeny problems need only $a = \tilde{O}(\lambda)$ [BBC+26a, Section 4.4, Remark 10], and the salted bound constrains the digest length, not the exponent [BBC+26a, Equation 18]. Taking $e' = e$ would only make the chains 1.7 times longer. This reading is ours. Figure 3 places the sizes against SQIsign's at both rounds' primes.

**The density of** $\text{Squares}_{e'}$ How often is $2^{2e'-1} - s^2$ prime? Section 5.1 of [BBC+26a] estimates the density among odd $s$ as $\sqrt{2} \ln((e' - 1) \ln 2)/((e' - 1) \ln 2)$, from Conjecture H of Hardy and Littlewood [HL23]. The conjecture's exact prediction is $2C / \ln 2^{2e'-1}$ per odd $s$, with $C = \prod_{p \text{ odd}} (1 - (2/p)/(p - 1)) = 1.8500544\ldots$ computed by Cohen's method [Coh98]; $C$ is the same at every level because the Legendre symbol of $2^{2e'-1}$ is that of 2. We measured it twice (Figure 4). Sage, every odd $s$ at $e' \leq 18$ and 4000 random odd $s$ at each larger exponent, 51, 33 and 30 of them giving a prime at the three levels: 0.0128/0.0083/0.0075 at $e' = 193/257/321$. The Rust signer, 3000 signatures per level, the density as the reciprocal of the mean number of candidates per signature: 0.0139/0.0106/0.0084. The prediction is 0.0139/0.0104/0.0083; the paper's estimate is 0.0520/0.0413/0.0344, which appears to be about three times too high. Three factors would explain the gap: the estimate takes the GRH bound $\ln \ln n$ for the singular series where the truth is the constant $C$; it uses $\ln 2^{e'-1}$ where the count has $\ln 2^{2e'-1}$; and it spreads the count over all $s$ where every representation has odd $s$. The three multiply to 3.74 against the measured 3.75. The density

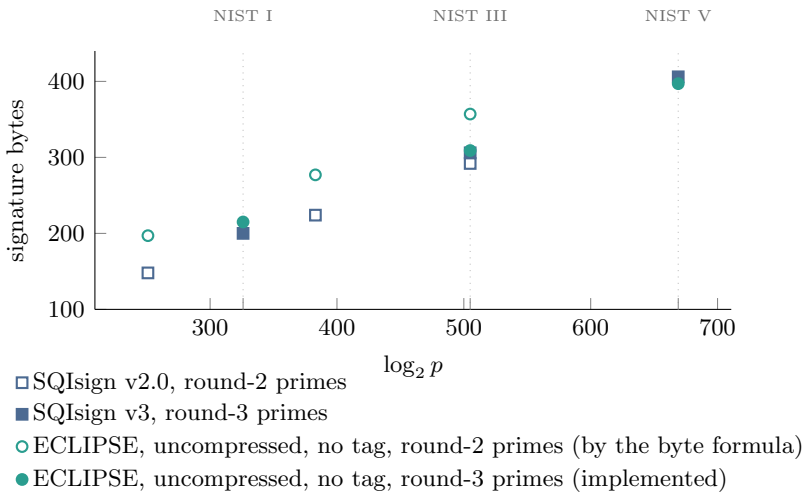



Figure 3: Signature size against the prime, SQIsign and ECLIPSE at the matched state (four coefficients, no tag). SQIsign sizes from the two specifications' Table 1 [AAA+25, AAA+26]; the ECLIPSE round-2 points are by the same byte formula, since the scheme is not instantiated on the round-2 primes. Level markers at the round-3 primes.

sets the signing cost, about 72/95/119 candidates per signature rather than the estimate's 19/24/29, and the salt bound below.

**Miller–Rabin rounds** A composite $d$ accepted by the verifier gives the signature isogeny a factored degree, and a small factor can be rerouted by anyone, the malleability Section 5 of [BBC+26a] opens with; the salt lets a forger re-randomise $d$ for free, so the round count $t$ is the only protection. The paper leaves the count to Cottaar, Galbraith, Maino and Trimoska [BBC+26a, Section 3.3], who combine the smoothness the attack needs with the Damgård–Landrock–Pomerance bounds [DLP93] on a uniform odd $k$-bit integer surviving $t$ rounds and obtain $t = 22/39/55$ for the $a$-bit $q$ [CGMT26, Sections 5.1 and 5.2, Table 4]. We re-ran their model with two inputs changed: the tested number is $d$, of $k = 2e' - 1$ bits, and one number rather than two must be smooth and be factored. With their inputs it returns their Table 4; with ours, $d$ gives $t = 9/19/28$ at the bare thresholds $2^\lambda$. Because $d$ is not a uniform odd integer (it is 7 mod 8 and of a quadratic form, with no pseudoprime result known for that set), we require the modelled attack to cost more than $2^{\lambda+32}$, which gives $t = 18/28/38$ with random bases, at 0.031/0.059/0.100 ms per round on the machine and build of Section 8.

**The bases** The verifier draws its Miller–Rabin bases from the random number generator the caller passes to Verify, uniformly in $[2, d-2]$ for each round; the signer draws its own from its signing stream. Bases the forger does not know are the assumption behind the bound of [CGMT26] and so behind $t$: its pseudoprime probability is the Damgård–Landrock–Pomerance estimate for a random integer under random bases, and [CGMT26, Remark 1 and Section 6] says so. A verifier whose generator is seeded with a constant is outside that model, since strong pseudoprimes to a known base set can be constructed; the artifact's examples seed it from operating-system entropy. For consensus use, where every node must reach the same verdict, a deterministic profile would derive the bases from the public key and the digest; a forger who grinds the salt then knows the bases before choosing $d$, and the bound that survives is the worst case $4^{-t}$ per verification, which asks for about $\lambda$ rounds.

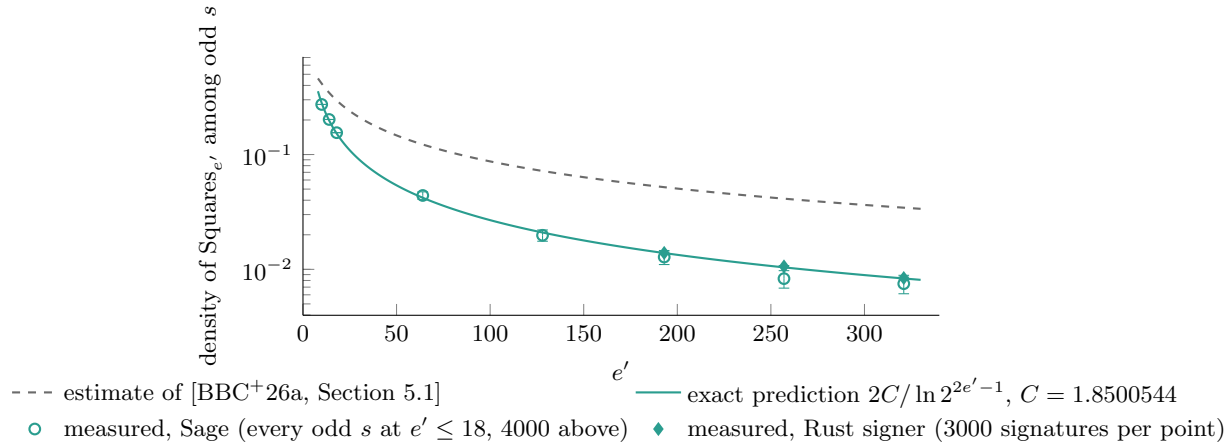



Figure 4: Density of Squares$_{e'}$ against $e'$: the Sage and Rust measurements with their sampling error, the estimate of [BBC+26a, Section 5.1], and the exact Hardy–Littlewood prediction with the constant $C$ of that section's own form. The Sage points at $e' = 10, 14, 18$ are exhaustive (70 of 256, 829 of 4096 and 10 149 of 65 536 odd $s$) and carry no sampling error; the five sampled points carry $1/\sqrt{\text{hits}}$, from 7.5 % at $e' = 64$ (176 of 4000) to 18 % at $e' = 321$ (30 of 4000); the Rust points carry $1/\sqrt{3000}$, 1.8 %.

Measured on the build of Section 8, $\lambda$ rounds cost 11.0/31.5/72.5 megacycles against 1.6/4.6/10.8 for the $t$ rounds shipped, and would take a cold verification from 21.6/50.0/120.4 megacycles to 31.1/76.9/182.1, 44 / 54 / 51 % more. A tighter count for composites of the form $2^{2e'-1} - s^2$ under known bases is left open.

**The salt** With $\gamma$ the density over the full digest range, the bound of [BBC+26a, Equation 18], $k \ge \lambda + 64 - \log_2 \gamma$, is 200/264/328 bits. The paper's conservative $k = 2\lambda = 256/384/512$ [BBC+26a, after Remark 13] is kept as the default, because the bound rests on a heuristic density; the shorter salt is a parameter worth 7 / 15 / 23 bytes.

**Nakagawa–Onuki** The attack of [NO26, Theorem 2] forges a PRISM-id response in $\tilde{O}(2^{6a/7})$ when the attacker may refuse challenge primes; in hash-and-sign, refusing a challenge is grinding the hash input, so the bound carries over, and [BBC+26a, Remarks 9 and 12] record it as $a \ge 7\lambda/6$. With the ECLIPSE degree's $2e' - 1$ bits in place of $a$ the exponent is 330/440/549, assuming divisibility of $\Phi_k(-p)$ is independent of the form $2^{2e'-1} - s^2$; their conservative $a \ge 4\lambda/3$ [NO26, Section 5.3] reads $e' \ge 86/129/172$ here. Table 3 collects the parameters with the source of each row.

Table 3: Parameters per level, with the source of each row.

| | NIST I | NIST III | NIST V |
|---|---|---|---|
| $p$[a] | $3 \cdot 2^{324} - 1$ | $27 \cdot 2^{500} - 1$ | $17 \cdot 2^{664} - 1$ |
| $\lceil \log_2 p \rceil$[a] | 326 | 505 | 669 |
| $e$ (two-adic exponent of $p+1$)[a] | 324 | 500 | 664 |
| $\lambda$[b] | 128 | 192 | 256 |
| $n = a = \lambda + 64$ (digest bits)[b] | 192 | 256 | 320 |
| $e' = a + 1$ (torsion exponent)[c] | 193 | 257 | 321 |
| bits of $d = 2^{2e'-1} - s^2$[c] | 385 | 513 | 641 |
| $k = 2\lambda$ (salt bits)[f] | 256 | 384 | 512 |
| torsion of bases and signature points, $e' + 2$[c] | 195 | 259 | 323 |
| $t$, Miller–Rabin rounds on $d$[d] | 18 | 28 | 38 |
| (salt-PRISM: rounds on $q$)[d] | 22 | 39 | 55 |
| public key, bytes[e] | 83 | 129 | 169 |
| ECLIPSE signature, bytes (wire format v3)[e] | 206 | 300 | 388 |
| salt-PRISM signature, bytes (wire format v3)[e] | 231 | 333 | 429 |

[a] The SQIsign round-3 specification [AAA+26].
[b] [BBC+26a, Equation 18 and the text after it]: $n \geq \lambda + \log_2 N_{\text{sign}}$, $N_{\text{sign}} = 2^{64}$, $a = n$.
[c] This work, Section 5: $e' = a + 1$, from which the bit length of $d$ and the torsion $e' + 2$ follow.
[d] The model of [CGMT26, Section 5.1], re-run for the ECLIPSE degree (Section 5); salt-PRISM from its Table 4.
[e] This work, Section 6.
[f] The conservative choice $k = 2\lambda$ of [BBC+26a], paragraph after its Remark 13; Equation 18 alone gives $k \geq \lambda + 64 - \log_2 \gamma$ (Section 5).

## 6 Encoding

Strong unforgeability is a statement about bytes: a second byte string that verifies is a forgery whether or not it encodes a new isogeny. So every freedom in the representation gets one rule, and the decoder checks them all.

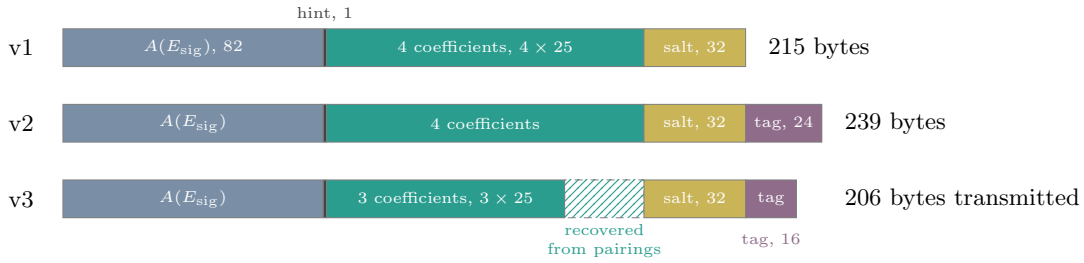



Figure 5: Byte layout of the ECLIPSE signature at level I in the three wire formats, segment widths proportional to bytes. v1: the curve, the hint, four coefficients of $\lceil (e'+2)/8 \rceil$ bytes and the salt. v2 adds an $n$-bit tag. v3, the shipped format, transmits three coefficients, recovers the fourth from two Weil pairings (hatched, drawn at its width but not transmitted) and carries a $\lambda$-bit tag of 16 bytes.

**The format** Figure 5 is the byte layout and Table 4 the fields in wire order. Every bit of a coefficient is part of the object: by Lemma 1 an accepted pair of points is $\pm[u]\sigma(P_{\text{vk}}, Q_{\text{vk}})$ exactly at order $2^t$, so a point shifted by the 2- or 4-torsion above the kernel is rejected and no coefficient bit is redundant.

Table 4: The ECLIPSE signature field by field, in wire order, with the byte width of each field at NIST levels I, III and V.

| field (wire order) | content | I | III | V |
|---|---|---|---|---|
| $A(E_{\text{sig}})$ | Montgomery coefficient of the signature curve, two field elements | 82 | 128 | 168 |
| hint | one byte naming the deterministic basis $(P, Q)$ of $E_{\text{sig}}[2^t]$ | 1 | 1 | 1 |
| $m_{00}$ | first coefficient of $M$: $P_{\text{sig}} = [m_{00}]P + [m_{10}]Q$ | 25 | 33 | 41 |
| $c$ | two more coefficients: $(m_{01}, m_{10})$ if $m_{00}$ is odd, else $(m_{10}, m_{11})$; $Q_{\text{sig}} = [m_{01}]P + [m_{11}]Q$ | 50 | 66 | 82 |
| salt | $k$ bits, hashed with the key and the message | 32 | 48 | 64 |
| tag | $\lambda$ bits of the digest's high half | 16 | 24 | 32 |
| total | $2\,\text{FP} + 1 + 3\lceil (e'+2)/8 \rceil + k/8 + \lambda/8$ | 206 | 300 | 388 |

**The freedoms and their rules** Before any rule an isogeny has several byte representations: the sign of the matrix ($M$ and $-M$ give the same Kummer points), the Montgomery model of $E_{\text{sig}}$, the basis the coefficients refer to, the fourth coefficient, which the other three determine, and the automorphisms of $E_{\text{sig}}$, which carry an isogeny to another with the same kernel. Figure 6 pairs each with its rule. The first entry of $M$ that negation modulo $2^{e'+2}$ moves, that is neither 0 nor $2^{e'+1}$, must be below $2^{e'+1}$, SQIsign's NormalizeMatrix rule, so $-M$ is rejected. $A(E_{\text{sig}})$ must equal the coefficient that NormalizeMontgomery [AAA+26, Algorithm 4.96] returns from a 4-torsion point of the hint basis, and $A = \pm 2$ is rejected. That coefficient is a function of the $\mathbb{F}_{p^2}$-isomorphism

class of $E_{\text{sig}}$: the algorithm evaluates $2(x^4+z^4)/(x^4-z^4)$ at the coordinates $(x:z)$ of a point $T$ of order 4 with $[2]T \neq (0,0)$ and at the two null points $(x+z:x-z)$ and $(ix+z:x+iz)$ derived from it, which are the Montgomery coefficients of the three models of $E_{\text{sig}}$ with one of its three 2-torsion points at $(0,0)$, adds their negatives, the models with $x \mapsto -x$, and keeps the largest of the six under the total order of $\mathbb{F}_{p^2}$; another choice of $T$ above the same 2-torsion point replaces $x$ by $x$ or by $1/x$, which negates the value, and a point above another 2-torsion point permutes the three, so the set of six is the set of Montgomery coefficients of the models of $E_{\text{sig}}$ and its maximum is one coefficient per class. Curves with $j \in \{0, 1728\}$, whose automorphism group is larger than $\{\pm 1\}$, are rejected by the decoder, so an accepted $E_{\text{sig}}$ has $\mathrm{Aut}(E_{\text{sig}}) = \{\pm 1\}$ and an isogeny into it is determined by its kernel up to sign; the signer draws a fresh salt on such a codomain and key generation resamples such a key. The hint must be the deterministic hint of the curve. The fourth coefficient is recovered, not transmitted. The rest is what any strict decoder has: field elements below $p$, coefficients below $2^t$, exact lengths, fixed framing. Encoding what the decoder accepts returns the input byte string (Lemma 5).

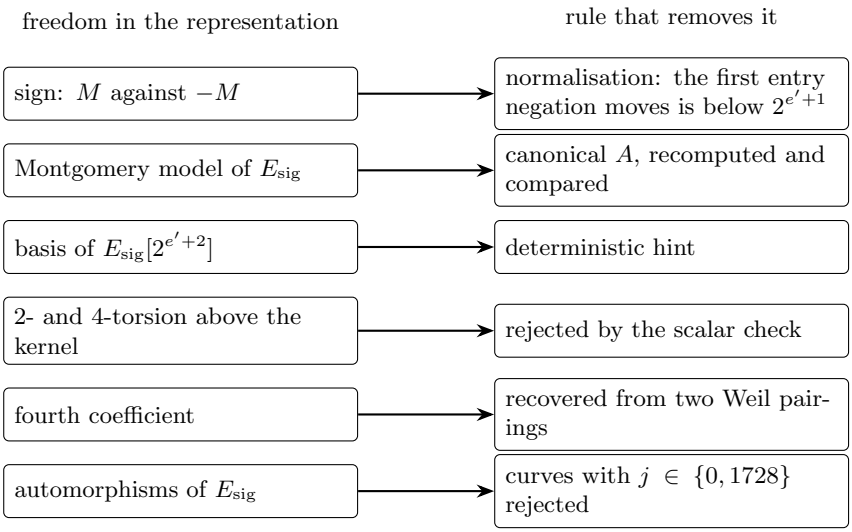



Figure 6: Each freedom in the representation of a signature (left) and the rule that removes it (right): the wire format's normalisation, canonical curve model and deterministic hint, the verifier's scalar check, the pairing recovery of the fourth coefficient, and the rejection of the two curves with extra automorphisms.

**The fourth coefficient** The signer's points are $[s^{-1}]\sigma(P_{\text{vk}}, Q_{\text{vk}})$, so by bilinearity of the Weil pairing at order $2^t$, $e(P_{\text{sig}}, Q_{\text{sig}}) = e(P_{\text{vk}}, Q_{\text{vk}})^{ds^{-2}}$, and $d \equiv -s^2 \pmod{2^{2e'-1}}$ with $t \leq 2e'-1$ makes the exponent $-1$ modulo $2^t$. With $e(P_{\text{sig}}, Q_{\text{sig}}) = e(P,Q)^{\det M}$ on the hint basis, $\det M \equiv -\log_{e(P,Q)} e(P_{\text{vk}}, Q_{\text{vk}}) \pmod{2^t}$, an odd number: two Weil pairings and one 2-adic discrete logarithm give the determinant, and the fourth coefficient follows. The identity holds at exactly the coefficients' modulus $2^t$, so the recovered coefficient is determined for every honest signature. In SQIsign the kernel is generated by the [4]-multiples of the basis, so the relation holds only to the fourth power and fixes the determinant two bits short; that gap is where the residual bits of a pairing-compressed SQIsign signature come from (the format of [Ray26d] transmits four); here there are none. Since $\det M$ is odd, each row has an odd entry, so the parity of $m_{00}$ says which three entries are written, and a triple with no odd pivot is rejected before any curve work; the normalisation is checked after

recovery, so the negated triple is rejected there. The decoder needs $e(P_{\mathrm{vk}}, Q_{\mathrm{vk}})$, so it takes the public key: the same bytes under another key are another object.

**The tag** The digest is squeezed to $2n$ bits; the low half gives $s$ and $\lambda$ bits of the high half ride in the signature, compared in constant time before any isogeny work. This is the device of [AAA+26, Section 8.4.1]: a signer wanting one signature valid for two messages under one key needs two $2n$-bit digests to agree, so message binding stands at $(n+\lambda)/2 = 160/224/288$ bits; the tag width is a parameter. The hash input includes the public curve, salt-PRISM's choice for non-resignability [BBC+26a, Sections 4.4 and 6]. What remains is inherent to hash-and-sign: a second preimage costs about $2^n/\gamma$ evaluations, the $n \geq \lambda + 64$ design point, and a signer can find two of its own messages with one digest in about $2^{n/2}/\sqrt{\gamma}$ offline.

**Sizes** Signatures are 206 / 300 / 388 bytes in the default format and 199 / 285 / 365 with the salt at the bound of Theorem 2 of [BBC+26a]; public keys 83 / 129 / 169. Section 11 places them against SQIsign.

# 7 Implementation

The Rust implementation and the SageMath model accept each other's signatures, and the salt-PRISM layer accepts the signatures of the authors' C implementation [BBC+26b] given the same basis. The codomain-reroute attack of [Ray26a, Section 6, Proposition 6.1], run at the level I prime where the 3-torsion is rational, forges salt-PRISM: on a signature whose auxiliary degree $2^a - q$ is divisible by 3, three of the twelve candidate replacements of the degree-3 factor verify. Against ECLIPSE none of the twelve verifies, each failing the null-point match. The artifact's test suite [Ray26c] contains the known-answer, tampering, mutation and fuzzing tests of the decoder and the verifier.

The proved statements are Lemma 1, Theorem 3 and Lemmas 4 and 5; the tests above are tests of the implementation.

**Strong unforgeability** Write $N = 2^{2e'-1}$, $n = e' - 1$, $\mathrm{Squares}_{e'} = \{\, s \text{ odd},\ 0 < s < 2^{e'-1} : N - s^2 \text{ prime} \,\}$ and $\gamma = \#\mathrm{Squares}_{e'}/2^n$, as $\gamma_{n,a} = \#\mathrm{Primes}_a/2^n$ in [BBC+26a, Theorem 2]; the odd $s$ below $2^{e'-1}$ number $2^{n-1}$. The hash $H$ of Algorithm 1, line 4, to $2n$ bits is a random oracle; $s(h)$ is the odd $s$ read off a digest $h$ and $\mathrm{tag}(h)$ its tag bits. As in [BBC+26a, Theorem 2] the membership test is idealised: Hsquares outputs $s(h)$ exactly when $s(h) \in \mathrm{Squares}_{e'}$. The signing stream SHAKE256("ECLIPSE-det-v1" $\|\, sk \,\|\, msg$) is modelled as a random function of $msg$ that only the holder of $sk$ evaluates, so a fresh message gets fresh uniform salts and a repeated message the same ones. The SUF-CMA game is [BBC+26a, Definition 4]: $A$ receives $vk$, obtains signatures $\sigma_i$ on messages $msg_i$ of its choice, and wins with $(msg^*, \sigma^*)$ accepted by Verify and different, as a pair of byte strings, from every $(msg_i, \sigma_i)$.

**Definition 2 ([BBC+26a, Definition 6 and Problem 4])** *A special squared degree isogeny oracle, SPEDIO$_2$, takes a supersingular curve $E$ over $\mathbb{F}_{p^2}$ and $s \in \mathrm{Squares}_{e'}$ and returns a uniformly random cyclic isogeny of degree $N - s^2$ from $E$. Problem 4: given a random supersingular curve $E$ and a SPEDIO$_2$, output an isogeny of degree $N - \bar{s}^2$ with $\bar{s} \in \mathrm{Squares}_{e'}$, different from all the isogenies the oracle generated.*

As [BBC+26a] does for its Problem 2, we take the curve $E$ of Problem 4 from the distribution that the key generation of Section 4 produces, $E_0/I_{sk}$ for a random ideal of prime norm; the gap to

a uniformly random supersingular curve is the one [BBC+26a] carries. Problem 4 is an instance of Problem 5.1 of the open-problems survey of Castryck, De Feo, Galbraith, Kutas, Reijnders and Wesolowski, for which no algorithm is known [CFG+26, Problem 5.1]. Isogenies are counted by their kernels, and the oracle's representation allows evaluation on $E[2^t]$, as the lists of isogenies in the reductions of [BBC+26a] do. $\mathrm{Adv}^{\mathrm{P4}}(B)$ is the probability that $B$ outputs a valid solution.

**Theorem 3 (SUF-CMA of ECLIPSE under Problem 4, ROM)** *Let $A$ be a SUF-CMA adversary against ECLIPSE making at most $N_{\mathrm{sign}}$ signing queries and $N_H$ queries to $H$. In the random oracle model, with the membership test idealised and the signing stream modelled as above, there is an adversary $B$ against Problem 4 that runs $A$ once and makes at most $N_{\mathrm{sign}}$ oracle queries plus one for each draw whose codomain has $j \in \{0, 1728\}$, with*

$$\mathrm{Adv}^{\mathrm{SUF}}(A) \le \mathrm{Adv}^{\mathrm{P4}}(B) + \frac{N_{\mathrm{sign}}(N_{\mathrm{sign}}-1)}{2\gamma 2^n} + \frac{(N_H+1)N_{\mathrm{sign}}}{2^{n-1}} + N_{\mathrm{sign}} 2^{-(2e'-3)}.$$

The second term is the chance that two signing queries with different message and salt hash to the same $s$, the third that the forgery's message and salt, when not a signed pair, hash to the $s$ of a signed one, both in the quantities of [BBC+26a, Equation 12]; the last is the distance of the honest signer's isogeny from uniform (fact (i)). There is no term in $k$: the reduction never programs the oracle. The probability that the verifier accepts a composite $d$ (Section 5) is omitted from the bound by the idealisation. Theorem 3 does not use the salt; it is kept for existential unforgeability under the non-interactive Problem 3 of [BBC+26a] through its Theorem 2 and Remark 11, which does. The implemented Miller–Rabin test replaces the idealised one; the remarks below give the two terms this adds.

*Proof. The reduction.* $B$ receives $E$ and the oracle, sets $E_{\mathrm{vk}} = E$, derives the hint and the basis $(P_{\mathrm{vk}}, Q_{\mathrm{vk}})$ of $E_{\mathrm{vk}}[2^t]$ as key generation does (public functions of the curve) and runs $A$ on $vk$. Oracle queries to $H$ are answered by the random oracle, with no programming. A repeated signing query gets the stored answer. A fresh one is answered as the signer would: draw $\mathrm{salt} \leftarrow \{0,1\}^k$, compute $h = H(E_{\mathrm{vk}}, msg_i, \mathrm{salt})$, repeat until $s_i = s(h) \in \mathrm{Squares}_{e'}$; query the oracle at $s_i$ for a uniform isogeny $\varphi_i \colon E_{\mathrm{vk}} \to E_i$ of degree $d_i = N - s_i^2$; with $u_i = s_i^{-1} \bmod 2^t$ return $\sigma_i = \mathrm{Encode}(E_{\mathrm{vk}}, E_i, [u_i]\varphi_i(P_{\mathrm{vk}}, Q_{\mathrm{vk}}), \mathrm{salt}, \mathrm{tag}(h))$, $E_i$ in canonical form. On $A$'s output $(msg^*, \sigma^*)$, $B$ decodes $\sigma^*$, computes $s^* = s(H(E_{\mathrm{vk}}, msg^*, \mathrm{salt}^*))$ and outputs the isogeny $\sigma^*$ represents: by Lemma 1, an accepted $\sigma^*$ determines an isogeny $E_{\mathrm{vk}} \to E_{\mathrm{sig}}$ over $\mathbb{F}_{p^2}$ of degree $N - s^{*2}$ with the transmitted images, which $B$ computes in the representation of Section 6 itself; being over $\mathbb{F}_{p^2}$, it is an isogeny "originating from $E$" in the sense of Problem 4.

*The simulation.* Two facts, argued from the code in the paragraphs after the proof. (i) The honest signer's isogeny on a digest $s$ is uniform over the $d+1$ isogenies of degree $d = N - s^2$ from $E_{\mathrm{vk}}$ up to statistical distance $2/(d+1) < 2^{-(2e'-3)}$, as [BBC+26a, Algorithm 2 and Remark 3] prescribe; the oracle's is exactly uniform. (ii) Under a fixed key and digest, an accepted byte string is a function of the isogeny it represents (Lemma 5). Salts, digests, tags, the canonical codomain and the reduction of the points to order $2^t$ are produced as the signer produces them, so $A$'s view is within $N_{\mathrm{sign}} 2^{-(2e'-3)}$ of the real game. When the oracle's isogeny has a codomain with $j \in \{0, 1728\}$, $B$ discards it and re-draws the salt as the signer does, so $B$ makes at most $N_{\mathrm{sign}}$ plus the number of such draws oracle queries and its output must differ from the discarded isogenies as well; each draw is a $24/p$ event, and the bound is unchanged up to that term.

*The excluded events.* $E_1$: two signing queries with different $(msg_i, \mathrm{salt}_i)$ have $s_i = s_j$; an accepted digest is uniform on $\mathrm{Squares}_{e'}$, so $\Pr[E_1] \le N_{\mathrm{sign}}(N_{\mathrm{sign}}-1)/(2\gamma 2^n)$. $E_2$: $(msg^*, \mathrm{salt}^*)$ is no signed pair but $s^* = s_j$ for some $j$; $s(h)$ of a fresh oracle query is uniform over $2^{n-1}$ values

independent of the $s_j$, and $A$ learns $s^*$ only through its $N_H$ queries or the verification of its forgery, so $\Pr[E_2] \le (N_H+1)N_{\text{sign}}/2^{n-1}$.

*The three cases.* Let $A$ win outside $E_1 \cup E_2$; $s^* \in \text{Squares}_{e'}$ by the idealised test, so $B$'s output has a degree of the required form, and it remains to see that it is none of $\varphi_1, \ldots, \varphi_{N_{\text{sign}}}$. (1) $s^* \notin \{s_j\}$: the output's degree was never returned. (2) $(msg^*, \text{salt}^*) = (msg_i, \text{salt}_i)$: the same digest, and $\sigma^* \neq \sigma_i$ as byte strings, so by (ii) a different isogeny from $\varphi_i$; outside $E_1$ no other $\varphi_j$ has degree $d_i$. (3) $(msg^*, \text{salt}^*)$ no signed pair with $s^* \in \{s_j\}$: this is $E_2$. So outside $E_1 \cup E_2$ and the simulation's distance, $A$'s win is $B$'s. □

**Lemma 4 (Uniqueness)** *Let $\sigma, \sigma' \colon E_{\text{vk}} \to E_{\text{sig}}$ be isogenies of the same odd degree $d < 2^{2t-2}$ with $\sigma' = \pm\sigma$ on $E_{\text{vk}}[2^t]$, one sign for all points. Then $\sigma' = \pm\sigma$.*

*Proof.* With $\varepsilon$ the common sign, $\sigma' - \varepsilon\sigma$ vanishes on $E_{\text{vk}}[2^t]$, so $\sigma' - \varepsilon\sigma = [2^t]\beta$ for some $\beta \in \text{Hom}(E_{\text{vk}}, E_{\text{sig}})$. The degree is a positive definite quadratic form on $\text{Hom}(E_{\text{vk}}, E_{\text{sig}})$, so $\deg(\sigma' - \varepsilon\sigma) \le (\sqrt{d} + \sqrt{d})^2 = 4d < 2^{2t}$, hence $\deg\beta < 1$, $\beta = 0$ and $\sigma' = \varepsilon\sigma$. □

For an accepted string, Lemma 1 gives a degree-$d$ isogeny agreeing with $\pm[s](P_{\text{sig}}, Q_{\text{sig}})$ on the hint basis, and $d < 2^{2e'-1} < 2^{2t-2}$; by Lemma 4 it is unique up to sign, so "the isogeny $\beta$ represents" is a function of the string.

**Lemma 5 (Encoding)** *Fix $vk = (E_{\text{vk}}, \text{hint})$ and a digest $(s, \text{tag})$ with its salt, and let $\sigma \colon E_{\text{vk}} \to E_{\text{sig}}$ be an isogeny of degree $d = N - s^2$. At most one byte string accepted by the decoder under $vk$ and by* Verify *with this digest represents $\sigma$, namely* $\text{Encode}(E_{\text{vk}}, E_{\text{sig}}, [u]\sigma(P_{\text{vk}}, Q_{\text{vk}}), \text{salt}, \text{tag})$ *with $E_{\text{sig}}$ canonical; and two distinct accepted strings with this digest represent two distinct isogenies.*

*Proof.* Let $\beta$ be accepted and $\sigma_\beta$ the isogeny it represents, with $\sigma_\beta(P_{\text{vk}}, Q_{\text{vk}}) = \varepsilon[s](P_{\text{sig}}, Q_{\text{sig}})$. Every field of $\beta$ is a function of $(\sigma_\beta, s, \text{salt}, \text{tag})$: the curve bytes are the canonical coefficient of the codomain $E_{\text{sig}}$, one per isomorphism class (Section 6); the hint is the deterministic hint of $E_{\text{sig}}$, which fixes the basis $(P, Q)$ of $E_{\text{sig}}[2^t]$; the coefficient matrix $M$ of $(P_{\text{sig}}, Q_{\text{sig}})$ in $(P, Q)$ is $\pm M_\sigma$, the matrix of $[u]\sigma_\beta(P_{\text{vk}}, Q_{\text{vk}})$, its entries are residues below $2^t$, the omitted entry and its position are functions of the other three and of $vk$, and the sign rule accepts exactly one of $M_\sigma$ and $-M_\sigma$, since an odd-determinant matrix has an entry negation moves; the salt and the tag are the digest's; the length is exact. So $\beta$ is the one string named, and distinct accepted strings on one digest have distinct $\sigma_\beta$, hence distinct kernels: the decoder accepts $E_{\text{sig}}$ only with $j \notin \{0, 1728\}$, so $\text{Aut}(E_{\text{sig}}) = \{\pm 1\}$ and two isogenies with one kernel differ by the sign the encoding already quotients. □

**Fact (i), from the code** GenIsogenyStrong (Algorithm 1, lines 8 to 10) samples a left $O_0$-ideal of norm $d$, intersects it with the secret ideal and translates, as [BBC+26a, Algorithm 2] does with its RandomIdealGivenNorm:

```
// crates/eclipse-rs/src/eclipse.rs, gen_isogeny_strong
let ichall = QuatIdeal::random_given_prime_norm(&d, alg, rng)?;
let inter = QuatIdeal::intersect_o0(&ichall, &sk.secret_ideal, None);
let out = ideal_to_isogeny_qlapoty::<L, N>(&inter, alg, act, rng,
                                           stats)?;
```

The map from left $O_0$-ideals of norm $d$ to isogenies of degree $d$ from $E_{\text{vk}}$, the push-forward along $\varphi_{sk}$, is a bijection because $d$ is coprime to $n(I_{sk})$, and a uniform ideal gives a uniform isogeny: this is [BBC+26a, Remark 3], "the output of GenIsogeny is a uniformly random cyclic isogeny" from $E_{\text{vk}}$. What remains is the uniformity of our sampler, which is not RandomIdealGivenNorm:

```
// crates/eclipse-rs/src/quat/ideal.rs, random_given_prime_norm
loop {
    x = Ibz::rand_interval(&Ibz::zero(), &nm1, rng)?; // uniform in [0, d-1]
    x = x.add(&x);                                    // 2x
    tmp = x.mul(&x).modulo(&n4);                      // 4x^2 mod 4d
    tmp = alg.p.add(&tmp).neg().modulo(&n4);          // -(p + 4x^2) mod 4d
    rp = tmp.modulo(norm);
    if rp.legendre(norm) == 1 && (tmp.get() & 3) < 2 { break; }
}
rp = rp.sqrt_mod_p(norm, rng).expect("square residue");
...
let bit = Ibz::<N>::rand_interval(&Ibz::zero(), &Ibz::one(), rng)?;
if bit.get() & 1 == 1 { y = y.neg(); }              // one of the two roots
...
Some(Self::from_xyn(&x, &y, norm))
```

The ideal is $O_0 d + O_0\gamma$ for $\gamma = x + \frac{2y+1}{2}i + \frac{1}{2}j$, whose norm $x^2 + (2y+1)^2/4 + p/4$ is divisible by $d$ because $(2x)^2 + (2y+1)^2 \equiv -p \pmod{4d}$. With $X = 2x$ and $Y = 2y+1$ modulo $d$, the ideals of this shape correspond one to one to the points $(X, Y)$ of the conic $X^2 + Y^2 \equiv -p \pmod{d}$; as $d \equiv 7 \pmod 8$, $-1$ is a non-residue modulo $d$, the conic has $d+1$ points and every ideal of norm $d$ has this shape. The sampler draws $X$ uniformly, accepts exactly when $-p - X^2$ is a non-zero square modulo $d$, and returns one of the two points above $X$ with a fair bit, so by rejection sampling its output is uniform over the conic points with $Y \neq 0$; the points with $Y = 0$, two when $-p$ is a square modulo $d$ and none otherwise, are never returned. The output is therefore uniform over $d+1$ or $d-1$ of the $d+1$ ideals, at statistical distance 0 or $2/(d+1) < 2^{-(2e'-3)}$ from uniform. The condition `(tmp & 3) < 2` always holds ($p \equiv 3 \bmod 4$ gives $-(p + 4x^2) \equiv 1 \bmod 4$) and adds no bias. This is the paper's "one square root modulo $d$ with no rejection": the loop rejects about half the $x$, not the output.

**Remarks on the theorem**

- *The implemented membership test.* With $t_{\mathrm{MR}}$ Miller–Rabin rounds on random bases in place of the idealised test, two terms join the bound: $\varepsilon_{\mathrm{MR}}$, the probability that the forgery's degree is composite yet accepted, which is at most $4^{-t_{\mathrm{MR}}}$ over the verifier's bases whatever the forger chose (Monier and Rabin, as [CGMT26, Lemma 1] states it), $2^{-36}/2^{-56}/2^{-76}$ at the shipped $t_{\mathrm{MR}} = 18/28/38$; and $N_{\mathrm{sign}}\varepsilon_{\mathrm{sign}}$, the chance that an honest signing accepts a composite and the oracle query is ill-formed, where the signer's candidates are random so the bound of [CGMT26, Theorem 2] after [DLP93] applies, below $2^{-127}/2^{-189}/2^{-253}$ with the measured 72/95/119 candidates per signature. The worst case $4^{-t_{\mathrm{MR}}}$ dominates the bound; the model of [CGMT26, Section 5] that fixed $t_{\mathrm{MR}}$ charges the forger in addition for a composite smooth enough to build an isogeny of its degree without the endomorphism ring, a heuristic cost outside the theorem. A verifier at $\lambda$ rounds makes the term $2^{-2\lambda}$ at 11.0/31.5/72.5 megacycles, 44 / 54 / 51 % of a cold verification (Section 5).
- *Problem 5.* The oracle-free Problem 5 of [BBC+26a] draws its target independently of the given isogenies, so the reused-digest case (2) is not an instance of it except on the coincidence $\bar{s} = s_i$; Problem 4 covers it through the oracle.
- *Determinism.* Deterministic signing for a given degree, which [BBC+26a, Remark 8 and Section 5.1] impose for the reduction from the list of Problem 3, is not needed under Problem 4: the oracle answers each digest afresh and "different from all the isogenies formerly generated" is the

winning condition. Our signer is deterministic anyway: in the default mode the salt, the sampler's $x$ and sign bit and the translation's randomness all come from the stream of $(sk, msg)$, so the signature is a function of $(sk, msg)$ and a fortiori of $(sk, msg, \text{salt})$; in the randomised mode they come from one stream seeded by fresh entropy, so the isogeny is not a function of the salt alone.

- *ROM only.* The quantum random oracle bound of [BBC+26a, Equation 13] is not transported.

# 8 Performance

**Method** One session on one machine, an Intel Xeon at 2.80 GHz (8 vCPUs, a virtual machine), on 2026-10-07, in `rdtsc` megacycles. Our code is the default release build (fat link-time optimisation, no target-CPU flag), whose x86-64 field arithmetic is inline assembly in the reference's `mulx/adcx/adox` schedule, chosen at run time; `criterion` means, 100 samples for verification and 1000 for signing, each signing sample one signature on a fresh message, so the mean, median and maximum of Table 5 are of the same 1000 signatures. The C reference is the-sqisign at tag `nist-v3` [The26], built `Release` with `-DSQISIGN_BUILD_TYPE=broadwell`, its assembly-backed arithmetic, measured by its own benchmark at 1000 iterations per level, means with its medians and maxima. The Rust port's SQIsign verifier runs in the same binary as a control: it verifies at 1.08 / 1.09 / 1.10 times the reference, so a ratio against that port is a ratio on identical arithmetic. Verification is from bytes: decode the key, decode the signature, verify.

**Cycles** Table 5 gives signing and verification; key generation is SQIsign's with one basis matrix added and is not tabulated. ECLIPSE signs 2.6 / 2.4 / 2.8 times faster than the reference's assembly build and 2.3 / 2.1 / 2.7 times faster than salt-PRISM, on means; the medians and maxima say the same, and the signing tail is the subject of the next paragraph. Its verification costs 2.0 / 1.7 / 1.6 times the reference's cold and 1.6 / 1.4 / 1.3 with the public key prepared once (basis, canonical-model check, pairing and discrete-logarithm table, 18 % of a verification where the same preparation buys SQIsign 3 %); against that port, 1.9 / 1.5 / 1.4 cold and 1.6 / 1.3 / 1.2 prepared. Table 6 says what the work is: the forward chain has about the $(2, 2)$-steps of either scheme's response and the pull-back adds a tenth, so the difference is fixed work, the decoder's two Weil pairings, the three ladders of the scalar check and a membership test on a number twice as long.

**Signing** The two signers share every phase but two. The challenge ideal: our prime-norm sampler costs 0.4 megacycles at level I, one square root modulo $d$ with no rejection, where salt-PRISM's composite norm $q(2^a - q)$ goes through the reference's arbitrary-odd-norm sampler [The26] at 69 megacycles, about $\ln N$ draws each with a primality test and a square root at $2a$ bits. The hash: Hsquares tests numbers of $2e' - 1$ bits with $t = 18/28/38$ rounds where salt-PRISM tests $a$-bit numbers with 22/39/55 [CGMT26, Table 4], a small factor on a small phase. Net, ECLIPSE signs faster at every level and the gap grows with the level. The same two phases explain the tails in Table 5: ECLIPSE's only rejection loop is Hsquares, cheap against the chain, so over the 1000 level I signatures its maximum is 70 megacycles against a median of 34.7; salt-PRISM's sampler loop gives 465 against 71.2, and SQIsign's own tail is 177 against 91.8.

**Verification** A verifier's cost is work that scales with the chain, a fixed number of $\mathbb{F}_{p^2}$ operations per step, plus work that does not: the basis from the hint, the pairings, the decoding and the hash. Round-2 level V and round-3 level III share the prime $27 \cdot 2^{500} - 1$ with $a = 320$ and 256 [AAA+25, AAA+26], so two salt-PRISM runs there, differing only in chain length, separate the

Table 5: Megacycles in one session (2026-10-07). Signing: mean, median and maximum of 1000 signatures per scheme and level, the C reference's assembly build by its own benchmark and our default build by `criterion`. Verification: means, ECLIPSE cold and with a prepared key.

| | NIST I | NIST III | NIST V |
|---|---|---|---|
| signing, mean (median; maximum) | | | |
| SQIsign (C) | 95.1 (91.8; 177) | 343.0 (321.2; 890) | 706.1 (669.9; 1623) |
| SQIsign (Rust) | 102.6 (98.9; 177) | 395.9 (364.0; 1374) | 772.8 (737.1; 1698) |
| salt-PRISM | 82.7 (71.2; 465) | 296.4 (261.8; 1353) | 673.7 (583.6; 2642) |
| ECLIPSE | 36.0 (34.7; 70) | 142.3 (119.1; 642) | 254.0 (236.0; 695) |
| verification, mean | | | |
| SQIsign (C) | 11.1 | 29.4 | 76.0 |
| SQIsign (Rust) | 12.0 | 32.1 | 83.7 |
| salt-PRISM | 17.0 | 37.7 | 88.9 |
| ECLIPSE, cold | 22.3 | 49.7 | 121.0 |
| ECLIPSE, prepared key | 18.2 | 40.9 | 100.7 |

Table 6: Work per verification: $(2,2)$-isogeny steps, pairings and the primality membership test, by scheme and level.

| | NIST I | NIST III | NIST V |
|---|---|---|---|
| ECLIPSE: $(2,2)$-steps, $e'$ full + $(e'-1)$ pull-backs | 193 + 192 | 257 + 256 | 321 + 320 |
| ECLIPSE: pairings | 0 | 0 | 0 |
| ECLIPSE: membership test on $d$, Mcycles | 1.63 | 4.50 | 10.73 |
| salt-PRISM: $(2,2)$-steps, $a$ | 192 | 256 | 320 |
| salt-PRISM: Weil pairings at $2^a$ | 2 | 2 | 2 |
| salt-PRISM: membership test on $q$, Mcycles | 0.56 | 1.59 | 3.99 |
| SQIsign v3 [AAA$^+$26]: $(2,2)$-steps, response | 196 | 302 | 400 |
| SQIsign v3 [AAA$^+$26]: 2-isogeny steps, challenge ($\lambda$) | 128 | 192 | 256 |

terms: slope 0.319 megacycles per step, intercept -2.5 megacycles, within noise of zero. The cost is the chain length, not the prime. ECLIPSE's $e'$ steps plus $e' - 1$ pull-backs at a tenth of a step, against SQIsign's response chain of 196/302/400 steps and challenge chain of $\lambda$ one-dimensional steps [AAA$^+$26], gives ratios near 1.1 for the chains alone; the measured 1.5 to 2.0 is the fixed work, of which the prepared key removes the part that depends on the key alone.

**Compression** Compression leaves the verifier's input unchanged; it adds two Weil pairings at $2^t$ and one 2-adic discrete logarithm inside the decoder. Measured at level I in one session, built before and after the change: the recovery alone is 4.6 megacycles, decoding rises by that amount, verification on the decoded signature is unchanged, and verification from bytes rises by 10.6 %, for 33 bytes.

**The second parameter set** Table 7 is ECLIPSE on MIKE's rigorous prime, 1000 interleaved runs per row, with the signing tail. Verification sits where the field arithmetic puts it, 32.7 megacycles cold against 24.1 at the shared prime in the session of Table 9. Key generation and signing cost about four times the shared prime's, with a long tail: the translation of the key's ideal into an isogeny searches the norm equation of IdealToIsogeny for a representable norm, and that search tries 8 112 values per signature at the cofactor 117 against 268 at the cofactor 3, two thirds of a signature (Appendix B). The table also gives MIKE's own key generation and shared secret, from its Rust implementation, at the shared prime and at both of its own sets. MIKE's cost on a prime is the limb count of the field, and the cofactor does not enter [BDD$^+$26, Remark J.3]: at the shared prime (326 bits, six limbs) MIKE costs 1.5 times its fast set (318 bits, five limbs) on key generation and 1.7 times on the shared secret, and 0.6 times its rigorous set (381 bits, seven limbs) on both.

Table 7: The second parameter set, megacycles. Top: ECLIPSE at MIKE's rigorous prime and at the shared prime, 1000 interleaved runs per row, mean (median); the trial count is the norm-equation search's second loop, averaged per call. Bottom: MIKE's own key generation and shared secret at its three sets, `criterion` means of 100 samples.

| ECLIPSE | $117 \cdot 2^{374} - 1$ | $3 \cdot 2^{324} - 1$ |
|---|---|---|
| key generation | 170.0 (117.5) | 40.7 (38.9) |
| signing | 168.2 (114.6) | 43.8 (41.0) |
| verification, cold / prepared | 32.7 / 24.7 (32.4 / 24.5) | |
| search trials, key generation | 7 634 | 268 |
| search trials, signing | 8 112 | |

| | set | key generation | shared secret |
|---|---|---|---|
| MIKE (Rust) | shared, $3 \cdot 2^{324} - 1$, $e = 228$ | 4.4 | 38.2 |
| MIKE (Rust) | rigorous, $117 \cdot 2^{374} - 1$, $e = 372$ | 7.7 | 64.3 |
| MIKE (Rust) | fast, $633 \cdot 2^{308} - 1$, $e = 228$ | 2.9 | 23.0 |

**Milliseconds and throughput** Table 8 gives wall-clock times at level I on two machines: the Xeon column is rdtsc cycles at the nominal 2.8 GHz on the assembly backend, the M4 Pro column is wall clock on the portable backend from a Criterion run of 2026-10-07, and no C reference was run on the Mac. The scheme ratios agree across the two: ECLIPSE verifies at 1.86 times SQIsign

and signs at 0.35 on the Xeon, 1.74 and 0.36 on the M4 Pro. Throughput at level I, a batch of 256 verifications under 8 prepared keys, is 149 verifications per second on one thread and 584 on seven.

Table 8: Milliseconds at level I. Xeon: rdtsc, assembly, 2.8 GHz, the session of Table 5. M4 Pro: wall clock, portable backend, no C reference run.

| scheme | operation | Xeon (ms) | M4 Pro (ms) |
|---|---|---|---|
| SQIsign v3, C reference (`broadwell`) | sign | 34.0 | not run |
| | verify | 4.0 | not run |
| SQIsign v3, this substrate | sign | 36.6 | 20.72 |
| | verify | 4.3 | 2.55 |
| salt-PRISM | sign | 29.5 | 15.93 |
| | verify | 6.1 | 3.31 |
| ECLIPSE | sign | 12.9 | 7.41 |
| | verify | 8.0 | 4.44 |
| | verify, prepared key | 6.5 | 3.65 |

## 9 Randomisable keys

Signatures with randomisable keys are defined in [BBC+26c, Definition 3] after Celi, Griffy, Hanzlik, Perez-Kempner and Slamanig [CGH+25]: RandPK$(pk, \rho)$ derives a public key from public data, RandSK$(sk, \rho)$ the matching secret key (our secret key carries its public key), and VerKey$(pk, sk)$ checks a pair. This section instantiates the three on ECLIPSE (Figure 7), keeps their unlinkability bound, and keeps the public key in the hash. No new mathematics is involved.

**RandPK, RandSK, VerKey** RandPK expands the SHAKE256 of "ECLIPSE-rk-path-v1", $A(E_{\text{vk}})$ and $\rho$ into two scalars and walks two hops from $E_{\text{vk}}$: on the current curve, the deterministic basis $(P_i, Q_i)$ of the $2^f$-torsion, the kernel point $P_i + [r_i]Q_i$, and the reference's chain of 4-isogenies with kernel $\langle [4](P_i + [r_i]Q_i)\rangle$, degree $2^{f-2}$, codomain in canonical form. The endpoint and its hint are the derived key, in the encoding of a fresh one; anyone can run it. RandSK follows the walk on the ideal side: per hop, the kernel is pulled back to $E_0$ through the current images, its ideal composed with the running secret ideal, replaced by an equivalent of odd prime norm so that the basis matrix exists, and translated; the codomain must equal the public walk's. Three translations in all. VerKey translates the derived key's ideal once and compares codomain, hint and matrix. At level I, RandPK costs 11.35 megacycles (a fifth of a key generation), RandSK 168.8 (3.26 key generations) and VerKey about 40. The authors' SageMath RandSK, which translates by a generic route, costs 4.99/6.54/7.78 key generations at the three levels and RandPK about 0.9, as relative running times [BBC+26c, Table 4].

**Walk length** Theorem 1 of [BBC+26c], a case of the bound on non-backtracking walks [BCC+23, Theorem 11] on the 2-isogeny graph, which is Ramanujan [BCC+23, Theorem 3] (first derived from Hecke operators by Pizer [Piz90, Piz98]), bounds the distance of a $k$-step walk's endpoint from the stationary distribution, and their Proposition 6 concludes statistical unlinkability against unbounded adversaries once $k \geq \log_2 p + 2\lambda + \log_2 k$. Two hops of $2^{f-2}$ give $k = 644/996/1324$ against the condition's 591/899/1191; one hop ($k = 322/498/662$) does not reach it. A backtrack at

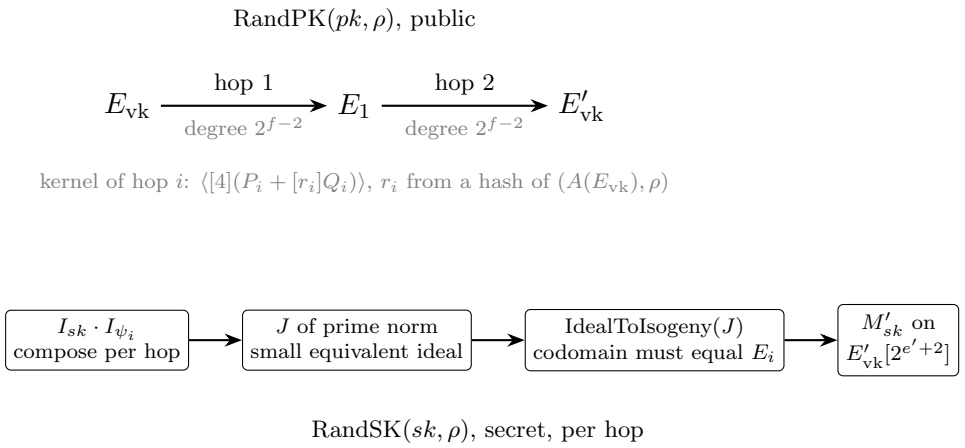



Figure 7: Randomisable keys. Top: RandPK walks two hops of degree $2^{f-2}$ from the public curve along kernels derived from a hash of the curve and the randomiser $\rho$; anyone can run it. Bottom: RandSK follows the same walk on the ideal side, replacing the composed ideal by an equivalent one of odd prime norm before each translation, and checks each codomain against the public walk.

the hop boundary cancels two steps, 642 against 591 at level I, within the margin. At the toy prime $5 \cdot 2^{14} - 1$ (Figure 8) one hop is at statistical distance 0.550 from stationary, two hops are within 0.004 of the sampling floor, four are at the floor. A signature reveals nothing about its key beyond the key [BBC+26a, Remark 3], and two derived keys signing one message share no salt, digest or isogeny. Unlinkability is claimed against observers who do not know $\rho$.

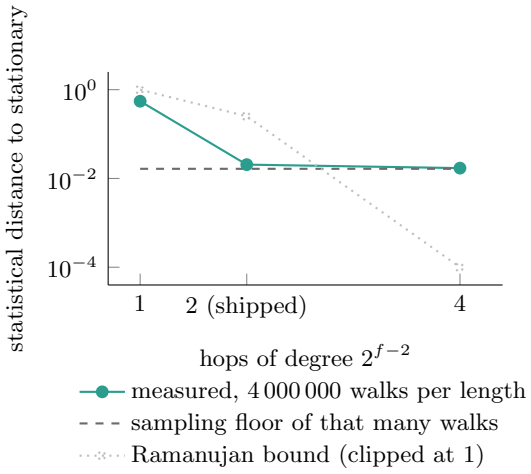



Figure 8: Statistical distance of the derived-key distribution to the stationary distribution at the toy prime $5 \cdot 2^{14} - 1$ against the number of hops, with the sampling floor and the bound; the shipped two hops sit on the flat part of the curve.

**The key in the hash** PRISM-RK has an adaptation algorithm [BBC+26c, Section 4.3] that moves a signature to a derived key without the signer, possible because its hash takes no key, $q \leftarrow H(msg, \text{salt})$ [BBC+26c, Figure 7, line 2]. Under a notion that counts (key, message, signature) tuples, adaptation is a forgery, so the hash input stays $(E'_{\text{vk}}, msg, \text{salt})$ and there is no adaptation:

an adapted isogeny has the old key's degree and the verifier under the new key demands the new one. The hop degree is then free of the restriction adaptation imposes.

**Profiles** Both profiles are RandPK and RandSK with a different source of $\rho$. Hierarchical derivation takes $\rho_i = \text{SHA3-256}(\text{“eclipse-hd”} \,\|\, pk_{\text{master}} \,\|\, i)$ for a 32-bit index: anyone with the master public key derives every child public key, the master secret derives the child secret, there is no chain code, and children are linkable to the parent by anyone holding it. A hardened child takes $\rho_i$ from the master secret key and the index under “eclipse-hd-hardened” instead, so only the secret's holder derives it. The domain strings and the path format are the implementation's current ones and are not yet frozen. Stealth addresses take a uniform $\rho$ of $2\lambda$ bits drawn by the sender, who pays to $\text{RandPK}(pk_{\text{recipient}}, \rho)$ and publishes $\rho$; the recipient scans with RandPK and spends with RandSK, unlinkable to observers without $\rho$. A one-time key is the stealth profile with a fresh $\rho$ per key, limited by the cost of RandSK, 3.26 key generations per key.

**Strong unforgeability under derived keys** The notion we target records (derived key, message, signature) tuples, so a second signature on a signed message counts under the same or another derived key, and the winning condition is over the derived curve, not over $\rho$. Theorem 3 extends by one step per case, pushing an instance isogeny forward along the public walk and pulling a forgery back along its dual, both exact because the degrees are coprime; the extension is sketched here, not written out.

## 10 One key, both roles

MIKE is a non-interactive key exchange whose public key is a supersingular curve over $\mathbb{F}_{p^2}$, published as one Montgomery coefficient [BDD+26, Rob24]; an ECLIPSE public key is such a curve with a hint byte. At a shared prime one curve holds both roles: MIKE's key agreement on it, untouched, and ECLIPSE signatures under it (Figure 9). The precedent is the group-action pair CSIDH and CSI-FiSh [CLM+18, BKV19], one secret ideal class and one public curve for both; here the key is a walk, the hard problem is the endomorphism ring, and both operations run in tens of megacycles. Each choice below is marked MIKE's (theirs) or ours. Appendix B has ECLIPSE on MIKE's own primes.

**A MIKE key (theirs)** MIKE's setup is a prime $p = 2^f c - 1$ with $c$ small and odd, an integer $e \le f - 2$, and a basis $(P_0, Q_0)$ of $E'_{M,0}[2^f]$, a Montgomery model of $E'_0 : y^2 = x^3 - x$ [BDD+26, Section 4.4]. Key generation samples a left $O_0$-ideal $I_A$ of norm $2^e$ by its generator $\gamma = (\omega - 1)(1 - x_A i)$ and evaluates the isogeny $\phi_A : E'_0 \to E_A$ it corresponds to; the public key is $E_A$ [BDD+26, Section 4.1]. Concretely (their Algorithm 1): the secret is $x_A \in \mathbb{Z}_{2^e}$ with $x_A \equiv 3 \pmod 8$, the kernel is $K_A = [2^{f-e}](P_0 + [x_A]Q_0)$, and $E_A$ is computed as a chain of 4-isogenies. The published coefficient is normalised over $E_A$ and its Frobenius conjugate $E_A^\sigma$ together, since the key represents the Weil restriction $E_A \times E_A^\sigma$ [BDD+26, Remark 4.1]; the implementation takes the smallest of the twelve coefficients [BDD+26, Section 6]. Key validation is a supersingularity test, an $\mathbb{F}_{p^2}$-rational basis of the $2^f$-torsion, which decides maximality when $2^{f-2} \ge c$ [BDD+26, Remark 4.2]. The shared secret is the absolute invariants of the codomain of a dimension-four $2^e$-isogeny from the other party's curve [BDD+26, Sections 4.2 and 4.3]; Pope's pseudocode [Pop26] is the same in three lines. The published level I sets are $633 \cdot 2^{308} - 1$ with $e = 228$ (fast) and $117 \cdot 2^{374} - 1$ with $e = 372$ (rigorous) [BDD+26, Table 1].

**The shared prime (theirs, then ours)** Section 5 quotes MIKE's setup, its own listing of the SQIsign primes [BDD+26, Appendix J.4] and Damien Robert's reply [Rob26]. Our configuration is MIKE unchanged on ECLIPSE's level I prime $3 \cdot 2^{324} - 1$ with its fast walk length $e = 228$, and ECLIPSE as in Section 5. The set's constants, $E_0$ in its Montgomery model and the basis of $E_0[2^{230}]$, are what MIKE's SageMath code derives from $(f, c, e) = (324, 3, 228)$ as for its own sets [MIK26b]; MIKE's Rust and SageMath implementations agree on the shared set. The artifact [Ray26c] vendors that implementation at commit `0d79121`, Apache-2.0, with the one set added and no other change, for its example and tests only; the published crates link none of MIKE's code and carry only the set's public constants, digest-checked. Adding the set is ours; its constants are MIKE's code.

**The re-encoding rule (ours)** A supersingular curve over $\mathbb{F}_{p^2}$ with $p \equiv 3 \pmod 4$ has six Montgomery coefficients, one per 2-torsion point sent to $(0, 0)$ and sign. ECLIPSE publishes the largest of the six under the little-endian comparison of re $\|$ im, the SQIsign reference's NormalizeMontgomery (Section 6); MIKE publishes the smallest of twelve, the six and their conjugates, under the same comparison. So the two canonical coefficients of one curve always differ, and MIKE's may belong to $E_A^\sigma$ rather than $E_A$. The rule: the ECLIPSE public key of a MIKE public key is the curve the MIKE bytes name, as published, in ECLIPSE's canonical model with the hint at $2^{e'+2}$; the decoder accepts MIKE's canonical coefficient only, as MIKE's own does. It is arithmetic on a public value, one basis search and six coefficients, which any verifier can do. The reverse encoding exists for every ECLIPSE key and is an encoding only: a key from ECLIPSE's key generation is not a MIKE key (below).

**The derivation (ours)** The ECLIPSE secret key of a MIKE key pair is a function of the MIKE secret $s$ alone. The walk's codomain is recomputed from $s$ on MIKE's model; MIKE's rule says whether the published curve is the codomain or its conjugate; the target is the published curve in ECLIPSE's canonical model. MIKE's $E_0$ is $y^2 = x^3 + x$ under $x \mapsto \lambda(x - r)$ with $2r^2 = 1$ and $\lambda^2 = -2$, determined up to the endomorphism $i$, which does not change a walk's codomain; the kernel's coordinates on the precomputed basis of $E_0[2^e]$ are read by two Tate-pairing discrete logarithms. The kernel's left $O_0$-ideal of norm $2^e$ carries a factor $1 + i$ of degree 2 that leaves class and codomain unchanged; it is dropped and the ideal kept has norm $2^{e-1}$. For a conjugate, the kernel's image under Frobenius is used. From there the derivation is the hop-to-ideal step of RandSK (Section 9) for one hop of length $e$ and no secret ideal: a prime-norm equivalent, as key generation reduces $D_{\text{mix}}$, IdealToIsogeny onto the target, then the hint and $M_{\text{sk}}$ at $2^{e'+2}$, with all randomness from SHAKE256("ECLIPSE-mike-sk-v1" $\|$ $s$), so two derivations give the same bytes. The derived key's public key is the re-encoding of the MIKE public key.

**Cost** Table 9 is a separate session of 2026-10-06 at the shared prime, every row sampled 1000 times round-robin; its ECLIPSE rows (40.7 key generation, 43.8 signing, 24.1 verification cold and 18.8 prepared) are that session's control and sit within the session-to-session spread of Table 5. MIKE costs 4.4 for key generation and 38.1 for the shared secret. The derivation costs 44.2 and 51.5 megacycles for the two keys, 1.1 to 1.3 key generations: its translation is key generation's on an ideal of the same norm, and the discrete logarithms and the $2^{228}$-walk add a few megacycles. Re-encoding and reverse encoding are under a megacycle each. Once derived, the key is an ordinary ECLIPSE key.

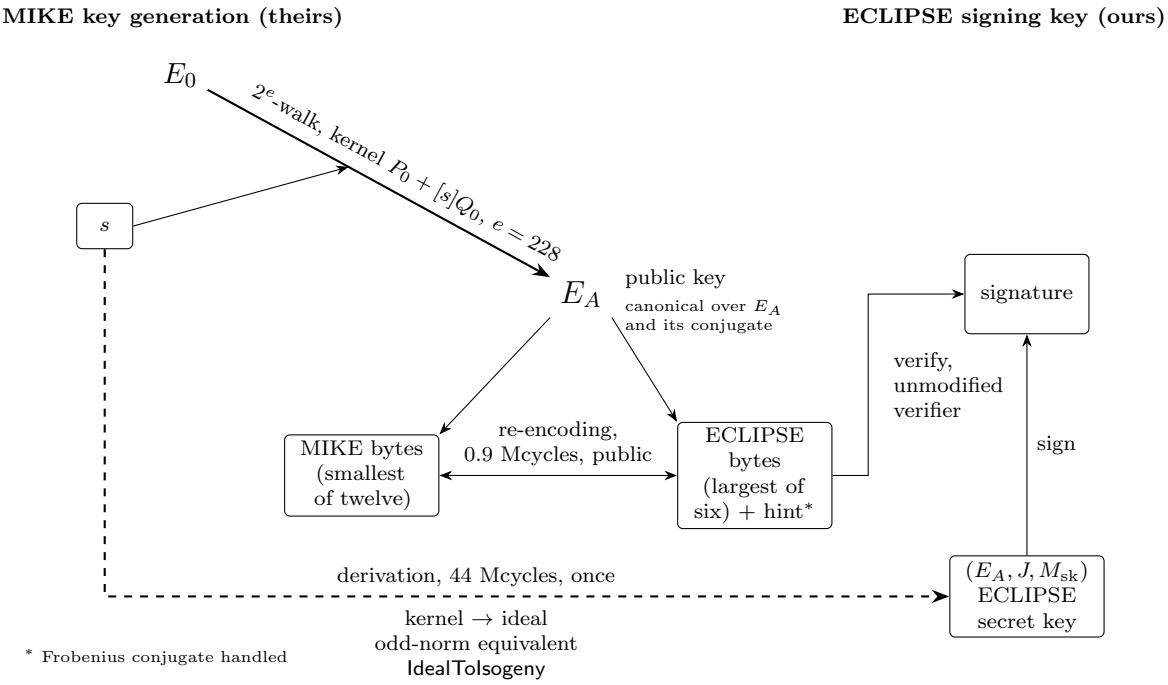



Figure 9: One curve in both roles at the shared prime. MIKE's key generation (left) walks from $E_0$ to $E_A$; the same curve re-encoded (middle) is an ECLIPSE public key, and the ECLIPSE secret key (right) is derived once from the MIKE secret. Figure 10 has the curve's other uses.




Figure 10: Other uses of the same curve: key agreement still runs on $E_A$, and derived children sign only, hardened children do both.

Table 9: One key, both roles at the shared level I prime, megacycles, one session (means of 1000 interleaved runs; medians in parentheses where the tail matters).

| operation | megacycles |
|---|---|
| ECLIPSE key generation | 40.7 (38.9) |
| ECLIPSE signing | 43.8 (41.0) |
| ECLIPSE verification, cold / prepared key | 24.1 / 18.8 |
| derivation of the ECLIPSE secret key from the MIKE secret, two keys | 44.2 (43.7) and 51.5 (51.0) |
| re-encoding of a MIKE public key | 0.93 |
| reverse encoding of an ECLIPSE public key | 0.52 |
| MIKE key generation (Rust, seeded DRBG included) | 4.40 |
| MIKE shared secret (Rust) | 38.1 |

**Direction** Keys start on the MIKE side. A MIKE key is one of at most $2^{e-3}$ curves at the end of a $2^e$-walk from $E_0$; an ECLIPSE key is $E_0/I_{\mathrm{sk}}$ for an ideal of prime norm near $\sqrt{p}$, one of about $p/12$ classes. At the shared prime that is $2^{225}$ among $2^{322}$: a key from ECLIPSE's key generation is a MIKE key with probability about $2^{-97}$, and finding its $s$ would be MIKE's key-recovery problem. The reverse encoding writes MIKE's bytes for such a key and the other party's computation runs on them, but the holder has no MIKE secret for it.

**Hierarchical derivation** RandPK (Section 9) walks two $2^{f-2}$-hops from the key, so the child's kernel is not of MIKE's form: a derived child signs but does not do key agreement, which is all the BIP352-style output key needs, since the shared secret is computed with the recipient's static key; no public derivation stays inside the set of MIKE keys. Hardened derivation does: $s_i = \mathrm{KDF}(\mathrm{seed}, i)$ is a MIKE secret, its public key a MIKE key, and its ECLIPSE key is derived from $s_i$ as above.

**MIKE's security model (theirs)** MIKE is proved an actively secure key exchange in the quantum random-oracle model under a gap Diffie–Hellman assumption, in the CKS-light model, polynomially equivalent to CKS [BDD+26, Section 5.1, Theorem 5.1 and Corollary 5.1]. That adversary registers keys of its own choice beside the honest ones and asks for shared keys between a corrupt and an honest party [BDD+26, Appendix I.1], the static-static setting a long-lived identity key in both roles requires. In their algebraic isogeny model in higher dimension, rigorous MIKE's Diffie–Hellman problem is equivalent to the endomorphism-ring problem [BDD+26, Section 5.4, Corollary 5.2]; fast MIKE has the same reduction under the assumption that its public keys are indistinguishable from uniformly random supersingular curves [BDD+26, Problem 5.3 and Corollary 5.3].

**The inherited assumption** A MIKE-derived ECLIPSE key is one of at most $2^{225}$ curves among $2^{322}$, so using it as an ECLIPSE key rests on fast MIKE's key-indistinguishability assumption [BDD+26, Problem 5.3], of which the project page says that "even if this assumption is broken, it would not give a direct attack on MIKE" [MIK26a, note 1]. Rigorous MIKE takes $e = f - 2$ and does without it; at the shared prime that is $e = 322$, listed in [BDD+26, Appendix J.4] and not built here. Whether Theorem 3 needs anything of the key beyond that is not examined.

**Joint security** Whether one curve may serve as both a MIKE key and an ECLIPSE key is a question about an adversary who sees prime-degree isogenies out of $E_A$ (the signatures) and MIKE's shared-secret computations on $E_A$. Neither scheme's argument is stated for a key used in another scheme, and nothing here bears on it: this section gives the mechanism and its cost, not the security of the combination.

## 11 Comparison

**Sizes at matched states** A size comparison means something only at the same compression and message-binding state, so Table 10 gives both schemes in all four combinations: "compressed" is one coefficient recovered from Weil pairings, "BUFF" a $\lambda$-bit tag in the signature. The ECLIPSE rows are implemented; SQIsign's compressed rows are the measured format of sqisign-rs 0.6 [Ray26d], three entries and one byte of residual bits, and its BUFF rows add $\lambda/8$ bytes for the tag of [AAA+26, Section 8.4.1]. At every matched state ECLIPSE is within 15 bytes of SQIsign.

Table 10: Signature bytes of ECLIPSE and of SQIsign v3 at matched compression and BUFF states.

| state | scheme | NIST I | NIST III | NIST V |
|---|---|---|---|---|
| uncompressed, no BUFF | ECLIPSE (v1: 4 coefficients, no tag) | 215 | 309 | 397 |
| | SQIsign v3 (as specified) | 200 | 306 | 406 |
| compressed, no BUFF | ECLIPSE (3 coefficients, no tag) | 190 | 276 | 356 |
| | SQIsign v3, measured[a] | 176 | 269 | 353 |
| uncompressed, BUFF | ECLIPSE (4 coefficients, $\lambda$-bit tag) | 231 | 333 | 429 |
| | SQIsign v3 + $\lambda$-bit tag, computed[b] | 216 | 330 | 438 |
| compressed, BUFF | ECLIPSE v3, as shipped | 206 | 300 | 388 |
| | SQIsign v3, measured + $\lambda$-bit tag[a] | 192 | 293 | 385 |

[a] The compressed SQIsign format of sqisign-rs 0.6 [Ray26d]: three matrix entries and one byte carrying four residual bits, one per entry, since in SQIsign the pairing relation fixes the determinant only two bits below the matrix modulus (Section 6); the ECLIPSE format has no residual bits.

[b] The round-3 layout (spec Table 11 [AAA+26]) plus the $\lambda/8$ bytes of its Section 8.4.1 tag.

**The dimension-4 route** The salted paper considers a dimension-4 variant in which the prime degree $q$ is embedded natively and $2^a - q$ must be a sum of two squares, and sets it aside at an estimated 7 times the verification cost and a hash said to cost about the same as hashing into $2^{2e-1} - s^2$ [BBC+26a, Remark 1 and Section 5.2]; a verifier recovers the two squares from $q$ by Cornacchia's algorithm. Both figures are now measured (Table 11): the hash needs $q$ prime and $2^a - q$ prime and $\equiv 1 \pmod 4$, 13 291/24 697/39 014 candidates per hit against Hsquares's 72/95/119 and 7.3/8.1/6.3 times the cycles, and the dimension-4 verifier of one code base runs about ten times its dimension-2 verifier at the round-3 level I prime, 120.3 against 12.2 megacycles [Ray26b, Table 3]. Dimension 4 removes the auxiliary curve, which a PRISM signature does not have in any dimension, so the variant would be the same size within a few bytes at a sparser hash and a slower verifier. It is not built.

## 12 Conclusion

ECLIPSE is the prime-degree construction of Section 5 of [BBC+26a], with its salt, built and measured for the first time, on the round-3 SQIsign primes: a signer whose challenge ideal has prime norm and is found without rejection; a verifier with one $(2,2)$-chain, a constant change of theta structure and a pull-back of three points; parameters with $e' = a + 1$, an exact count of the hash target and Miller–Rabin rounds for the prime degree; a canonical wire format with a strict decoder and a message-binding tag; randomisable keys with the key in the hash; and a MIKE key pair at the level I prime made an ECLIPSE key pair. Signatures are 206 / 300 / 388 bytes, within 15 bytes of SQIsign at every matched state; signing is 2.6 / 2.4 / 2.8 times faster than the reference's assembly build and verification costs 2.0 / 1.7 / 1.6 times as much, 1.6 / 1.4 / 1.3 with a prepared key.

**Future work** The proof: Theorem 3 is in the random oracle model with the membership test idealised, and its extension to derived keys is sketched in Section 9, not written; the quantum random oracle model is open; Step 2 of Lemma 1 uses a standard fact on level-2 theta null points as stated; completeness of the verifier on honest signatures with the constant $\omega$ is not proved, only derived (Appendix A) and tested. The joint security of one curve as a MIKE key and an

Table 11: The dimension-4 route, measured: hash density and cost against Hsquares, and the verification multiple.

| | NIST I | NIST III | NIST V |
|---|---|---|---|
| Hsquares candidates per member (measured, 3000 signatures) | 72 | 95 | 119 |
| dimension-4 hash candidates per hit (measured) | 13 291 | 24 697 | 39 014 |
| dimension-4 hash, Hardy–Littlewood prediction | 13 414 | 23 848 | 37 262 |
| ratio of candidates | 185x | 260x | 328x |
| Hsquares cost per signature, median Mcycles | 12.9 | 29.9 | 71.4 |
| dimension-4 hash cost per signature, median Mcycles | 94.7 | 240.8 | 447.6 |
| ratio of cost | 7.3x | 8.1x | 6.3x |
| verification, dimension 4 against dimension 2: estimate of [BBC$^+$26a, Remark 1] | roughly 7× | | |
| the two SQIsign verifiers of one code base at the round-3 level I prime, measured [Ray26b, Table 3] | 9.9x (120.3 / 12.2 Mcycles) | | |

ECLIPSE key (Section 10) is open. MIKE's module action is a monoid action without inverses, by design [BDD$^+$26, Section 1.1], so a threshold variant of the key-agreement side by partial actions is excluded; distributed generation of a single key for both roles is the open question there. Appendix B shows ECLIPSE's signing cost on a prime set by the norm-equation search of the ideal-to-isogeny translation, growing with the cofactor; a translation that does not search at large cofactors is future work. Then the tests the implementation still owes: levels III and V against the Sage model and a constant-time signer. The constructions here are young, ours included; the numbers are the sessions of 2026-10-06 and 2026-10-07 (Sections 8 and 10) on the code of that week, and the density measurements of September 2026.

**Thanks** To the PRISM authors, Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren and Benjamin Wesolowski, for the construction and the sketch this paper instantiates; to the SQIsign team for the reference code the substrate began from; to the MIKE authors for their code and the parameter definition Section 10 runs on; and to Damien Robert for confirming on the pqc-forum list that the shared prime is a MIKE prime.

## A The constant $\omega$

Write $\Phi_1 = \Psi_{e'} \circ \Phi_2 \circ \iota$ as in [BBC$^+$26a, Section 5], $\Psi_{e'}$ the last step and $\iota$ the sign on the second factor; let $A_1$, $A_2$ be the codomains of $\Phi_1$, $\Phi_2$ as the theta-model chains compute them, and $\eta = \omega \circ \Psi_{e'} \colon A_2 \to A_2$. From $\hat{F}F = [2^{2e'-1}]$ and $F\iota = \iota\hat{F}$, $\eta^2 = [2]$ and $\ker \eta$ is the kernel $K$ of the last step: $\eta$ is a square root of two on $A_2$ whose kernel and image on $A_2[2]$ are both $K$. A level-2 theta structure is a symplectic splitting of the 2-torsion, and a step replaces it by the dual splitting

on the quotient [DMPR24], so transporting $A_1$'s structure to $A_2$ by $\omega$ gives the dual of $A_2$'s own (the Hadamard transform) with the two index bits exchanged by $\eta$ (the swap of the middle coordinates). $\eta$ is computable on $A_2[8]$ from the signature alone, $\eta(\Phi_2(z)) = \Phi_2(\iota F(w))$ for $[2^{e'-1}]w = z$, which pins $\omega$ as the unique projective $4 \times 4$ matrix through ten point correspondences; computed from point correspondences at two toy primes and the level I prime, it is the Hadamard transform followed by the swap in each. What the verifier's check of $\omega$ proves, that the two codomains are one polarised surface and the composite of the chain, $\omega$ and the pull-back is an endomorphism of $E_{\mathrm{vk}} \times E_{\mathrm{sig}}$, is Lemma 1; this appendix is the derivation of the constant.

# B ECLIPSE on MIKE's published primes

Before the shared prime, the other configuration was built and measured: ECLIPSE with its level I scheme parameters on MIKE's two published level I primes, $633 \cdot 2^{308} - 1$ (318 bits, fast) and $117 \cdot 2^{374} - 1$ (381 bits, rigorous), MIKE untouched. Public keys are 81 and 97 bytes, signatures 204 and 220. The derivation and the re-encoding of Section 10 are built for both primes. The rigorous prime is the second shipped set (Section 5); the fast prime is not shipped, and its measurements are kept here.

**Field arithmetic** The x86-64 schedule keeps every stored value below $2^{64n}$ for $n$ saturated limbs and applies no correction after a product, which lies below $ab/R + p$ with $R = 2^{64n}$; that is sound only when $2^{64n} - p$ exceeds the largest $ab/R$ the lazily reduced inputs produce. The SQIsign primes have room ($3 \cdot 2^{324} - 1$ uses 326 of 384 bits). MIKE's cofactors sit just under a power of two: $633 \cdot 2^{308} - 1$ leaves $391 \cdot 2^{308}$ below $2^{320}$ at five limbs and $117 \cdot 2^{374} - 1$ leaves $11 \cdot 2^{374}$ below $2^{384}$ at six, too little in both cases. The generator was generalised to a Montgomery fold one limb below the top, the carry running on to a top limb that stays zero, giving six and seven limbs with 66 and 67 spare bits under the same run-time dispatch.

**Cost** Table 12 is one session of 1000 interleaved runs per row, with the C reference's own benchmark at its level I prime for scale, and the shared-prime rows of Table 9 as the control. Verification sits where the field arithmetic puts it, 25.0 and 32.7 megacycles cold against 24.1 at the shared prime. Key generation and signing do not: 463 and 170 megacycles for key generation and 468 and 168 for signing against 40.7 and 43.8, with long tails (medians 236 and 117, maxima 4670 and 1868).

**Where the cost is** Key generation, signing and the derivation all translate an ideal of $O_0$ into an isogeny, and the translation starts with the norm equation of the reference's IdealToIsogeny [AAA+26]: two equivalent ideals whose norms add up to $n(I)\,2^f$, found by a search over a scalar for a representable norm. With $p = c \cdot 2^f - 1$ the equation sits at $2^f = (p+1)/c$, so a larger cofactor leaves fewer solutions in range and the search tries more values. Over 10 000 key-generation-shaped ideals per prime no translation failed, and the search tried 268 values per ideal at $c = 3$, 7 712 at $c = 117$ and 35 275 at $c = 633$. In the session it took 8 112 trials and 69 % of signing at $c = 117$, 34 310 trials and 92 % at $c = 633$; the derivation of Section 10 costs 255 and 122 megacycles for the two keys at $c = 117$ and 385 and 151 at $c = 633$, against 44.2 and 51.5 at the shared prime (Table 9), the translation 85 % and 94 % of it. The attribution is the search alone: verification does not translate, and its rows sit between the two level I kernel paths of Section 8.

**Why the shared prime** MIKE's requirement on the prime is the power of two in $p + 1$ and its arithmetic is generic in the modulus; ECLIPSE's parameters and numbers are tied to the SQIsign

Table 12: ECLIPSE on MIKE's published level I primes, megacycles, one session: means (medians in parentheses) of 1000 interleaved runs on the x86-64 assembly path; the SQIsign reference's own benchmark at its level I prime for scale; MIKE's Rust implementation at its own sets. The shared-prime rows are the session of Table 9 (2026-10-06), the control the text compares to.

| scheme | prime | key generation | signing | verification, cold / prepared |
|---|---|---|---|---|
| ECLIPSE, $e' = 193$ | $117 \cdot 2^{374} - 1$ | 170.0 (117.5) | 168.2 (114.6) | 32.7 / 24.7 |
| ECLIPSE, $e' = 193$ | $633 \cdot 2^{308} - 1$ | 463.1 (236.4) | 467.8 (256.6) | 25.0 / 19.3 |
| ECLIPSE, Table 9 | $3 \cdot 2^{324} - 1$ | 40.7 (38.9) | 43.8 (41.0) | 24.1 / 18.8 |
| ECLIPSE, portable kernels | $117 \cdot 2^{374} - 1$ | | | 82.6 / |
| ECLIPSE, portable kernels | $633 \cdot 2^{308} - 1$ | | | 65.3 / |
| SQIsign round 3, C `broadwell` | $3 \cdot 2^{324} - 1$ | 36.2 (34.9) | 107.1 (102.7) | 11.6 / |
| SQIsign round 3, C `ref` | $3 \cdot 2^{324} - 1$ | 51.3 (50.0) | 148.2 (143.8) | 19.4 / |

| | | derivation, two keys | re-encoding | MIKE key generation / shared secret |
|---|---|---|---|---|
| MIKE rigorous, $e = 372$ | $117 \cdot 2^{374} - 1$ | 255.0 / 122.1 | 1.52 | 7.70 / 63.9 |
| MIKE fast, $e = 228$ | $633 \cdot 2^{308} - 1$ | 385.1 / 150.7 | 1.08 | 2.94 / 23.2 |
| MIKE, $e = 228$, Table 9 | $3 \cdot 2^{324} - 1$ | 44.2 / 51.5 | 0.93 | 4.40 / 38.1 |

primes and its signing cost to the cofactor. Moving MIKE to the SQIsign level I prime keeps every published ECLIPSE number and analysis, costs MIKE a field of 326 bits in place of 318, and removes the cofactor's cost: signing returns to 43.8 megacycles, the derivation to 44.2 and 51.5 (Table 9). That is the default; the rigorous prime is kept as a second set for keys that already exist at it.

## References

[AAA+25] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign – algorithm specifications and supporting documentation, version 2.0. NIST PQC Digital Signatures Round 2 Submission, 2025. `https://sqisign.org/spec/sqisign-20250205.pdf`.

[AAA+26] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Giacomo Borin, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Thomas Espitau, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, Riccardo Invernizzi, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Marzio Mula, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, Frederik Vercauteren, Alexandre Wallet, Benjamin Wesolowski, and Wessel van Woerden. SQIsign – algorithm specifications and supporting documentation, version 3.0. NIST PQC Digital Signatures Round 3 Submission, 2026. `https://sqisign.org/spec/sqisign-20260901.pdf`.

[ADR02] Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 83–107. Springer, 2002.

[BBC+25] Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. PRISM: Simple and compact identification and signatures from large prime degree isogenies. In *Public-Key Cryptography – PKC 2025*, volume 15676 of *LNCS*, pages 300–332. Springer, 2025. ePrint 2025/135.

[BBC+26a] Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. PRISM with a pinch of salt: Simple, efficient, and strongly unforgeable signatures from isogenies. *Journal of Cryptology*, 39(4):30, 2026. Article 30, 45 pages, open access; numbering follows the journal version. Preprint: Cryptology ePrint Archive, Paper 2026/443, `https://eprint.iacr.org/2026/443`.

[BBC+26b] Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. PRISM_v2: A C implementation of PRISM v2 (the salted variant).

`https://github.com/mariascrs/PRISM_v2`, 2026. The PRISM team's implementation, per its NOTICE; commit `87fad7da`; salt-PRISM on the round-2 SQIsign primes.

[BBC+26c] Andrea Basso, Giacomo Borin, Maria Corte-Real Santos, Pierrick Dartois, Riccardo Invernizzi, Luciano Maino, Robi Pedersen, and Michel Seck. Isogeny-based signatures with randomizable keys. Cryptology ePrint Archive, Paper 2026/1169, 2026. `https://eprint.iacr.org/2026/1169`.

[BCC+23] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In *Advances in Cryptology – EUROCRYPT 2023*, volume 14005 of *LNCS*, pages 405–437. Springer, 2023.

[BDD+26] Andrea Basso, Pierrick Dartois, Max Duparc, Jonathan Komada Eriksen, Sabrina Kunzweiler, Michael Meyer, Giacomo Pope, Krijn Reijnders, Damien Robert, Ryan Rueger, and Sina Schaeffler. MIKE: a fast and compact post-quantum NIKE. `https://mike.isogeni.es/mike.pdf`, 2026. October 2026, 139 pages.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Advances in Cryptology – ASIACRYPT 2019*, volume 11921 of *LNCS*, pages 227–247. Springer, 2019.

[BL04] Christina Birkenhake and Herbert Lange. *Complex Abelian Varieties*, volume 302 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2nd edition, 2004. Chapter 10, Abelian Surfaces: Section 10.2, The $16_6$ Configuration of the Kummer Surface, and Section 10.3, An Equation for the Kummer Surface.

[CFG+26] Wouter Castryck, Luca De Feo, Steven D. Galbraith, Péter Kutas, Krijn Reijnders, and Benjamin Wesolowski. The isogeny problems. Cryptology ePrint Archive, Paper 2026/1431, 2026. `https://eprint.iacr.org/2026/1431`.

[CGH+25] Sofía Celi, Scott Griffy, Lucjan Hanzlik, Octavio Perez-Kempner, and Daniel Slamanig. SoK: Signatures with randomizable keys. In *Financial Cryptography and Data Security – FC 2024*, LNCS, pages 160–187. Springer, 2025. ePrint 2023/1524.

[CGMT26] Jolijn Cottaar, Steven D. Galbraith, Luciano Maino, and Monika Trimoska. An analysis of a weakened version of PRISM. *IACR Communications in Cryptology*, 3(2), 2026. ePrint 2026/906, `https://eprint.iacr.org/2026/906`.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In *Advances in Cryptology – ASIACRYPT 2018*, volume 11274 of *LNCS*, pages 395–427. Springer, 2018.

[Coh98] Henri Cohen. High precision computation of Hardy–Littlewood constants. Draft of a preprint, Université Bordeaux I, 1998. `https://www.math.u-bordeaux.fr/~hecohen/hardylw.dvi`.

[Dar24] Pierrick Dartois. Fast computation of 2-isogenies in dimension 4 and cryptographic applications. Cryptology ePrint Archive, Paper 2024/1180, 2024. `https://eprint.iacr.org/2024/1180`; Theorem 12 is the change of theta coordinates.

[DLP93] Ivan Damgård, Peter Landrock, and Carl Pomerance. Average case error estimates for the strong probable prime test. *Math. Comp.*, 61(203):177–194, 1993.

[DMPR24] Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert. An algorithmic approach to $(2,2)$-isogenies in the theta model and applications to isogeny-based cryptography. In *Advances in Cryptology – ASIACRYPT 2024*, LNCS, pages 304–338. Springer, 2024. ePrint 2023/1747; code at `https://github.com/ThetaIsogenies/two-isogenies`.

[FLMZ26] Tako Boris Fouotsa, Trey Li, Bernardo Magri, and Shancheng Zhang. iX3DH: Post-quantum subversion-resilient X3DH key-exchange from isogenies. Cryptology ePrint Archive, Paper 2026/2296, 2026. Received 2026-10-01, revised 2026-10-05; `https://eprint.iacr.org/2026/2296`.

[Fou25] Tako Boris Fouotsa. WaterSQI and PRISMO: Quaternion signatures for supersingular isogeny group actions. Cryptology ePrint Archive, Paper 2025/1737, 2025. Revision of 2026-09-05; `https://eprint.iacr.org/2025/1737`.

[GD94] Maria R. Gonzalez-Dorrego. $(16,6)$ configurations and geometry of Kummer surfaces in $\mathbb{P}^3$. *Mem. Amer. Math. Soc.*, 107(512):vi+101, 1994.

[HL23] G. H. Hardy and J. E. Littlewood. Some problems of 'Partitio Numerorum'; III: On the expression of a number as a sum of primes. *Acta Math.*, 44:1–70, 1923.

[IC26] Riccardo Invernizzi and Maria Corte-Real Santos. isogeny_swrk: Isogeny signatures with randomizable keys in Sage. `https://github.com/giacomoborin/isogeny_swrk`, 2026. Repository of Giacomo Borin; the SageMath implementation of SQIsign-RK, PRISM-RK and DeuringVUF-RK; commit `37f504a`.

[IVB26] Riccardo Invernizzi, Frederik Vercauteren, and Giacomo Borin. sqisign_prism_v2: SQIsign / PRISM v2 in Sage. `https://github.com/KULeuven-COSIC/sqisign_prism_v2`, 2026. Repository of KU Leuven COSIC; a SageMath proof of concept of SQIsign v2.0 and salt-PRISM; commit `297ddf1f`.

[Kan97] Ernst Kani. The number of curves of genus two with elliptic differentials. *J. Reine Angew. Math.*, 485:93–122, 1997.

[KKL26] Hyeonhak Kim, Won Kim, and Changmin Lee. SPRITZ: A short PRISM-based threshold signature via zero-knowledge proof. Cryptology ePrint Archive, Paper 2026/945, 2026. `https://eprint.iacr.org/2026/945`.

[MIK26a] MIKE team. MIKE: a fast and compact post-quantum NIKE, project page. `https://mike.isogeni.es/`, 2026. Note 1 states the fast variant's additional assumption.

[MIK26b] MIKE team. mike_py: MIKE in SageMath. `https://github.com/tensor-MIKE/mike_py`, 2026. Commit `4c9bbf3`, 2026-10-04; Apache-2.0.

[Mum66] David Mumford. On the equations defining abelian varieties. I. *Invent. Math.*, 1:287–354, 1966.

[NO26] Kohei Nakagawa and Hiroshi Onuki. Attacks on PRISM-id via torsion over small extension fields. In Shi Bai and Edoardo Persichetti, editors, *Public-Key Cryptography – PKC 2026, Part I*, LNCS, pages 480–509. Springer, 2026. ePrint 2025/1602, whose numbering the locators follow; `https://eprint.iacr.org/2025/1602`.

[Piz90] Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bull. Amer. Math. Soc. (N.S.)*, 23(1):127–137, 1990.

[Piz98] Arnold K. Pizer. Ramanujan graphs. In *Computational Perspectives on Number Theory (Chicago, IL, 1995)*, volume 7 of *AMS/IP Stud. Adv. Math.*, pages 159–178. Amer. Math. Soc., Providence, RI, 1998.

[Pop26] Giacomo Pope. A string theorist's guide to implementing MIKE. Blog post, `https://giacomopope.com/mike/`, 2026.

[Por13] Thomas Pornin. Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA). RFC 6979, RFC Editor, 2013.

[Ray26a] Dustin Ray. Auxiliary isogeny freedom in SQIsign's two-dimensional representation. Cryptology ePrint Archive, Paper 2026/1305, 2026. `https://eprint.iacr.org/2026/1305`.

[Ray26b] Dustin Ray. Dimension-4 SQIsign at round-3 parameters: Sizes and costs of the compact format. Cryptology ePrint Archive, Paper 2026/2221, 2026. `https://eprint.iacr.org/2026/2221`.

[Ray26c] Dustin Ray. eclipse-rs: ECLIPSE and salt-PRISM on the SQIsign round-3 substrate. Software, `https://github.com/anchorlabsinc/eclipse-rs`, 2026. The artifact of this paper; commit `4b15890`.

[Ray26d] Dustin Ray. sqisign-rs: SQIsign round 3 in pure Rust, with the compressed and the dimension-4 compact formats. `https://github.com/anchorageoss/sqisign-rs`, 2026. Release 0.6.

[Rob24] Damien Robert. The module action for isogeny based cryptography. Cryptology ePrint Archive, Paper 2024/1556, 2024. `https://eprint.iacr.org/2024/1556`.

[Rob26] Damien Robert. Re: [pqc-forum] MIKE (Module Isogeny Key Exchange): a fast and compact post-quantum NIKE. Message to the pqc-forum list (Google Groups), in the thread of the same subject opened by the author on 2026-10-06, 2026. Reply of 2026-10-06.

[The26] The SQIsign team. the-sqisign: reference implementation of SQIsign. `https://github.com/SQISign/the-sqisign`, 2026. tag `nist-v3`, commit `6d017708`.

[Wes26] Benjamin Wesolowski. The supersingular isogeny problem in time and memory $p^{1/3+o(1)}$. Cryptology ePrint Archive, Paper 2026/1486, 2026. `https://eprint.iacr.org/2026/1486`.