

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. A formal strong-unforgeability (SUF-CMA) argument is left to future work.

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 [C⁺26] and SageMath [I⁺26] 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 ECLIPSE¹. It contributes:

1. the first implementation, on the SQIsign round-3 primes, in Rust [Ray26c], and a SageMath model that accepts its signatures and whose signatures it accepts (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 (Section 4, Appendix A);
3. 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);
4. a canonical wire format with a strict decoder, one coefficient recovered from two Weil pairings, and a message-binding tag (Section 6);
5. one benchmark session against the SQIsign reference’s assembly build and our own port of it (Section 8);
6. randomisable keys with the key kept in the hash (Section 9);
7. 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 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 says what was checked and what was not; 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 and Remark 11 on adding the salt. Its parameters, benchmarks and C code are for salt-PRISM on the round-2 SQIsign primes. Three code bases exist, the authors’ C implementation [C⁺26], a SageMath proof of concept of SQIsign version 2.0 and PRISM [I⁺26], 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⁺26b]. Kim, Kim and

¹Encoding-Canonical Large-prime Isogeny Signature Extension: Section 5 of [BBC⁺26a] with the salt of its Section 6 as Remark 11 suggests. 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 <code>rdtsc</code> ; 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 $\mathbb{F}_{p^2}$.

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. Our port reproduces the 300 round-3 known-answer vectors byte for byte, 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 To our knowledge no implementation of the Section 5 construction existed before this one. The three PRISM code bases above [C⁺26, I⁺26, 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_{vk} and a one-byte hint for the deterministic basis ($P_{\text{vk}}, Q_{\text{vk}}$) of $E_{\text{vk}}[2^t]$; the secret key is $(E_{\text{vk}}, I_{\text{sk}}, M_{\text{sk}})$, the ideal of $E_0 \rightarrow E_{\text{vk}}$ 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 as Remark 11 suggests. 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 $(E_{\text{vk}}, \text{msg}, \text{salt})$. Ours: the primes and the exponents $e' = a + 1$, $t = e' + 2$ (Section 5), the tag, the single chain with the constant ω and the pull-back in place of a second chain, the scalar check on three points of $E_{\text{vk}}[2^t]$, the deterministic salt, and the wire format with its decoder (Section 6). Algorithms 1 and 2 mark each block with its source; Figure 1 is the data flow with each phase’s share of the level I cost.

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

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

Algorithm 2 Verify(vk, msg, σ)**Input:** $vk = (A(E_{vk}), \text{hint})$; a message msg ; a byte string σ .**Output:** accept or reject.

- 1: $(E_{sig}, P_{sig}, Q_{sig}, salt, tag) \leftarrow \text{Decode}(vk, \sigma)$; **reject** on $\perp \triangleright$ strict; the fourth coefficient from two Weil pairings
- 2: $(s, tag') \leftarrow$ the hash of lines 4 and 5 of Algorithm 1 on $(E_{vk}, msg, salt)$
- 3: **reject** unless $tag' = tag$ (compared in constant time) and $d = 2^{2e'-1} - s^2$ passes the t rounds
- 4: $(P_{vk}, Q_{vk}) \leftarrow$ the basis of $E_{vk}[2^t]$ from the hint; $u \leftarrow s^{-1} \bmod 2^t$
- 5: Run the $(2, 2)$ -chain of e' steps from $E_{vk} \times E_{sig}$ with kernel $\langle (P_{vk}, -P_{sig}), (Q_{vk}, -Q_{sig}) \rangle$, checking the kernel at every step, carrying $([u]T, 0)$ for $T \in \{P_{vk}, Q_{vk}, P_{vk} - Q_{vk}\}$ and recording the null point θ_j of every intermediate surface; **reject** if a step fails $\triangleright \Phi_1$ of [BBC⁺26a, Section 5]
- 6: **reject** unless $\omega(\theta_{e'}) = \theta_{e'-1} \triangleright$ 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 ω **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 $\pm(T, T_{sig})$ on the two Kummer lines, T_{sig} the matching signature point
- 10: **end for**
- 11: **return** accept

Why it is a proof of the degree Section 5 of [BBC⁺26a] embeds σ in the $(2^{2e'-1}, 2^{2e'-1})$ -isogeny $F = \hat{\Phi}_2 \circ \Phi_1$ of $E_{vk} \times E_{sig}$, with $\Phi_1 = \Psi_{e'} \circ \Phi_2 \circ \iota$, and states that F embeds an isogeny of degree d exactly when its top-left block is $[s]$. It tests this as $\Psi_{e'} \circ \Phi_2([2^{e'-1}]T, 0) = \Phi_2([s]T, 0)$ on a basis of $E_{vk}[N]$ for any N with $N^2 > 2^{2e'} + 1$, taking $N = p + 1$. Algorithm 2 evaluates the same condition with three changes: Φ_1 is the chain of line 5 and Φ_2 its first $e' - 1$ steps, computed with two theta structures that the constant ω identifies; $\hat{\Phi}_2$ is evaluated by pulling the carried points back through

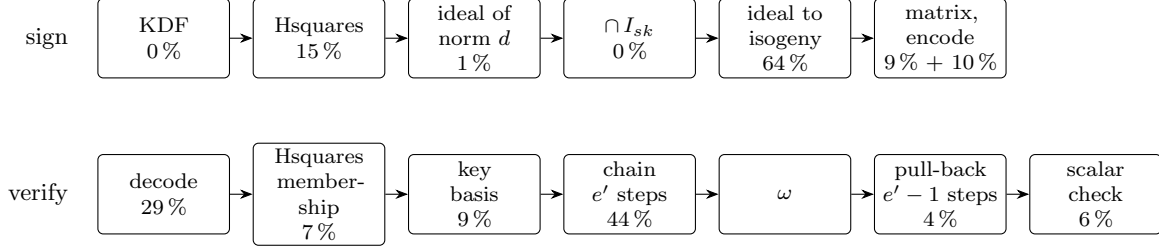


Figure 1: Signing (top) and verification (bottom) of ECLIPSE, one box per phase, with each box’s share of the level I median of the phase profile of 2026-09-25 (signing 44.4 megacycles including encoding, verification 22.5 from bytes). The ω box and the dual of the gluing that ends the pull-back are below one percent each; the pairing recovery inside the decoder is 12% of verification.

the recorded steps, not by a second chain; and $N = 2^t$ meets the condition, so three points of $E_{\text{vk}}[2^t]$, whose basis the hint already gives, replace a basis of $E_{\text{vk}}[p + 1]$.

The pull-back The dual of a $(2, 2)$ -step $f: A \rightarrow B$ 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 $E_{\text{vk}} \times E_{\text{sig}}$. It costs a tenth of the forward chain.

The constant ω In [BBC⁺26a, Section 5] the codomains of Φ_1 and Φ_2 are one surface. Computed as theta-model chains they are that surface with two theta structures, A_1 and A_2 , and $\omega: A_1 \rightarrow A_2$ is the change between them: a constant, the Hadamard transform followed by the swap of the two middle coordinates, the same at every instance (Figure 2; 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 .

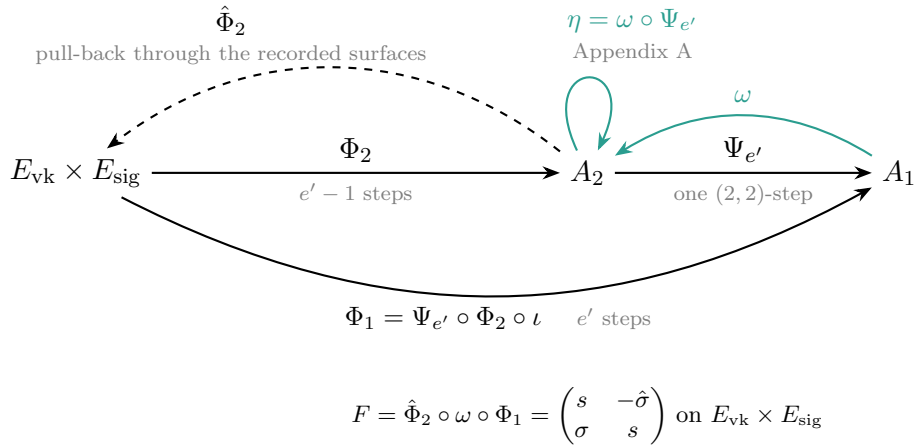


Figure 2: The two isogenies of the criterion, the map between their codomains, and the verifier’s way back. Φ_2 is the first $e' - 1$ steps of the chain of Φ_1 and $\Psi_{e'}$ its last step. $\omega: A_1 \rightarrow A_2$ is the constant change of theta structure that identifies the two codomains; the 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.

The scalar check on $E_{\text{vk}}[2^t]$ $N = 2^t$ 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 , since σ is determined by its images on the $2^{e'}$ -torsion [BBC⁺26a, Section 5]. 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 the strong variant’s reduction requires [BBC⁺26a, Remark 8 and Section 5.1]. 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_{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.

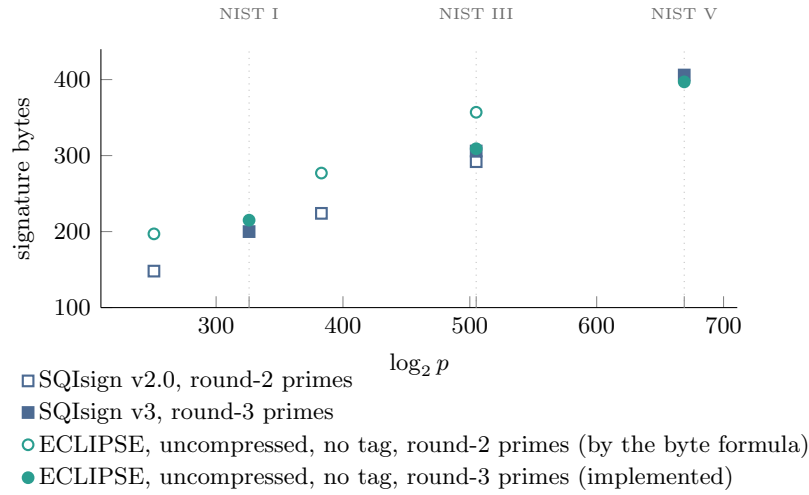


Figure 3: Signature size against the prime, SQIsign and ECLIPSE at the matched state (four coefficients, no tag). SQIsign sizes from the two specifications' Table 1 [AAA⁺25, AAA⁺26]; the ECLIPSE round-2 points are by the same byte formula, since the scheme is not instantiated on the round-2 primes. Level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

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 $C = \prod_{p \text{ odd}} (1 - (2/p)/(p-1)) = 1.8500544 \dots$ 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, 4000 odd s per exponent: 0.0128/0.0083/0.0075 at $e' = 193/257/321$. The Rust signer, 300 signatures per level: 0.0149/0.0112/0.0088. 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 sets the signing cost, about 67/89/113 candidates per signature rather than the estimate's 19/24/29, and the salt bound below.

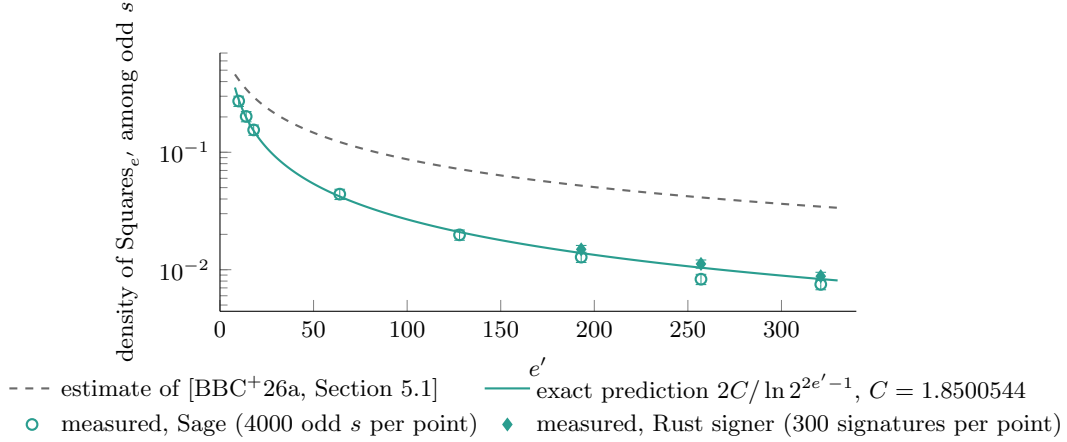


Figure 4: Density of $\text{Squares}_{e'}$ against e' : the Sage and Rust measurements with their statistical error (about 10 % and 8 %), the estimate of [BBC⁺26a, Section 5.1], and the exact Hardy–Littlewood prediction with the constant C of that section’s own form.

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–Landroock–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. It reproduces their Table 4 first; for d it gives $t = 9/19/28$ at the bare thresholds 2^λ . 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 0.09/0.18/0.32 ms per round on the machine of Section 8.

The salt With γ the density over the full digest range, the bound of [BBC⁺26a, Equation 18], $k \geq \lambda + 64 - \log_2 \gamma$, 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 $a \geq 7\lambda/6$. 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 $a \geq 4\lambda/3$ [NO26, Section 5.3] reads $e' \geq 86/129/172$ 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
λ^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 $d = 2^{2e'-1} - s^{2c}$	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.

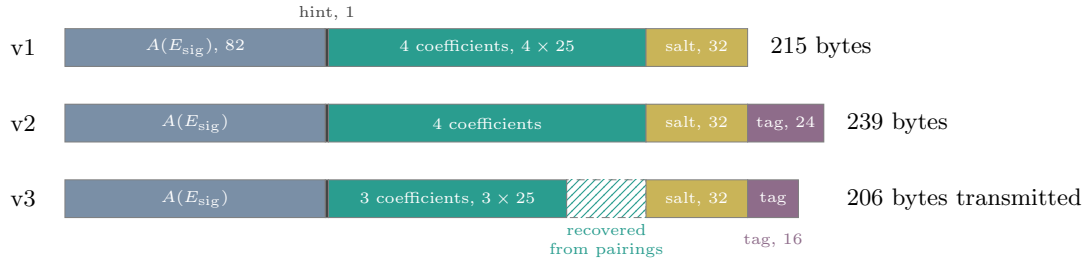


Figure 5: Byte layout of the ECLIPSE signature at level I in the three wire formats, segment widths proportional to bytes. v1: the curve, the hint, four coefficients of $\lceil (e' + 2)/8 \rceil$ bytes and the salt. v2 adds an n -bit tag. v3, the shipped format, transmits three coefficients, recovers the fourth from two Weil pairings (hatched, drawn at its width but not transmitted) and carries a λ -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: the scalar check rejects a point shifted by the 2- or 4-torsion above the kernel, so 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	λ bits of the digest's high half	16	24	32
total	$2 \text{ FP} + 1 + 3[(e' + 2)/8] + k/8 + \lambda/8$	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_{sig} , the basis the coefficients refer to, and the fourth coefficient, which the other three determine. Figure 6 pairs each with its rule. The first non-zero entry of M must be below $2^{e'+1}$, so $-M$ is rejected. $A(E_{\text{sig}})$ must equal the coefficient recomputed from a 4-torsion point of the hint basis, the model the theta splitting produces, and $A = \pm 2$ is rejected. 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.

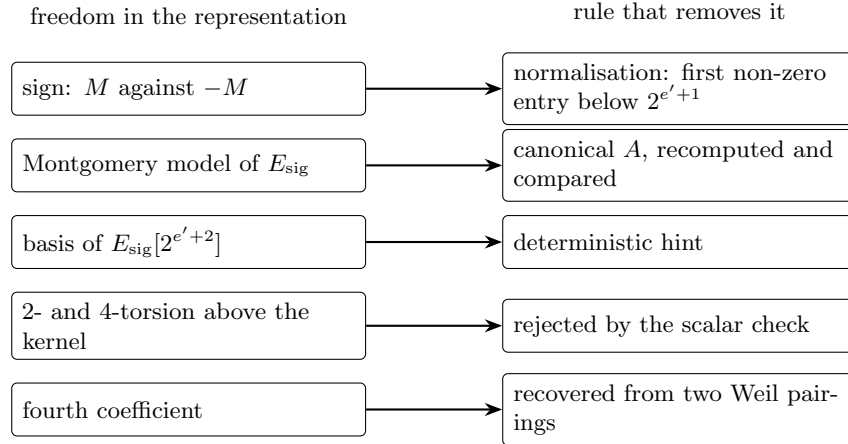


Figure 6: Each freedom in the representation of a signature (left) and the rule that removes it (right): the wire format's normalisation, canonical curve model and deterministic hint, the verifier's scalar check, and the pairing recovery of the fourth coefficient.

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 $e(P_{\text{vk}}, Q_{\text{vk}})$, 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 λ 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 What we checked

Two implementations The Rust implementation and a SageMath model accept each other’s signatures. At the level I prime, three Rust signatures are accepted by the Sage verifier and their wrong- s variants rejected, 6 of 6; at the model’s 17-bit prime with $e' = 8$, four Sage signatures re-expressed on the Rust hint basis are accepted by the Rust verifier and the wrong- s , swapped and wrong-key variants rejected, 16 of 16. Keys and signatures cannot match byte for byte across the two (different key generation and challenge randomness); cross-acceptance is the achievable statement. Our salt-PRISM layer accepts the isogenies of the authors’ C implementation [C⁺26] at every round-2 level given the same basis, 26 of 26 per level.

Toy exhaustive checks At two 17-bit primes the Sage model enumerated, for six honest signatures per prime, every representation of the honest isogeny reachable without the endomorphism ring: the common sign, the 15 non-trivial 2-torsion shifts of the pair, and the automorphism images. Exactly the honest representation is accepted; the sign flip, a wrong s and a wrong key are rejected; a second isogeny of the same degree from a fresh challenge ideal is a distinct valid signature, as it must be, and needs the secret key. What the enumeration cannot cover is the other isogenies of degree N from the public curve, about 2^{27} even at toy size.

Full-size negatives At all three levels, nine tamperings of an honest signature are rejected: the next member of $\text{Squares}_{e'}$, another message, another key, the points swapped, $[3]P_{\text{sig}}$, $P_{\text{sig}} + Q_{\text{sig}}$, the curve replaced, and P_{sig} shifted by the 2-torsion or the 4-torsion point above Q_{sig} . The 2-torsion shift is the one our salt-PRISM verifier accepts; here it fails the scalar check.

The positive control The codomain-reroute attack of [Ray26a, Section 6, Proposition 6.1] was run at the level I prime, where the 3-torsion is rational. On a salt-PRISM signature whose auxiliary degree $2^a - q$ is divisible by 3 it tries every 3-isogeny as the dual of the unknown factor and every 3-isogeny onward, twelve candidates, and exactly three verify, the three replacements of the degree-3 factor. On an ECLIPSE signature the same twelve give nothing: every one fails at the null-point match, because the two chain codomains are not isomorphic surfaces. That is the prime-degree obstruction observed directly.

Mutation and fuzzing Every bit of a level I signature (1720) and public key (664) was flipped: no flip verifies. The same test on salt-PRISM accepts exactly one flip, the 4-torsion bit of a coefficient. Seven libFuzzer harnesses (the two signature decoders, the public-key decoder, both verifiers, the hash to primes and Hsquares) ran 90 seconds each from seeded corpora: no accepted non-canonical encoding, nothing but the seed pair verifies, every verifying input re-encodes identically, and one hang in the Miller–Rabin base sampler on a slice with leading zero words, fixed. Table 5 collects the checks.

Table 5: What was checked, what each check rules out, and the result.

check	rules out	result
two implementations agree	a verifier that accepts its own signer only	Sage accepts Rust 6/6, Rust accepts Sage 16/16 at the level I prime; the isogenies of PRISM_v2 [C ⁺ 26] under our salt-PRISM verifier 26/26, 26/26, 26/26 at the round-2 levels
toy exhaustive checks (Sage, two 17-bit primes)	a second accepted representation of the honest isogeny	0 / 15 2-torsion shifts and 0 / 1 automorphism images accepted; honest 6 / 6 per prime; the 2^{27} isogenies of degree N not enumerated
full-size negatives (Rust, three levels)	acceptance of a tampered signature	9 tamperings per level (wrong s , message, key; swapped, scaled, shifted points; replaced curve), all rejected
positive control (level I, degree-3 reroute)	that the reroute fails for the wrong reason	salt-PRISM: exactly 3 of 12 candidates verify; ECLIPSE: 0 of 12
mutation (level I)	a byte string one flip away that verifies	1720 signature bits and 664 key bits flipped, 0 accepted; salt-PRISM 1 (its 4-torsion bit)
fuzzing (libFuzzer)	an accepted non-canonical encoding; a hang	7 harnesses; every verifying input re-encodes identically; one hang found and fixed
derived keys (Rust, three levels; Sage toy)	a signature that moves to another key; a push-forward forgery	cross-key negatives rejected at every level; toy battery under 2 derived keys 0 / 60 shifts accepted; the push-forward is exact and fails only on the derived key’s digest, salt grind 51 and 27 of $2^{e'-2} = 64$
test suite		269 test functions, property batteries and byte-for-byte comparisons with the C reference

What is not established The strong-unforgeability argument follows Theorem 2 of [BBC⁺26a] with its Problem 5 (given isogenies of degrees $2^{2e'-1} - s_i^2$ for random s_i and a target $\bar{s}$, produce an isogeny of degree $2^{2e'-1} - \bar{s}^2$ not in the list) in place of Problem 3. A forgery is one of three cases. A fresh digest: the forged isogeny solves the instance at the target the reduction guessed among the N_H hash queries. The same digest with a different isogeny: Problem 5’s second clause. The same isogeny with a different encoding: excluded by the scalar check and the canonical decoder, which accept one byte string per isogeny under a key (Sections 4 and 6). The paper sketches this for its unsalted variant [BBC⁺26a, Section 5.1] and remarks that Theorem 2 carries over to a salted version [BBC⁺26a, Remark 11], but writes neither; no written reduction with its loss exists, and

this paper does not provide one. Problem 5 is a conjectured assumption with no reduction from the endomorphism ring problem, as Problem 3 is for salt-PRISM.

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 6 are of the same 1000 signatures. The C reference is the `sqsign` 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. Our port of the `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 our own `SQIsign` is a ratio on identical arithmetic. Verification is from bytes: decode the key, decode the signature, verify.

Cycles Table 6 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 our own `SQIsign`, 1.9 / 1.5 / 1.4 cold and 1.6 / 1.3 / 1.2 prepared. Table 7 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.

Table 6: 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, and how ECLIPSE compares with the C reference (cold / prepared) and with salt-PRISM.

signing, mean (median; maximum)		NIST I		NIST III		NIST V	
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)			
ECLIPSE vs C / vs salt-PRISM (means)		2.6x / 2.3x faster	2.4x / 2.1x faster	2.8x / 2.7x faster			

operation level		SQIsign (C)	SQIsign (Rust)	salt-PRISM	ECLIPSE	ECLIPSE, prep.	vs C	vs salt-PRISM
verify	I	11.1	12.0	17.0	22.3	18.2	2.0x/1.6x slower	1.3x slower
	III	29.4	32.1	37.7	49.7	40.9	1.7x/1.4x slower	1.3x slower
	V	76.0	83.7	88.9	121.0	100.7	1.6x/1.3x slower	1.4x slower

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

Table 7: 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 (λ)	128	192	256

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 6: 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 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 λ 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 8 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 10. 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 *IdealTolsogeny* 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 8: The second parameter set and MIKE’s context, megacycles, one session kind (1000 interleaved runs per row; MIKE’s rows are **criterion** means of 100 samples). The signing tail is the mean, the median and the extremes over the 1000 runs; the trial count is the norm-equation search’s second loop, averaged per call.

	prime	mean	median	min	max	search trials
ECLIPSE key generation	$117 \cdot 2^{374} - 1$	170.0	117.5	50.5	1868	7 634
ECLIPSE signing	$117 \cdot 2^{374} - 1$	168.2	114.6	47.0	1330	8 112
ECLIPSE verification, cold / prepared	$117 \cdot 2^{374} - 1$	32.7 / 24.7	32.4 / 24.5			
ECLIPSE key generation	$3 \cdot 2^{324} - 1$	40.7	38.9	32.1	81.5	268
ECLIPSE signing	$3 \cdot 2^{324} - 1$	43.8	41.0	31.2	104.9	
	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 9 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 9: Milliseconds at level I. Xeon: rdtsc, assembly, 2.8 GHz, the session of Table 6. 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⁺26b, Definition 3] after Celi, Griffy, Hanzlik, Perez-Kempner and Slamanig [CGH⁺25]: $\text{RandPK}(pk, \rho)$ derives a public key from public data, $\text{RandSK}(sk, \rho)$ the matching secret key (our secret key carries its public key), and $\text{VerKey}(pk, sk)$ 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 ρ into two scalars and walks two hops from E_{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⁺26b, Table 4].

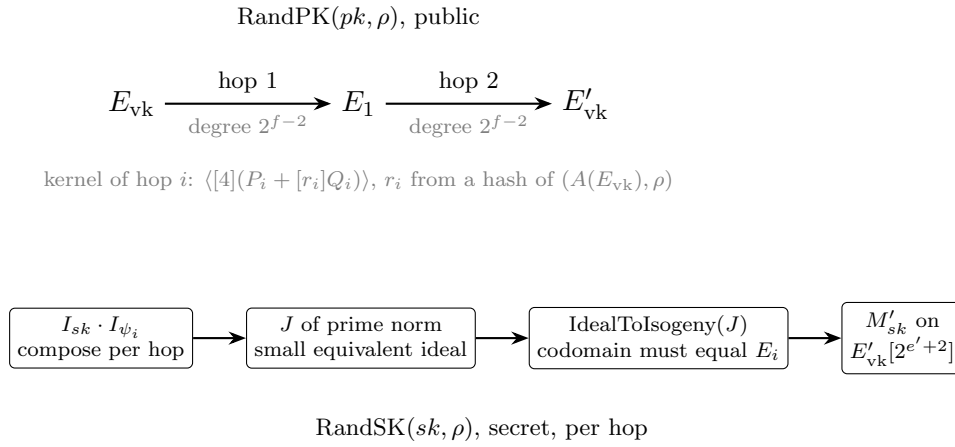


Figure 7: Randomisable keys. Top: RandPK walks two hops of degree 2^{f-2} from the public curve along kernels derived from a hash of the curve and the randomiser ρ ; anyone can run it. Bottom: RandSK follows the same walk on the ideal side, replacing the composed ideal by an equivalent one of odd prime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

Walk length Theorem 1 of [BBC⁺26b], 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. At the toy prime $5 \cdot 2^{14} - 1$ with 4 000 000 walks per length, one hop reaches 3072 of 6827 vertices at distance 0.550, two hops are within 0.004 of the sampling floor (0.0205 against 0.0165), four are at the floor (Figure 8). 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 ρ .

The key in the hash PRISM-RK has an adaptation algorithm [BBC⁺26b, Section 4.3] that moves a signature to a derived key without the signer, possible because its hash takes no key,

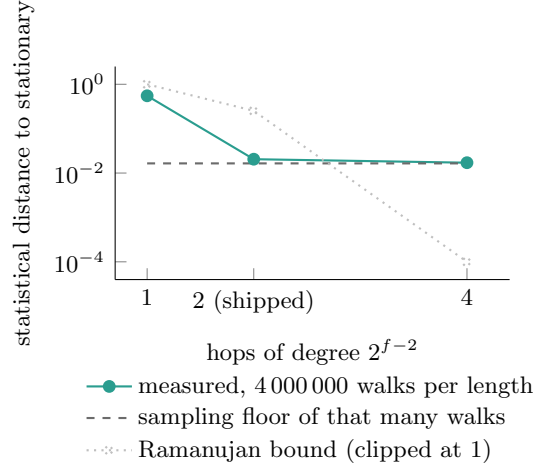


Figure 8: Statistical distance of the derived-key distribution to the stationary distribution at the toy prime $5 \cdot 2^{14} - 1$ against the number of hops, with the sampling floor and the bound; the shipped two hops sit on the flat part of the curve.

$q \leftarrow H(msg, salt)$ [BBC⁺26b, Figure 7, line 2]. Under a notion that counts (key, message, signature) tuples, adaptation is a forgery, so the hash input stays $(E'_{vk}, msg, salt)$ 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 ρ . Hierarchical derivation takes $\rho_i = \text{SHA3-256}(\text{“eclipse-hd”} \parallel pk_{\text{master}} \parallel i)$ 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 ρ_i from the master secret key and the index under “eclipse-hd-hardened” instead, so only the secret’s holder derives it. Stealth addresses take a uniform ρ of 2λ bits drawn by the sender, who pays to $\text{RandPK}(pk_{\text{recipient}}, \rho)$ and publishes ρ ; the recipient scans with RandPK and spends with RandSK, unlinkable to observers without ρ . A one-time key is the stealth profile with a fresh ρ 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 ρ . The argument of Section 7 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; it is ours and has the status of the base argument. Its checks are run. At all three levels, a derived key’s signature is rejected under the master key and under another derived key, and the master’s under a derived key; two derived keys signing one message share no salt and no tag. On the toy model, four signatures under two derived keys pass the exhaustive battery of Section 7: the sign flip, all 60 two-torsion shifts, the automorphism images, a wrong s and both wrong keys are rejected, and a second isogeny of the same degree is a distinct valid signature. The transfer is exact: the master’s challenge pushed forward along the walk is a degree- d isogeny from the derived curve that the isogeny check accepts with the master’s s , and its first hop equals $E_{\text{sig}} / \langle \sigma(K_1) \rangle$ computed geometrically; a derived signature’s challenge pulled back to the master curve likewise. What stops

the push-forward as a forgery is the hash: the derived key’s own digest of the message is another s' , under which the pushed isogeny is rejected, and a salt that makes the derived digest equal the master’s s took 51 and 27 tries against the digest’s $2^{e'-2} = 64$.

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 $e \leq f - 2$, 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 $\phi_A : E'_0 \rightarrow E_A$ 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^σ 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 $2^{f-2} \geq c$ [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 implementation [MIK26c] with that one set added reproduces the SageMath keys and shared secret byte for byte. 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 $\text{re} \parallel \text{im}$, 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^σ 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_{mix} , IdealTolsogeny onto the target, then the hint and M_{sk} at $2^{e'+2}$, with all randomness from $\text{SHAKE256}(\text{“ECLIPSE-mike-sk-v1”} \parallel s)$, so two derivations give the same bytes. The derived key’s public key is the re-encoding of the MIKE public key; the test checks it.

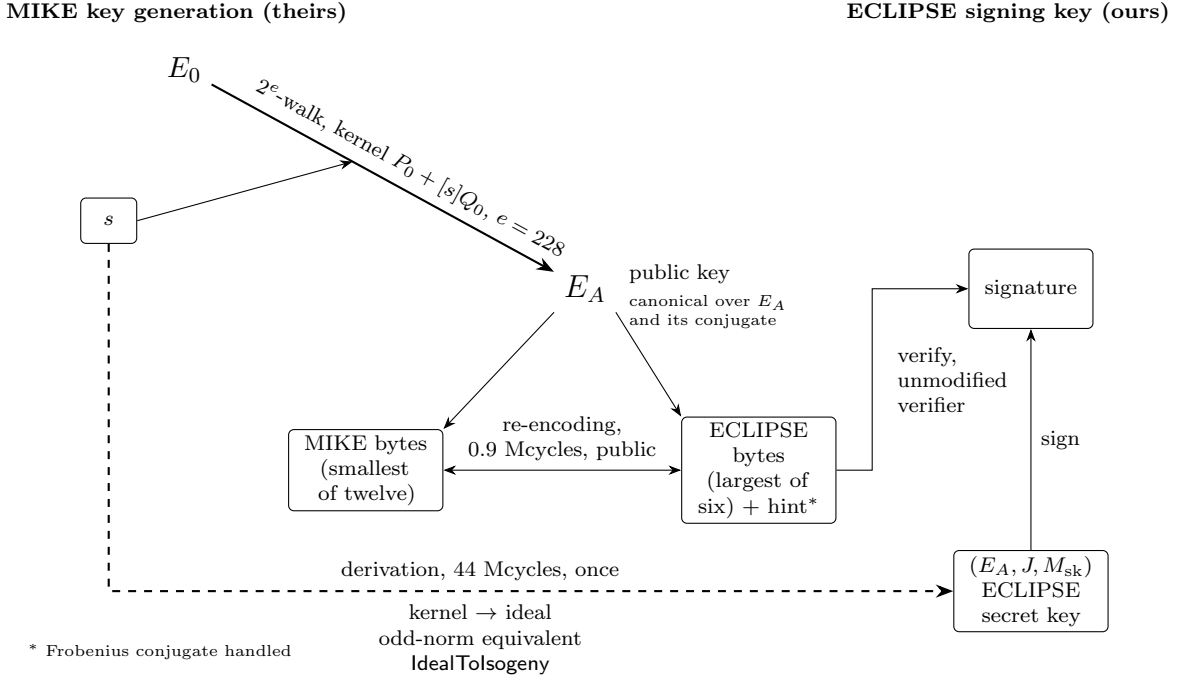


Figure 9: One curve in both roles at the shared prime. MIKE’s key generation (left) walks from E_0 to E_A ; the same curve re-encoded (middle) is an ECLIPSE public key, and the ECLIPSE secret key (right) is derived once from the MIKE secret. Figure 10 has the curve’s other uses.

What was checked Against MIKE’s Rust code at the shared set and the SageMath record of the same keys: the two key pairs and their shared secret agree byte for byte between the two MIKE implementations; the walk recomputed from each secret lands on the published coefficient (both keys

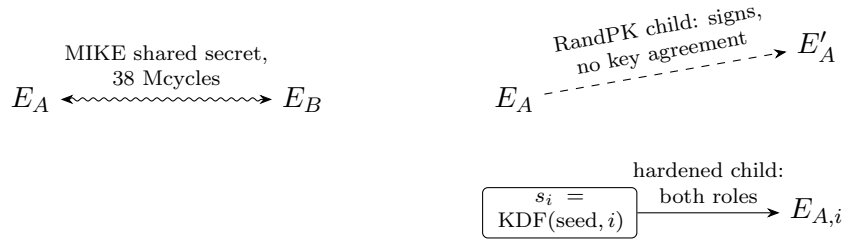


Figure 10: Other uses of the same curve: key agreement still runs on E_A , and derived children sign only, hardened children do both.

published the conjugate); re-encoding and reverse encoding are inverse, and the negated coefficient, the conjugate and a wrong length are rejected; the derived key publishes the re-encoded key, passes VerKey, is the same on a second derivation and differs from the other secret's; ten signatures per key verify and fifty negatives per key (another message, the other key, bytes flipped at the start, middle and end) are rejected; the SageMath verifier agrees on all twenty signatures and the two wrong- s variants exported with them, 22 of 22.

Cost Table 10 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 6. 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.

Table 10: 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_{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: $s_i = \text{KDF}(\text{seed}, i)$ 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 ECLIPSE’s own argument (Section 7) 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 establishes 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 11 gives both schemes in all four combinations: “compressed” is one coefficient recovered from Weil pairings, “BUFF” a λ -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.

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 12): 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 67/89/113 and 7.3/8.1/6.3 times the cycles, and the dimension-4 verifier of one code base runs about ten times

Table 11: 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, λ -bit tag)	231	333	429
	SQIsign v3 + λ -bit tag, computed ^b	216	330	438
compressed, BUFF	ECLIPSE v3, as shipped	206	300	388
	SQIsign v3, measured + λ -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.

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.

Table 12: 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)	67	89	113
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	198x	277x	345x
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)		

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. Two implementations accept each other’s signatures, the reroute attack forges salt-PRISM and produces nothing against ECLIPSE, and every tampering, mutation and fuzz input tried is rejected.

Future work The formal proof: a written SUF-CMA reduction to Problem 5 of [BBC⁺26a] along the three cases of Section 7, with its loss, and its extension to derived keys; the constant ω , derived and checked on every instance, proved rather than checked. The joint security of one curve as a MIKE key and an 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 checks 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 describe code as it is in September 2026.

Thanks To the PRISM authors, Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Cortes-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 ω

Write $\Phi_1 = \Psi_{e'} \circ \Phi_2 \circ \iota$ as in [BBC⁺26a, Section 5], $\Psi_{e'}$ the last step and ι the sign on the second factor; let A_1, A_2 be the codomains of Φ_1, Φ_2 as the theta-model chains compute them, and $\eta = \omega \circ \Psi_{e'}: A_2 \rightarrow A_2$. 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: η 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 ω gives the dual of A_2 ’s own (the Hadamard transform) with the two index bits exchanged by η (the swap of the middle coordinates). η 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 ω as the unique projective 4×4 matrix through ten point correspondences; solved at two 17-bit primes and at the level I prime it is the Hadamard transform followed by the swap, with unit scalars, every time. A derivation checked on every instance, not a proof.

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 work there as at the shared prime and the suites of Section 7 pass on both. 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; the existing backends regenerate byte for byte, and a differential test ran 10 000 operand sets per operation on both kernel paths.

Cost Table 13 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 10 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 `IdealTolsogeny` [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 10), 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.

Table 13: 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 10 (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 10	$3 \cdot 2^{324} - 1$	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 10	$3 \cdot 2^{324} - 1$	44.2 / 51.5	0.93	4.40 / 38.1

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 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 10). 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. *Cryptology ePrint Archive*, Paper 2026/443, 2026. Published in the *Journal of Cryptology* 39(4), article 30, 2026, doi 10.1007/s00145-026-09585-6. <https://eprint.iacr.org/2026/443>.
- [BBC⁺26b] 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 – EURO-CRYPT 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; authors listed alphabetically; no venue stated.
- [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.
- [C⁺26] Maria Corte-Real Santos et al. PRISM_v2: A C implementation of PRISM v2 (the salted variant). https://github.com/mariascrs/PRISM_v2, 2026. commit 87fad7da; salt-PRISM on the round-2 SQIsign primes.
- [CGH⁺25] Sofia 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. Undated in the text; the author’s file <https://www.math.u-bordeaux.fr/~hecohen/hardylw.dvi> is stamped 7 July 1998.
- [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>.
- [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.
- [I⁺26] Riccardo Invernizzi et al. 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.

- [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.
- [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.
- [MIK26c] MIKE team. mike_rs: MIKE in Rust. https://github.com/tensor-MIKE/mike_rs, 2026. Commit 0d79121, 2026-10-06; Apache-2.0.
- [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: tag `v0.6.34` at commit `078cc2af` (workspace version 0.6.34), the version bump on top of the 0.6 merge.
- [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>.