

TPOKÉ: Threshold Public-Key Encryption from POKÉ via Isogeny Sharing

Toshiki Takatera ^{*}, Hiroshi Onuki , and Tsuyoshi Takagi

Graduate School of Information Science and Technology, The University of Tokyo
takatera-toshiki527@g.ecc.u-tokyo.ac.jp
{onuki,takagi}@mist.i.u-tokyo.ac.jp

Abstract. POKÉ, introduced by Basso and Maino, is one of the most efficient isogeny-based public-key encryption schemes. The generic parallel-encryption approach yields a (t, n) -threshold variant by running n independent POKÉ instances and secret-sharing the message, but this multiplies the public-key and ciphertext sizes by roughly n . We present S-TPOKÉ, a POKÉ-based (t, n) -threshold construction for every $1 \leq t \leq n$, and Ch-TPOKÉ, an (n, n) variant that arranges the parties' secret isogenies differently. Both schemes avoid repeating part of the isogeny computation that appears in every independent POKÉ instance. Compared with the trivial construction, they reduce the number of degree-3^b isogenies computed during encryption from $2n$ to $n + 1$ and reduce the ciphertext size by about a factor of two when n is large. The two schemes differ in how the parties' secret isogenies are arranged. S-TPOKÉ is our main construction: the dealer samples n independent secret isogenies, while encryption samples one sender-side secret isogeny and shares it across the parties. Ch-TPOKÉ instead arranges the parties' secret isogenies into a chain. For S-TPOKÉ, we prove SIM-CPA security under static corruption in the random oracle model from the (k, n) -S-C-POKE assumptions for all $1 \leq k \leq t$, which extend C-POKE to the threshold setting. For Ch-TPOKÉ, we prove only SIM-CPA security under static maximal corruption from the (j, n) -Ch-D-POKE assumptions for all $j \in \{1, \dots, n\}$, which extend D-POKE to a chain-based threshold setting. This guarantee is strictly weaker than SIM-CPA security under static corruption. We also give asymptotic parameter choices targeting λ -bit security and derive closed-form expressions for the public-key, ciphertext, and partial-decryption sizes in terms of λ and n .

Keywords: Post-quantum cryptography · Isogeny-based cryptography · Threshold public-key encryption

1 Introduction

For any positive integers $n \geq t$, a (t, n) -threshold public-key encryption scheme $((t, n)$ -TPKE) distributes decryption capability among n parties so that the

^{*} Corresponding author.

plaintext can be reconstructed from any t valid partial decryptions, while it remains computationally hidden from adversaries corrupting fewer than t parties. Its main purpose is to mitigate key compromise by forcing an adversary to corrupt at least t parties before decryption becomes possible. NIST’s call for multi-party threshold schemes includes TPKE among its categories, with particular interest in threshold-friendly post-quantum PKE [11, Sections 2 and 11.2].

Threshold public-key encryption dates back to the ElGamal/Shamir-based construction of Desmedt and Frankel [28]. Later works achieved, for example, the IND-CCA security from Diffie–Hellman [56], validity checks for partial decryptions [31,32], and constructions from identity-based encryption without random oracles [7]. Dodis and Katz also gave a generic black-box construction of IND-CCA-secure (t, n) -TPKE from labeled IND-CCA-secure public-key encryption, secret sharing, and a one-time signature via parallel encryption [30], but its encryption and ciphertext costs typically grow with the number of parties.

At the same time, many of the most efficient TPKE schemes are based on the discrete logarithm problem or the integer factorization problem, and therefore are not secure against quantum computers because of Shor’s algorithm [55]. Hence, the main challenge is to construct TPKE schemes that are efficient while relying on quantum-resistant assumptions. Such post-quantum TPKE schemes have been proposed from lattices [5,17,18,39,42], codes [58], and isogenies [27]. In this work, we focus on isogeny-based TPKE schemes.

In 2020, De Feo and Meyer [27] proposed an isogeny-based (t, n) -TPKE in the Hard Homogeneous Spaces (HHS) framework [20], instantiated with CSI-FiSh [6]. While the construction provides threshold cryptography in the class-group-action setting, the class-group precomputation restricts the CSI-FiSh instantiation to CSIDH-512 [27, Section 4.2], whose claimed NIST level 1 quantum security has been challenged [8,50]. Subsequent schemes such as PEGASIS [23] and qt-Pegasis [24] show that HHS/group-action approaches remain viable, but the black-box HHS TPKE decryption remains sequential, so partial decryptions cannot be computed independently in parallel.

SIDH-like protocols have also inspired several isogeny-based public-key encryption schemes. SIDH [34] and its KEM variant SIKE [33] were broken by polynomial-time key-recovery attacks [15,40,52]. The higher-dimensional isogenies behind these attacks became constructive tools for new designs, especially via Kani’s lemma [35]. This led to two main directions. One builds trapdoored one-way functions directly from the SIDH attack mechanism, as in FESTA [4]. The other consists of ElGamal-type schemes built on SIDH-like commutative diagrams, including POKÉ [3], INKE [36], PIKE [14], and LIT-SiGamal [46].

We take POKÉ as our starting point because it has compact public keys and ciphertexts and because its square commutative diagram lets our threshold constructions share a sender-side isogeny across parties [3]. A black-box threshold construction can be obtained by running independent POKÉ instances in parallel, following the generic paradigm of Dodis and Katz [30]. This, however, requires a separate public key and a separate computation of a sender-side secret isogeny for every party. Our goal is to obtain threshold schemes that reduce

the ciphertext size and the number of isogeny computations performed during encryption while providing SIM-CPA guarantees.

1.1 Our Contribution

We first propose S-TPOKÉ, a POKÉ-based TPKE scheme. We prove its SIM-CPA security under static corruption in the random oracle model from the (k, n) -S-C-POKE assumptions for all $1 \leq k \leq t$, which extend C-POKE to the threshold setting. The scheme starts from the Dodis–Katz parallel-encryption approach instantiated with POKÉ [30]: key generation creates n independent party keys, but encryption samples one sender-side secret isogeny and shares it across the n parties. S-TPOKÉ keeps the public-key size and the parallel decryption structure of the trivial construction. It reduces the number of degree- 3^b isogenies computed during encryption from $2n$ to $n + 1$ and reduces the ciphertext size by roughly a factor of two when n is large. For $t < n$, the construction retains the n independent POKÉ key pairs and applies Shamir secret sharing to the message.

We also give Ch-TPOKÉ, an (n, n) construction that shares the sender’s isogeny along a chain instead. Ch-TPOKÉ is obtained from the Dodis–Katz parallel-encryption approach instantiated with POKÉ by replacing the n independent secret isogenies of that approach with a chain of n secret isogenies. This chain structure reduces the number of elliptic-curve points included in the public key relative to S-TPOKÉ while leaving the ciphertext size unchanged. Under the (j, n) -Ch-D-POKE assumptions for all $j \in \{1, \dots, n\}$, which extend D-POKE to a chain-based threshold setting, we prove only SIM-CPA security under static maximal corruption for Ch-TPOKÉ; this guarantee is strictly weaker than SIM-CPA security under static corruption. Because Ch-TPOKÉ reduces the number of public-key points at the cost of larger partial decryptions and more reconstruction work, S-TPOKÉ is our main construction; Ch-TPOKÉ illustrates chain-based isogeny sharing.

The star-shaped and chain-based forms of isogeny sharing introduce correlated information absent from a single POKÉ instance. S-C-POKE models star-shaped instances sharing the sender-side isogeny, whereas Ch-D-POKE models a chain in which all but one edge is exposed. We do not establish the hardness of these general threshold problems from C-POKE or D-POKE via black-box reductions; doing so remains open.

Our security results establish SIM-CPA security as formalized in Section 2.4. We do not claim SIM-CCA security or robustness against maliciously generated ciphertexts. The generic transformation of Brzuska, Klooß, and Woo [13] gives a theoretical route to SIM-CCA security using additional generic primitives, whose efficient instantiation for TPOKÉ remains open.

The rest of the paper is organized as follows. Section 2 recalls the required isogeny and TPKE background. Sections 3 and 4 present S-TPOKÉ and Ch-TPOKÉ, respectively, and provide the corresponding security proofs. Section 5 analyzes concrete attacks, Section 6 selects parameters and compares the constructions, and Section 7 concludes.

2 Preliminaries

In this section, we introduce the mathematical and cryptographic preliminaries used in this paper. Section 2.1 fixes notation, and Section 2.2 describes the isogeny representations used in our constructions. Section 2.3 reviews the relevant interface of POKÉ, and Section 2.4 recalls threshold public-key encryption and SIM-CPA security. We use standard facts about elliptic curves and isogenies [57] and principally polarized abelian surfaces [43].

2.1 Notation

We use the following notation throughout the paper.

For a positive integer n , we write $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z}$. For a ring R , $R^\times$ and $\mathrm{SL}_2(R)$ denote its group of units and special linear group, respectively, and $\mathrm{diag}(u, v)$ denotes the 2×2 diagonal matrix with diagonal entries $u, v \in R$. For a finite set A , we write $a \xleftarrow{\$} A$ to mean that a is sampled uniformly at random from A . If E is an elliptic curve over a field K and N is a positive integer with $\mathrm{char}(K) \nmid N$, then $E[N]$ denotes the N -torsion subgroup of E , and by a basis (P, Q) of $E[N]$ we mean an ordered generating pair. For a matrix $M = (m_{k\ell}) \in \mathrm{M}_2(\mathbb{Z}_N)$ and points $P, Q \in E[N]$, we write $M[P, Q]^\top := [[m_{11}]P + [m_{12}]Q, [m_{21}]P + [m_{22}]Q]^\top$. Finally, we write e_N for the Weil pairing on $E[N]$.

2.2 Computation of Isogenies

For isogenies $\phi : E \rightarrow E_1$ and $\psi : E \rightarrow E_2$ of coprime degrees, we write $\psi_*\phi : E_2 \rightarrow E_3$ for the pushforward of ϕ along ψ , which satisfies $\ker \psi_*\phi = \psi(\ker \phi)$. The codomain E_3 coincides with that of the pushforward $\phi_*\psi$ up to isomorphism, and we identify isomorphic curves throughout.

Definition 1 (Based on Definition 1 in [3]). *Let q be a prime power, and let $\phi : E \rightarrow E'$ be an isogeny between elliptic curves over $\mathbb{F}_q$. An efficient representation of ϕ is some data $D \in \{0, 1\}^*$ of polynomial size in $\log_2(\deg \phi)$ and $\log_2 q$ such that the following algorithms exist:*

- a polynomial-time algorithm that, on input D , produces the domain and codomain of ϕ ;
- a polynomial-time algorithm that, on input D , produces the degree $\deg \phi$;
- an algorithm that, on input D and a point $P \in E(\mathbb{F}_{q^r})$, computes $\phi(P)$ in polynomial time in $r \log_2 q$ and $\log_2(\deg \phi)$.

We use two concrete types of efficient representation below.

First, fix a finite field $\mathbb{F}_q$, polynomially small primes $\ell_1, \dots, \ell_n$, and positive integers $e_1, \dots, e_n$. For an isogeny of degree $\prod_{i=1}^n \ell_i^{e_i}$, a generating set for its kernel consisting of points rational over $\mathbb{F}_q$ is called a *kernel representation*. Such a representation is evaluated using $O(\sum_{i=1}^n e_i \ell_i \log_2(e_i + 1))$ operations in $\mathbb{F}_q$, by combining Vélu's formulas [59] with balanced strategies for evaluating prime-power-degree isogenies [26, Sec. 4.2.2].

Second, we use a two-dimensional isogeny to obtain efficient representations of the isogenies in our schemes. The construction is based on the following form of Kani's lemma [35].

Theorem 1 (Kani's Lemma from [40]). *Let d_1, d_2 be positive integers with $\gcd(d_1, d_2) = 1$, and set $N := d_1 + d_2$. Let E_0, E_1, E'_0, E'_1 be elliptic curves connected by the following commutative diagram of isogenies, and define*

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_1} & E_1 \\ \downarrow \phi_2 & & \downarrow \phi'_2 \\ E'_0 & \xrightarrow{\phi'_1} & E'_1 \end{array} \quad \Phi := \begin{pmatrix} \phi_1 & -\hat{\phi}'_2 \\ \phi_2 & \hat{\phi}'_1 \end{pmatrix} : E_0 \times E'_1 \longrightarrow E_1 \times E'_0,$$

$$(P, Q) \mapsto (\phi_1(P) - \hat{\phi}'_2(Q), \phi_2(P) + \hat{\phi}'_1(Q)),$$

where $\deg \phi_i = \deg \phi'_i = d_i$ for $i \in \{1, 2\}$ and $\hat{\phi}'_i$ is the dual isogeny of ϕ'_i . Then Φ is an (N, N) -isogeny with $\ker \Phi = \{([-d_1]P, (\phi'_2 \circ \phi_1)(P)) \mid P \in E_0[N]\}$.

We apply the theorem with $N = 2^e$, $d_1 = q$, and $d_2 = 2^e - q$, as in FESTA [4] and QFESTA [48]. For any basis (P, Q) of $E_0[2^e]$, the tuple $(d_1, (P, (\phi'_2 \circ \phi_1)(P)), (Q, (\phi'_2 \circ \phi_1)(Q)))$ determines $\ker \Phi$ and hence Φ . Evaluating Φ on $E_0 \times \{0\}$ and $\{0\} \times E'_1$ yields evaluations of $\phi_1, \phi_2, \hat{\phi}'_1, \hat{\phi}'_2$. We call the tuple a *Kani representation* of $\phi_1, \phi_2, \hat{\phi}'_1, \hat{\phi}'_2$. When $E_0[2^e]$ is $\mathbb{F}_{p^2}$ -rational, Φ can be constructed from its kernel in $O(e \log_2 e)$ field operations in $\mathbb{F}_{p^2}$ [25].

Although Φ yields $\hat{\phi}'_2$ rather than ϕ'_2 , POKÉ shows how to evaluate ϕ'_2 from the Kani representation [3, Section 4.3, Algorithm 5, and Remark 6]. Hence, we also call the tuple above a *Kani representation* of $\phi'_2 \circ \phi_1$.

2.3 POKÉ

We recall only the interface of POKÉ [3] needed by our constructions and omit its isogeny-generation procedures and optimizations. Fix integers $a \geq 2$ and $b, c, f > 0$ such that $p = 2^a 3^b 5^c f - 1$ is prime. Then $p \equiv 3 \pmod{4}$, so the curve $E_0 : y^2 = x^3 + x$ is supersingular over $\mathbb{F}_{p^2}$. Fix bases $(P_0, Q_0), (R_0, S_0)$, and (X_0, Y_0) of $E_0[2^a], E_0[3^b]$, and $E_0[5^c]$, respectively. Fix also a key-derivation function KDF with output length ℓ , and set the message space to $\{0, 1\}^\ell$.

POKÉ uses the SIDH-like commutative diagram in Fig. 1 to derive the shared basis (X'_1, Y'_1) , whose KDF output masks the message. Key generation determines the receiver's secret isogeny $\phi_1 : E_0 \rightarrow E_1$, while encryption samples $\psi_0 : E_0 \rightarrow E'_0$ and computes $\psi_1 = (\phi_1)_* \psi_0$. During decryption, the receiver reconstructs $\phi'_1 = (\psi_0)_* \phi_1$ from the ciphertext and the secret key. Commutativity gives $X'_1 = [\delta_5] \phi'_1(X'_0)$ and $Y'_1 = [\delta_5] \phi'_1(Y'_0)$, so applying KDF to this basis un.masks the message.

In key generation, the receiver samples $q \in [1, 2^a - 1]$ such that $q(2^a - q)$ is coprime to 2, 3, 5, an isogeny $\phi_1 : E_0 \rightarrow E_1$ of degree $q(2^a - q)$, and $\alpha_2, \beta_2 \in \mathbb{Z}_{2^a}^\times$, $\gamma_3 \in \mathbb{Z}_{3^b}^\times$, $\delta_5 \in \mathbb{Z}_{5^c}^\times$. POKÉ gives two methods for generating ϕ_1 . The rigorous method samples a random left $\text{End}(E_0)$ -ideal of norm $q(2^a - q)$ and converts it

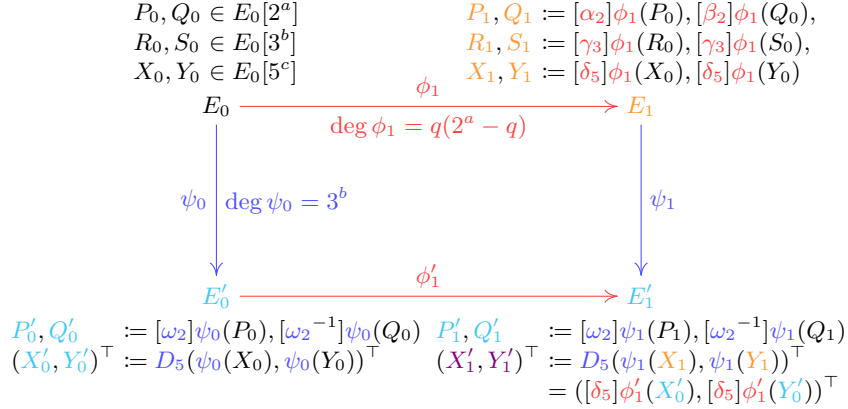


Fig. 1. Overview of POKÉ (adapted from [3]). Public parameters and annotations are in black. Elements in red are the receiver's secrets, elements in orange are part of the public key, elements in blue are ephemeral secrets used during encryption, and elements in cyan are part of the ciphertext. Purple indicates the shared basis, which is not transmitted.

into ϕ_1 with an ideal-to-isogeny algorithm, yielding a uniformly random isogeny of the required degree [3, Section 4.1]. The heuristic method derives ϕ_1 from an endomorphism of E_0 and is faster. Its security analysis assumes that the resulting public-key distribution is computationally indistinguishable from the distribution produced by the rigorous method [3, Section 5, Problem 10]. The receiver outputs $\text{sk} = (q, \alpha_2, \beta_2, \delta_5)$ and $\text{pk} = (E_1, P_1, Q_1, R_1, S_1, X_1, Y_1)$, where $(P_1, Q_1) = ([\alpha_2]\phi_1(P_0), [\beta_2]\phi_1(Q_0))$, $(R_1, S_1) = ([\gamma_3]\phi_1(R_0), [\gamma_3]\phi_1(S_0))$, $(X_1, Y_1) = ([\delta_5]\phi_1(X_0), [\delta_5]\phi_1(Y_0))$.

In encryption, the sender samples $r_3 \in \mathbb{Z}_{3^b}$ and constructs $\psi_0 : E_0 \rightarrow E'_0$ and $\psi_1 : E_1 \rightarrow E'_1$ with kernels $\langle R_0 + [r_3]S_0 \rangle$ and $\langle R_1 + [r_3]S_1 \rangle$, respectively. Since $R_1 = [\gamma_3]\phi_1(R_0)$ and $S_1 = [\gamma_3]\phi_1(S_0)$, $\ker \psi_1 = \phi_1(\ker \psi_0)$, and hence $\psi_1 = (\phi_1)_* \psi_0$. To mask the torsion-point images, the sender samples $\omega_2 \in \mathbb{Z}_{2^a}^\times$ and $D_5 \in \text{SL}_2(\mathbb{Z}_{5^c})$. For $i \in \{0, 1\}$, set $[P'_i, Q'_i]^\top = \text{diag}(\omega_2, \omega_2^{-1})[\psi_i(P_i), \psi_i(Q_i)]^\top$, $[X'_i, Y'_i]^\top = D_5[\psi_i(X_i), \psi_i(Y_i)]^\top$. The ciphertext is $\text{ct} = ((E'_0, P'_0, Q'_0, X'_0, Y'_0), (E'_1, P'_1, Q'_1), \text{ct})$, where $\text{ct} = \text{KDF}(X'_1, Y'_1) \oplus \text{msg}$.

For decryption, the receiver sets $P''_1 = [\alpha_2^{-1}]P'_1$ and $Q''_1 = [\beta_2^{-1}]Q'_1$. Then $(q, (P'_0, P''_1), (Q'_0, Q''_1))$ is a Kani representation of $\phi'_1 : E'_0 \rightarrow E'_1$. Evaluating this representation on X'_0, Y'_0 as in [3, Algorithm 5] gives $X'_1 = [\delta_5]\phi'_1(X'_0)$ and $Y'_1 = [\delta_5]\phi'_1(Y'_0)$; the receiver then recovers msg .

POKÉ is IND-CPA secure in the random oracle model under C-POKE [3, Theorem 8]. The C-POKE problem asks to recover the shared basis from the public key and ciphertext data [3, Problem 7]; D-POKE is its decisional variant [3, Problem 9].

$\text{Exp}_{\Pi, \mathcal{A}}^\tau(1^\lambda)$	$\text{ChalO}(\text{msg})$
$(\mathcal{C}, \text{st}_{\mathcal{A}}) \leftarrow \mathcal{A}(1^\lambda)$ // corrupt set $\mathcal{C}$ $\text{assert } \mathcal{C} \subseteq \{1, \dots, n\} \wedge \mathcal{C} < t$ $\text{if } \tau = 0 \text{ then } (\text{pk}, (\text{sk}_j)_{j=1}^n) \leftarrow \text{KGen}(1^\lambda)$ $\text{if } \tau = 1 \text{ then } (\text{pk}, (\text{sk}_j)_{j \in \mathcal{C}}, \text{st}_{\text{Sim}}) \leftarrow \text{Sim}(1^\lambda, \mathcal{C})$ $L_{\text{Enc}} \leftarrow \emptyset; L_{\text{Chal}} \leftarrow \emptyset; P[\cdot] \leftarrow \emptyset; M[\cdot] \leftarrow \emptyset$ $\text{st}_{\mathcal{A}} \leftarrow (\text{pk}, (\text{sk}_j)_{j \in \mathcal{C}}, \text{st}_{\mathcal{A}})$ $\text{return } \mathcal{A}^{\text{EncO}, \text{ChalO}, \text{ParDecO}}(\text{st}_{\mathcal{A}})$	$\text{if } \tau = 0 \text{ then } \text{ct} \leftarrow \text{Enc}(\text{pk}, \text{msg})$ $\text{if } \tau = 1 \text{ then } (\text{ct}, \text{st}_{\text{Sim}}) \leftarrow \text{Sim}(\text{st}_{\text{Sim}})$ $L_{\text{Chal}} \leftarrow L_{\text{Chal}} \cup \{\text{ct}\}; M[\text{ct}] \leftarrow \text{msg}$ return ct
$\text{EncO}(\text{msg})$	$\text{ParDecO}(\text{ct}, j)$
$\text{if } \tau = 0 \text{ then}$ $\quad \text{sample } \text{rnd}; \text{ct} \leftarrow \text{Enc}(\text{pk}, \text{msg}; \text{rnd})$ $\text{if } \tau = 1 \text{ then } (\text{ct}, \text{rnd}, \text{st}_{\text{Sim}}) \leftarrow \text{Sim}(\text{msg}, \text{st}_{\text{Sim}})$ $L_{\text{Enc}} \leftarrow L_{\text{Enc}} \cup \{\text{ct}\}; M[\text{ct}] \leftarrow \text{msg}$ $\text{return } (\text{ct}, \text{rnd})$	$\text{assert } \text{ct} \in L_{\text{Enc}} \cup L_{\text{Chal}}$ $P[\text{ct}] \leftarrow P[\text{ct}] \cup \{j\}$ $\text{if } \tau = 0 \text{ then } \text{pd}_j \leftarrow \text{ParDec}(\text{sk}_j, \text{ct})$ $\text{if } \tau = 1 \text{ then}$ $\quad \text{if } P[\text{ct}] \cup \mathcal{C} \geq t \wedge \text{ct} \in L_{\text{Chal}} \text{ then}$ $\quad \quad \text{msg} \leftarrow M[\text{ct}]$ $\quad \quad (\text{pd}_j, \text{st}_{\text{Sim}}) \leftarrow \text{Sim}(\text{ct}, \text{msg}, j, \text{st}_{\text{Sim}})$ $\quad \text{else } (\text{pd}_j, \text{st}_{\text{Sim}}) \leftarrow \text{Sim}(\text{ct}, j, \text{st}_{\text{Sim}})$ $\text{return } \text{pd}_j$

Fig. 2. SIM-CPA experiment for a (t, n) -TPKE Π , specialized from Figure 3 of Brzuska, Klooß, and Woo [13]. The simulator Sim maintains state st_{Sim} .

2.4 Threshold Public-Key Encryption

Following Brzuska, Klooß, and Woo [13, Definitions 1 and 3 and Figure 3], we recall the syntax of threshold public-key encryption (TPKE) and SIM-CPA security under static corruption, specialized to our fixed (t, n) setting. We also describe the trivial construction used in our comparison.

Definition 2. For positive integers $t \leq n$ and a message space $\mathcal{M}$, a (t, n) -TPKE consists of the following algorithms:

Key generation On input the security parameter 1^λ , $\text{KGen}(1^\lambda)$ outputs a public key pk and secret keys $(\text{sk}_i)_{i=1}^n$.

Encryption On input pk and $\text{msg} \in \mathcal{M}$, $\text{Enc}(\text{pk}, \text{msg})$ outputs a ciphertext ct .

Partial decryption On input sk_i and ct , $\text{ParDec}(\text{sk}_i, \text{ct})$ outputs a partial decryption pd_i .

Reconstruction On input pk , partial decryptions $(\text{pd}_i)_{i \in I}$ from an index set $I \subseteq \{1, \dots, n\}$, and ct , $\text{Rec}(\text{pk}, (\text{pd}_i)_{i \in I}, \text{ct})$ outputs a message msg .

For every $\text{msg} \in \mathcal{M}$ and $I \subseteq \{1, \dots, n\}$ with $|I| \geq t$, correctness requires

$$\Pr[\text{Rec}(\text{pk}, (\text{pd}_i)_{i \in I}, \text{ct}) = \text{msg}] = 1,$$

where $(\text{pk}, (\text{sk}_i)_{i=1}^n) \leftarrow \text{KGen}(1^\lambda)$, $\text{ct} \leftarrow \text{Enc}(\text{pk}, \text{msg})$, and $\text{pd}_i \leftarrow \text{ParDec}(\text{sk}_i, \text{ct})$ for every $i \in I$.

Definition 3. A (t, n) -TPKE Π is SIM-CPA secure (under static corruption) if there exists a probabilistic polynomial-time simulator Sim such that, for every probabilistic polynomial-time adversary $\mathcal{A}$, the experiments $\text{Exp}_{\Pi, \mathcal{A}}^0$ and $\text{Exp}_{\Pi, \mathcal{A}}^1$ in Fig. 2 are computationally indistinguishable. Restricting $\mathcal{A}$ to at most one ChalO query gives single-challenge SIM-CPA security; restricting it to $|\mathcal{C}| = t - 1$ gives SIM-CPA security under static maximal corruption.

Following Brzuska, Klooß, and Woo [13, Definition 4 and Theorem 3], we also consider SIM-CPA security under static maximal corruption, a strictly weaker notion than SIM-CPA security under static corruption. We prove S-TPOKÉ SIM-CPA secure under static corruption. For Ch-TPOKÉ, we prove only SIM-CPA security under static maximal corruption.

CPA-to-CCA transformation for TPKE. In the random oracle model, Brzuska, Klooß, and Woo give a generic SIM-CPA-to-SIM-CCA transformation for TPKE based on a simulation-extractable NIPoR proving that a ciphertext was generated using fresh, honestly sampled randomness [13, Theorem 10]. Such a NIPoR can be built from NIZKs and CCA-secure public-key encryption, as established in [13, Theorem 9 and Section 5.3] and [29, Section 4.4]. NIZKs for NP exist under standard post-quantum assumptions [21, 51], but efficiently realizing the transformation requires an efficient NIPoR, obtained directly or via an efficient NIZK. Although Brzuska et al. do not state this transformation for static maximal corruption, the proof in their full version forwards the corrupt set unchanged, so it also applies when both notions are restricted to static maximal corruption [12, Appendix F.2].

Trivial construction. Let $\mathcal{SS}$ be a perfect (t, n) secret-sharing scheme whose shares lie in the message space of an IND-CPA-secure PKE. Encrypting them under independent public keys yields a (t, n) -TPKE via the share-then-encrypt template of Dodis and Katz [30]; we call its POKÉ instantiation the *trivial construction*. Each party decrypts one share, and reconstruction recovers the message from any t shares. We omit the CCA-secure Dodis–Katz variant, which adds labeled encryption and a one-time strong EUF-CMA signature [30, Theorem 1].

For $t = n$, let $K_i := \text{KDF}(X'_i, Y'_i)$ be the POKÉ mask for $\mathcal{P}_i$. The n independent POKÉ encryptions would contain the payloads $ct_i = \text{msg}_i \oplus K_i$. We replace them with their XOR $ct' := \bigoplus_{i=1}^n ct_i = \text{msg} \oplus K_1 \oplus \dots \oplus K_n$. For $t < n$, we use Shamir secret sharing over $\mathbb{F}_{2^\ell}$ [54].

3 S-TPOKÉ

We present S-TPOKÉ and prove its SIM-CPA security under static corruption.

3.1 Overview of S-TPOKÉ

S-TPOKÉ uses the POKÉ public parameters. Fix positive integers $a \geq 2, b, c, f$ such that $p := 2^a 3^b 5^c f - 1$ is prime, and let the supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ be defined by $y^2 = x^3 + x$. Fix bases $(P_0, Q_0), (R_0, S_0), (X_0, Y_0)$ of $E_0[2^a], E_0[3^b], E_0[5^c]$. Section 6 specifies the parameter sizes as functions of λ .

As illustrated in Fig. 3, S-TPOKÉ retains the n independent secret isogenies $\phi_i : E_0 \rightarrow E_i$ of the trivial construction from Section 2.4, but uses one degree- 3^b isogeny $\psi_0 : E_0 \rightarrow E'_0$ for all parties and sets $\phi'_i := (\psi_0)_* \phi_i : E'_0 \rightarrow E'_i$.

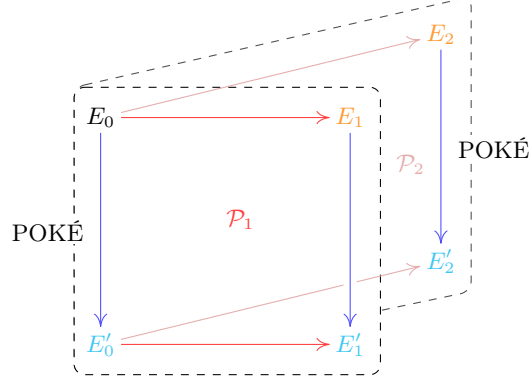


Fig. 3. A sketch of (2,2)-S-TPOKÉ. Public parameters and annotations are in black. Red and pink indicate $\mathcal{P}_1$ and $\mathcal{P}_2$ and their secrets, respectively. Elements in orange are part of the public key. Elements in blue are ephemeral secrets used during encryption, and elements in cyan are part of the ciphertext.

Thus the ciphertext contains the data on E'_0 once and the data on E'_i for each i . Party $\mathcal{P}_i$ uses its secret key and the ciphertext to reconstruct ϕ'_i , recovers (X'_i, Y'_i) using ϕ'_i , and outputs $\text{KDF}(X'_i, Y'_i)$; for general (t, n) , the sender secret-shares the message and masks share i with $\text{KDF}(X'_i, Y'_i)$.

Since only ψ_0 is shared while the ϕ_i remain independent, the construction also applies to the square-diagram schemes INKE [36], PIKE [14], and LIT-SiGamal [46].

3.2 S-TPOKÉ Algorithms

We now specify S-TPOKÉ. $\{\text{KGen}, \text{Enc}, \text{ParDec}, \text{Rec}\}$. For $t < n$, assume $n < 2^\ell$ and let SS be Shamir's (t, n) secret-sharing scheme over $\mathbb{F}_{2^\ell}$, with algorithms SS.Share and SS.Rec [54]. We identify $\mathbb{F}_{2^\ell}$ with $\{0, 1\}^\ell$.

Key Generation Algorithm The algorithm S-TPOKÉ.KGen does not depend on t . For $i = 1, \dots, n$, the dealer repeatedly samples q_i uniformly from $[1, 2^a - 1]$ until q_i and $2^a - q_i$ are prime and $q_i(2^a - q_i)$ is coprime to 2, 3, 5. The same sampling method is used in Threshold PRISM [37, Section 4.3.1]. Requiring both q_i and $2^a - q_i$ to be prime prevents the attack discussed under *Threshold-specific attacks* in Section 5. This condition preserves efficient key generation: the Hardy–Littlewood estimate gives approximately $2C_2 2^a / (a \ln 2)^2$ candidates for q_i , where $C_2 \approx 0.66$ is the twin-prime constant [37, Section 4.3.1]. Thus, the rejection sampler has an acceptance probability of order $1/a^2$, and its expected number of trials is polynomial in a .

The dealer then uses either the rigorous or the heuristic method described in Section 2.3 to complete POKÉ.KGen with q fixed to q_i and obtain $(\text{sk}_i, \text{pk}_i)$;

write $\phi_i : E_0 \rightarrow E_i$ for the secret isogeny. The dealer gives sk_i to party $\mathcal{P}_i$ and publishes $\text{pk} := (\text{pk}_1, \dots, \text{pk}_n)$.

Encryption Algorithm On input $\text{pk} = (\text{pk}_1, \dots, \text{pk}_n)$ and msg , the algorithm S-TPOKÉ.Enc samples $r_3 \in \mathbb{Z}_{3^b}$ and constructs a single isogeny $\psi_0 : E_0 \rightarrow E'_0$ with $\ker \psi_0 = \langle R_0 + [r_3]S_0 \rangle$. It also samples $\omega_2 \in \mathbb{Z}_{2^a}^\times$ and $D_5 \in \text{SL}_2(\mathbb{Z}_{5^c})$, and sets $P'_0 := [\omega_2]\psi_0(P_0)$, $Q'_0 := [\omega_2^{-1}]\psi_0(Q_0)$, $[X'_0, Y'_0]^\top := D_5[\psi_0(X_0), \psi_0(Y_0)]^\top$.

For each $i = 1, \dots, n$, using the public-key component pk_i , it constructs $\psi_i : E_i \rightarrow E'_i$ with kernel $\langle R_i + [r_3]S_i \rangle$. Since R_i and S_i use the same mask $\gamma_{3,i}$, we have $\psi_i = (\phi_i)_* \psi_0$. It computes the ciphertext points $P'_i := [\omega_2]\psi_i(P_i)$ and $Q'_i := [\omega_2^{-1}]\psi_i(Q_i)$, the shared basis $[X'_i, Y'_i]^\top := D_5[\psi_i(X_i), \psi_i(Y_i)]^\top$, and the mask $K_i := \text{KDF}(X'_i, Y'_i)$.

For the direct (n, n) construction, the message is encrypted as $ct' := \text{msg} \oplus K_1 \oplus \dots \oplus K_n$. The ciphertext is $ct := ((E'_0, P'_0, Q'_0, X'_0, Y'_0), (E'_1, P'_1, Q'_1), \dots, (E'_n, P'_n, Q'_n), ct')$.

For the (t, n) version with $t < n$, the sender computes $(\text{msg}_1, \dots, \text{msg}_n) \leftarrow \text{SS.Share}(\text{msg})$ and sets $ct_i := \text{msg}_i \oplus K_i$ for each $i = 1, \dots, n$. The ciphertext is $ct := ((E'_0, P'_0, Q'_0, X'_0, Y'_0), (E'_1, P'_1, Q'_1), \dots, (E'_n, P'_n, Q'_n), (ct_1, \dots, ct_n))$.

Partial Decryption Algorithm Given sk_i and a ciphertext ct , party $\mathcal{P}_i$ parses the data on E'_0 and the component (E'_i, P'_i, Q'_i) . It follows the POKÉ decryption algorithm from Section 2.3 to recover the basis (X'_i, Y'_i) and outputs the partial decryption $\text{pd}_i := \text{KDF}(X'_i, Y'_i)$.

Reconstruction Algorithm For the direct (n, n) construction, on input all n partial decryptions, the reconstruction algorithm parses the final component ct' of ct and outputs $\text{msg} := ct' \oplus \text{pd}_1 \oplus \dots \oplus \text{pd}_n$; if fewer than n partial decryptions are supplied, it outputs $\perp$.

For the (t, n) version with $t < n$, if $|I| < t$, the reconstruction algorithm outputs $\perp$. Otherwise, for each $i \in I$, it computes $\text{msg}_i := ct_i \oplus \text{pd}_i$. It then outputs $\text{SS.Rec}(I, \{(i, \text{msg}_i)\}_{i \in I})$.

Correctness. Correctness follows from that of POKÉ on each square, since sharing ψ_0 across the squares does not change the POKÉ decryption relation. For $t < n$, it additionally follows from the correctness of SS.

3.3 SIM-CPA Security of S-TPOKÉ

In this subsection, we introduce a star-shaped computational generalization of C-POKE and use it to state the SIM-CPA guarantee of S-TPOKÉ. The special case $(1, 1)$ coincides with C-POKE conditioned on q and $2^a - q$ being prime.

Problem 1. For integers n, k with $n \geq k \geq 1$, the (k, n) -S-C-POKE problem is defined as follows.

- Let E_0 be a supersingular elliptic curve with known endomorphism ring, and let $(P_0, Q_0), (R_0, S_0), (X_0, Y_0)$ be bases of $E_0[2^a], E_0[3^b], E_0[5^c]$, respectively.
- For each $i \in \{1, \dots, n\}$, sample q_i uniformly from $[1, 2^a - 1]$ conditioned on q_i and $2^a - q_i$ being prime and $q_i(2^a - q_i)$ being coprime to 2, 3, 5, and sample an isogeny $\phi_i : E_0 \rightarrow E_i$ of degree $q_i(2^a - q_i)$ uniformly at random.
- For each $i \in \{1, \dots, n - k + 1\}$, choose uniformly at random $\alpha_{2,i}, \beta_{2,i} \in \mathbb{Z}_{2^a}^\times$, $\gamma_{3,i} \in \mathbb{Z}_{3^b}^\times$, $\delta_{5,i} \in \mathbb{Z}_{5^c}^\times$, and define $P_i := [\alpha_{2,i}] \phi_i(P_0)$, $Q_i := [\beta_{2,i}] \phi_i(Q_0)$, $R_i := [\gamma_{3,i}] \phi_i(R_0)$, $S_i := [\gamma_{3,i}] \phi_i(S_0)$, $X_i := [\delta_{5,i}] \phi_i(X_0)$, $Y_i := [\delta_{5,i}] \phi_i(Y_0)$.
- Sample $r_3 \xleftarrow{\$} \mathbb{Z}_{3^b}$ and let $\psi_0 : E_0 \rightarrow E'_0$ be the isogeny with kernel $\langle R_0 + [r_3]S_0 \rangle$. Define $\psi_i := (\phi_i)_* \psi_0 : E_i \rightarrow E'_i$ for $1 \leq i \leq n$.
- For each $i \in \{1, \dots, n\}$, let $\phi'_i := (\psi_0)_* \phi_i : E'_0 \rightarrow E'_i$.
- Choose uniformly at random $\omega_2 \in \mathbb{Z}_{2^a}^\times$ and $D_5 \in \text{SL}_2(\mathbb{Z}_{5^c})$ and, for each $i \in \{0, \dots, n - k + 1\}$, define $P'_i := [\omega_2] \psi_i(P_i)$, $Q'_i := [\omega_2^{-1}] \psi_i(Q_i)$, $[X'_i, Y'_i]^\top := D_5[\psi_i(X_i), \psi_i(Y_i)]^\top$.

Given as input $E_0, \dots, E_n, E'_0, \dots, E'_n, ((\phi_i, q_i))_{i=n-k+2, \dots, n}, (\phi'_i)_{i=n-k+2, \dots, n}, (P_0, Q_0, R_0, S_0, X_0, Y_0), ((P_i, Q_i, R_i, S_i, X_i, Y_i))_{i=1, \dots, n-k+1}, (P'_0, Q'_0, X'_0, Y'_0)$, and $((P'_i, Q'_i))_{i=1, \dots, n-k+1}$, compute an index $i \in \{1, \dots, n - k + 1\}$ together with the points X'_i, Y'_i , up to automorphism of E'_i .

The theorem below states the SIM-CPA guarantee when S-TPOKÉ uses the rigorous method for key generation. If S-TPOKÉ instead uses the heuristic method for key generation, we assume, as in POKÉ, that its public-key distribution is computationally indistinguishable from the uniformly random distribution produced by the rigorous method [3, Section 5, Problem 10].

Theorem 2. *Suppose KDF is modeled as a random oracle and the (k, n) -S-C-POKE problem is hard for all $1 \leq k \leq t$. Then (t, n) -S-TPOKÉ is multi-challenge SIM-CPA secure.*

Proof. We prove the claim for the general share-based (t, n) construction with $t < n$; the direct (n, n) construction is handled at the end.

We first define the stateful simulator. On initial input $(1^\lambda, \mathcal{C})$, it runs the key-generation algorithm honestly, returns pk and $(\text{sk}_i)_{i \in \mathcal{C}}$, and retains all secret keys. It answers EncO and the associated ParDecO queries honestly. For ChalO , it independently samples $r_3 \xleftarrow{\$} \mathbb{Z}_{3^b}$, $\omega_2 \xleftarrow{\$} \mathbb{Z}_{2^a}^\times$, $D_5 \xleftarrow{\$} \text{SL}_2(\mathbb{Z}_{5^c})$, $(ct_1, \dots, ct_n) \xleftarrow{\$} (\{0, 1\}^\ell)^n$ and forms the curve and torsion components as in encryption. For a challenge ciphertext ct , it answers and caches ParDecO responses honestly while $|P[\text{ct}] \cup \mathcal{C}| < t$.

When $|P[\text{ct}] \cup \mathcal{C}|$ first reaches t , the simulator receives msg . Let J contain the $t - 1$ previously queried or corrupted indices and set $\text{msg}_i := ct_i \oplus \text{pd}_i$ for $i \in J$, using honest partial decryptions for corrupted indices. Together with $f(0) = \text{msg}$, the conditions $f(i) = \text{msg}_i$ for $i \in J$ determine a unique polynomial f of degree less than t . For the newly queried and all subsequent honest parties, the simulator returns $\text{pd}_i := ct_i \oplus f(i)$.

We next prove single-challenge security. Let $\mathcal{A}$ be a PPT adversary with non-negligible single-challenge advantage. By averaging over $|\mathcal{C}| \in \{0, \dots, t - 1\}$,

fix $k \in \{1, \dots, t\}$ such that the product of $\Pr[|\mathcal{C}| = k - 1]$ and the advantage conditioned on this event is non-negligible. We construct an algorithm $\mathcal{B}$ for the (k, n) -S-C-POKE problem; $\mathcal{B}$ aborts unless $|\mathcal{C}| = k - 1$.

The algorithm $\mathcal{B}$ receives a (k, n) -S-C-POKE instance. After $\mathcal{A}$ outputs $\mathcal{C}$, $\mathcal{B}$ permutes the indices so that the $k - 1$ indices for which the instance gives the explicit isogenies ϕ_i, ϕ'_i are exactly the corrupted indices. For each $i \in \mathcal{C}$, $\mathcal{B}$ constructs $(\text{sk}_i, \text{pk}_i)$ from q_i, ϕ_i as in S-TPOKE.KGen and uses ϕ'_i to compute the challenge values $P'_i := [\alpha_{2,i}]\phi'_i(P'_0)$ and $Q'_i := [\beta_{2,i}]\phi'_i(Q'_0)$. For $i \notin \mathcal{C}$, the public-key components and the challenge values P'_i, Q'_i are taken from the S-C-POKE instance. Hence the public key and the corrupted secret keys have the same distribution as in the real experiment.

The algorithm $\mathcal{B}$ implements KDF as a lazy random oracle and records all queries. It answers every EncO query by honest S-TPOKE encryption and stores the encryption randomness; for ParDecO on such ciphertexts, it uses this randomness to recompute the honest masks. When the unique ChalO(msg) query is made, $\mathcal{B}$ samples $ct_1, \dots, ct_n \xleftarrow{\$} \{0, 1\}^\ell$ independently and returns the ciphertext $\text{ct} := ((E'_0, P'_0, Q'_0, X'_0, Y'_0), (E'_1, P'_1, Q'_1), \dots, (E'_n, P'_n, Q'_n), (ct_1, \dots, ct_n))$. This is the simulator's challenge distribution. $\mathcal{B}$ answers corrupted-party ParDecO queries honestly. For a query ParDecO(ct, i) on the challenge ciphertext with $i \notin \mathcal{C}$ and $|P[\text{ct}] \cup \mathcal{C}| < t$, $\mathcal{B}$ does not know the basis (X'_i, Y'_i) and therefore returns and caches a uniform string. This has the same distribution as $\text{KDF}(X'_i, Y'_i)$ unless $\mathcal{A}$ queries KDF at (X'_i, Y'_i) . Once $|P[\text{ct}] \cup \mathcal{C}| \geq t$, $\mathcal{B}$ receives msg and completes the sharing as above.

It remains to compare this with the real $\tau = 0$ single-challenge experiment. Suppose $\mathcal{A}$ never queries KDF at (X'_i, Y'_i) for an honest $i \notin \mathcal{C}$. While $|P[\text{ct}] \cup \mathcal{C}| < t$, every unexposed mask makes its payload component uniform, and the values $(\text{msg}_i)_{i \in \mathcal{C} \cup P[\text{ct}]}$ are uniform by the privacy of Shamir sharing; once $|P[\text{ct}] \cup \mathcal{C}|$ reaches t , the completed sharing has exactly the conditional distribution of an honest sharing of msg. Hence the challenge ciphertext and the entire partial-decryption transcript are distributed as in the simulated experiment. Thus, any non-negligible distinguishing advantage implies that $\mathcal{A}$ queries KDF at (X'_i, Y'_i) for some $i \notin \mathcal{C}$ with non-negligible probability. Consequently, choosing uniformly among $\mathcal{A}$'s polynomially many KDF queries lets $\mathcal{B}$ solve S-C-POKE with non-negligible probability.

S-TPOKE's single-challenge security extends to multi-challenge security via the standard hybrid argument. Such an implication can fail for TPKE in general when a simulator's response for one challenge depends on the order of partial-decryption queries for other challenges [13, Section 3.4 and Theorem 5]; this issue does not arise here, since the simulator maintains independent state for each challenge. Since the KDF inputs for distinct challenges collide only with negligible probability, the standard hybrid argument applies as in multi-challenge PKE.

It remains to cover the direct (n, n) construction. The simulator has the same initialization and EncO behavior as above. For ChalO, it samples the curve and torsion components as above and $ct' \xleftarrow{\$} \{0, 1\}^\ell$. It answers and caches chal-

lenge partial decryptions honestly while $|P[\text{ct}] \cup \mathcal{C}| < n$. When a query for the last missing party i makes $|P[\text{ct}] \cup \mathcal{C}| = n$, it receives msg and returns $\text{pd}_i := \text{ct}' \oplus \text{msg} \oplus \bigoplus_{j \neq i} \text{pd}_j$, using honest partial decryptions for corrupted parties and cached responses for queried honest parties. The same proof then applies. $\square$

4 Ch-TPOKÉ

We present Ch-TPOKÉ, an (n, n) variant that chains the parties' secret isogenies, and prove its SIM-CPA security under static maximal corruption.

4.1 Overview of Ch-TPOKÉ

Ch-TPOKÉ uses the same public parameters $a, b, c, f, p, E_0, (P_0, Q_0), (R_0, S_0), (X_0, Y_0)$ as S-TPOKÉ.

For Ch-TPOKÉ, we fix a basis-derivation algorithm Basis_{5^c} with deterministic output and expected polynomial running time, following [19, Section 3] and [47, Section 4 and Remark 1]. On input a curve E , it outputs a basis of $E[5^c]$, allowing parties to agree on the same basis without communication. We write $\text{Basis}_{5^c}(E, \zeta)$ for the variant that additionally rescales the first basis point so that the Weil pairing of the two output points equals a given primitive 5^c -th root of unity ζ . Computing the rescaling factor amounts to solving a discrete logarithm in a group of order 5^c , which the Pohlig–Hellman algorithm does in time polynomial in c and $\log p$ since 5^c is smooth.

As illustrated in Fig. 4, Ch-TPOKÉ uses a chain $\phi_i : E_{i-1} \rightarrow E_i$ and propagates one degree- 3^b isogeny $\psi_0 : E_0 \rightarrow E'_0$ through $\psi_i := (\phi_i)_* \psi_{i-1}$. The public key omits the 5^c -torsion bases on $E_1, \dots, E_{n-1}$. Each party returns a restricted map, represented by its action on a deterministically chosen 5^c -torsion basis; the reconstruction algorithm composes these maps to recover (X'_n, Y'_n) . The message can be recovered only after composing all n maps, so extending Ch-TPOKÉ to (t, n) with $t < n$ is not immediate.

The construction also applies to PIKE [14] and LIT-SiGamal [46], which expose decryption maps that can be composed in the same way. It does not apply directly to INKE [36], whose shared secret is an intermediate curve in a two-dimensional isogeny representation rather than torsion points or their pairings.

4.2 Ch-TPOKÉ Algorithms

We now specify $\text{Ch-TPOKÉ}.\{\text{KGen}, \text{Enc}, \text{ParDec}, \text{Rec}\}$. These four algorithms form an (n, n) -TPKE scheme.

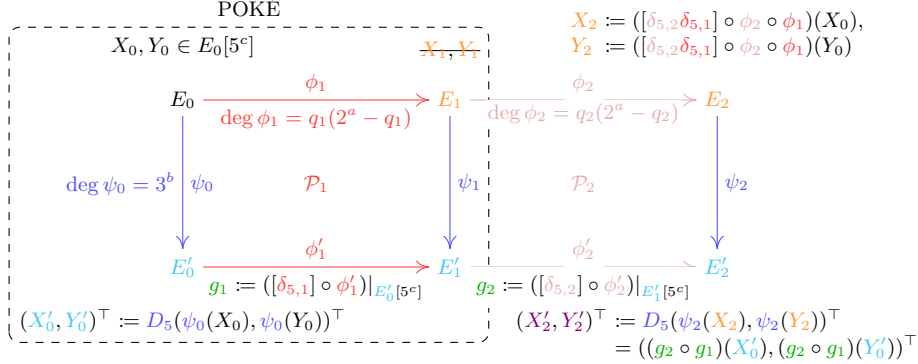


Fig. 4. A sketch of (2,2)-Ch-TPOKÉ. Public parameters and annotations are in black. Red and pink indicate $\mathcal{P}_1$ and $\mathcal{P}_2$ and their secrets, respectively. Elements in orange are part of the public key, except for the struck-through pair, which is never transmitted. Elements in blue are ephemeral secrets used during encryption, and elements in cyan are part of the ciphertext. Green denotes the restricted maps g_i represented by the partial-decryption shares, and purple indicates the shared secrets. The 2^a - and 3^b -torsion points are omitted; they behave as in POKÉ.

Key Generation Algorithm As in S-TPOKÉ, the dealer samples each q_i so that q_i and $2^a - q_i$ are prime and $q_i(2^a - q_i)$ is coprime to 2, 3, 5; it also requires the $2n$ primes $q_1, 2^a - q_1, \dots, q_n, 2^a - q_n$ to be distinct, so that the $\deg \phi_i$ are pairwise coprime. For each $i = 1, \dots, n$, it also samples the masking scalars $\alpha_{2,i}, \beta_{2,i} \in \mathbb{Z}_{2^a}^\times$, $\gamma_{3,i} \in \mathbb{Z}_{3^b}^\times$, and $\delta_{5,i} \in \mathbb{Z}_{5^c}^\times$ independently and uniformly.

The dealer next samples a secret isogeny chain $E_0 \xrightarrow{\phi_1} E_1 \xrightarrow{\phi_2} \dots \xrightarrow{\phi_n} E_n$, with $\deg \phi_i = q_i(2^a - q_i)$. Applying POKÉ key generation independently at each step would require an explicit presentation of $\text{End}(E_{i-1})$ for every i . We instead generate the chain by combining the rigorous key-generation method of POKÉ [3, Section 4.1] with the ideal-pushforward method used in SQIsign2D-West [2, Section 2.1]. The rigorous method uses the Deuring correspondence to sample a left $\text{End}(E_0)$ -ideal corresponding to an isogeny $\eta_i : E_0 \rightarrow F_i$ of degree $q_i(2^a - q_i)$. Set $\phi^{(0)} := \text{id}_{E_0}$. By the sampling condition above, $\deg \phi^{(i-1)}$ and $\deg \eta_i$ are coprime, so $\phi^{(i-1)}$ sends degree- $\deg \eta_i$ kernels bijectively, and pushing forward the uniform η_i yields a uniformly distributed ϕ_i . Given the ideals corresponding to $\phi^{(i-1)}$ and η_i , the ideal-pushforward computation yields ideals corresponding to $\phi_i := (\phi^{(i-1)})_* \eta_i : E_{i-1} \rightarrow E_i$ and $\phi^{(i)} := \phi_i \circ \phi^{(i-1)}$ [1, Section 3.1.6.1]. An applicable ideal-to-isogeny algorithm—one based on KLPT [38], Clapoti [49], or Qlapoti [9]—computes $\phi^{(i-1)}(P_0)$, $\phi^{(i-1)}(Q_0)$, $\phi^{(i)}(P_0)$, $\phi^{(i)}(Q_0)$. Since $\deg \phi^{(i-1)}$ is odd, $(\phi^{(i-1)}(P_0), \phi^{(i-1)}(Q_0))$ is a basis of $E_{i-1}[2^a]$, and $(q_i, (\phi^{(i-1)}(P_0), \phi^{(i)}(P_0)), (\phi^{(i-1)}(Q_0), \phi^{(i)}(Q_0)))$ is a Kani representation of ϕ_i . As in the rigorous key generation of POKÉ, this conversion uses (3,3)- and (5,5)-isogenies in addition to (2,2)-isogenies, since the available 2-power torsion alone is insufficient for our parameter set.

The 2^a - and 3^b -torsion bases are then pushed forward one step at a time along the chain and masked on each curve E_i : for every $i = 1, \dots, n$, the public key contains $P_i := [\alpha_{2,i}]\phi_i(P_{i-1})$, $Q_i := [\beta_{2,i}]\phi_i(Q_{i-1})$, $R_i := [\gamma_{3,i}]\phi_i(R_{i-1})$, $S_i := [\gamma_{3,i}]\phi_i(S_{i-1})$. For the 5^c -torsion, only the basis on E_n is published: writing $\delta_5 := \prod_{i=1}^n \delta_{5,i}$, the dealer computes $X_n := ([\delta_5] \circ \phi_n \circ \dots \circ \phi_1)(X_0)$ and $Y_n := ([\delta_5] \circ \phi_n \circ \dots \circ \phi_1)(Y_0)$. Thus the public key reveals the masked torsion data needed by encryption, while no 5^c -torsion basis on E_i is published for $i = 1, \dots, n-1$.

Finally, party $\mathcal{P}_i$ receives $\text{sk}_i = (q_i, \alpha_{2,i}, \beta_{2,i}, \delta_{5,i})$, and the public key is $\text{pk} := ((E_1, P_1, Q_1, R_1, S_1), \dots, (E_{n-1}, P_{n-1}, Q_{n-1}, R_{n-1}, S_{n-1}), (E_n, P_n, Q_n, R_n, S_n, X_n, Y_n))$.

Encryption Algorithm The message space of Ch-TPOKÉ is $\text{SL}_2(\mathbb{Z}_{5^c})$, and we denote a message by M_{msg} . In Ch-TPOKÉ.Enc, the sender samples $r_3 \in \mathbb{Z}_{3^b}$ and constructs $\psi_0 : E_0 \rightarrow E'_0$ with kernel $\langle R_0 + [r_3]S_0 \rangle$. For each $i = 1, \dots, n$, it constructs $\psi_i : E_i \rightarrow E'_i$ with kernel $\langle R_i + [r_3]S_i \rangle$; by the common scalar mask on R_i, S_i , this is the pushforward of ψ_{i-1} along ϕ_i .

Next, the sender samples $\omega_2 \in \mathbb{Z}_{2^a}^\times$ and sets the masked 2^a -torsion images $P'_i := [\omega_2]\psi_i(P_i)$ and $Q'_i := [\omega_2^{-1}]\psi_i(Q_i)$ for all $i = 0, \dots, n$. For each $i = 1, \dots, n$, party $\mathcal{P}_i$ uses (P'_{i-1}, Q'_{i-1}) and (P'_i, Q'_i) during partial decryption to reconstruct a Kani representation of the pushforward $\phi'_i : E'_{i-1} \rightarrow E'_i$.

For the 5^c -torsion part, the sender samples $D_5 \in \text{SL}_2(\mathbb{Z}_{5^c})$ and sets $[X'_0, Y'_0]^\top := D_5[\psi_0(X_0), \psi_0(Y_0)]^\top$, $[X'_n, Y'_n]^\top := D_5[\psi_n(X_n), \psi_n(Y_n)]^\top$. The pair (X'_0, Y'_0) is included in the ciphertext, while (X'_n, Y'_n) is used to mask the message matrix.

The sender also sets $(U, V) := \text{Basis}_{5^c}(E'_n, e_{5^c}(X_n, Y_n)^{3^b})$. Since $\deg \psi_n = 3^b$ and $D_5 \in \text{SL}_2(\mathbb{Z}_{5^c})$, we have $e_{5^c}(X_n, Y_n)^{3^b} = e_{5^c}(X'_n, Y'_n)$. Hence the change-of-basis matrix from (X'_n, Y'_n) to (U, V) has determinant 1, so there is a unique matrix $M \in \text{SL}_2(\mathbb{Z}_{5^c})$ satisfying $[U, V]^\top = M[X'_n, Y'_n]^\top$; its entries are the coordinates of U and V with respect to (X'_n, Y'_n) , computed by the Pohlig–Hellman algorithm over $\mathbb{Z}_{5^c}$. The sender sets $C := M \cdot M_{\text{msg}}$. The ciphertext is $\text{ct} := ((E'_0, P'_0, Q'_0, X'_0, Y'_0), (E'_1, P'_1, Q'_1), \dots, (E'_n, P'_n, Q'_n), C)$.

Partial Decryption Algorithm The algorithm Ch-TPOKÉ.ParDec is run locally by each party. Party $\mathcal{P}_i$ does not attempt to recover (X'_n, Y'_n) on $E'_n[5^c]$. Instead, given $\text{sk}_i = (q_i, \alpha_{2,i}, \beta_{2,i}, \delta_{5,i})$ and the ciphertext, it un.masks P'_i, Q'_i to obtain the exact images $([\alpha_{2,i}^{-1}]P'_i, [\beta_{2,i}^{-1}]Q'_i)$, so that $(q_i, (P'_{i-1}, [\alpha_{2,i}^{-1}]P'_i), (Q'_{i-1}, [\beta_{2,i}^{-1}]Q'_i))$ is a Kani representation of $\phi'_i : E'_{i-1} \rightarrow E'_i$. The party sets $(B_{i-1,1}, B_{i-1,2}) := \text{Basis}_{5^c}(E'_{i-1})$, computes $Z_{i,X} := [\delta_{5,i}]\phi'_i(B_{i-1,1})$ and $Z_{i,Y} := [\delta_{5,i}]\phi'_i(B_{i-1,2})$, and outputs $\text{pd}_i := (Z_{i,X}, Z_{i,Y})$. Thus, pd_i represents the restricted map $g_i := ([\delta_{5,i}] \circ \phi'_i)|_{E'_{i-1}[5^c]}$.

Reconstruction Algorithm The reconstruction algorithm outputs $\perp$ unless all n partial decryptions are supplied. It then checks every $\text{pd}_i = (Z_{i,X}, Z_{i,Y})$: the two points must lie in $E'_i[5^c]$ and form a basis; otherwise, it outputs $\perp$. For each

$i = 1, \dots, n$, let $B_{i-1,1}$ and $B_{i-1,2}$ be the two points output by $\text{Basis}_{5^c}(E'_{i-1})$. Then pd_i determines g_i , the unique $\mathbb{Z}_{5^c}$ -linear isomorphism $E'_{i-1}[5^c] \rightarrow E'_i[5^c]$ such that $g_i(B_{i-1,1}) = Z_{i,X}$ and $g_i(B_{i-1,2}) = Z_{i,Y}$. The reconstruction algorithm sets $(X'_n, Y'_n) := ((g_n \circ \dots \circ g_1)(X'_0), (g_n \circ \dots \circ g_1)(Y'_0))$.

From this point on, reconstruction mirrors the final step of encryption. Using X_n, Y_n from pk , it sets $(U, V) := \text{Basis}_{5^c}(E'_n, e_{5^c}(X_n, Y_n)^{3^b})$. If there is no matrix $M \in \text{SL}_2(\mathbb{Z}_{5^c})$ satisfying $[U, V]^\top = M[X'_n, Y'_n]^\top$, it outputs $\perp$; otherwise, it computes the unique such matrix and outputs $M_{\text{msg}} := M^{-1} \cdot C$.

Correctness. For honestly generated keys and ciphertexts, the partial decryption pd_i gives the images of $\text{Basis}_{5^c}(E'_{i-1})$ under g_i , so the reconstruction algorithm recovers g_i on $E'_{i-1}[5^c]$. Hence $g_n \circ \dots \circ g_1$ is the restriction of $[\delta_5] \circ \phi'_n \circ \dots \circ \phi'_1$, where $\delta_5 := \prod_{i=1}^n \delta_{5,i}$. The ciphertext basis satisfies $[X'_0, Y'_0]^\top = D_5[\psi_0(X_0), \psi_0(Y_0)]^\top$, and scalar multiplication by δ_5 commutes with the matrix D_5 on 5^c -torsion. Together with $\phi'_i \circ \psi_{i-1} = \psi_i \circ \phi_i$, this gives the same pair (X'_n, Y'_n) as the sender computes. Thus, encryption and reconstruction obtain the same basis $(U, V) = \text{Basis}_{5^c}(E'_n, e_{5^c}(X_n, Y_n)^{3^b})$, the same matrix M with $[U, V]^\top = M[X'_n, Y'_n]^\top$, and the reconstruction algorithm outputs $M^{-1}C = M_{\text{msg}}$.

4.3 SIM-CPA Security of Ch-TPOKÉ under Maximal Corruption

We prove the SIM-CPA security of Ch-TPOKÉ under static maximal corruption using the (j, n) -Ch-D-POKE problem, where j is the index of the unique uncorrupted party. Hardness is required for every j because the exposed boundary data and the other chain edges make the positions distributionally asymmetric. The special case $(1, 1)$ coincides with D-POKE conditioned on q and $2^a - q$ being prime.

Problem 2 ((j, n)-Ch-D-POKE). For positive integers $j \leq n$, (j, n) -Ch-D-POKE is defined as follows.

- Sample $q_1, \dots, q_n$ uniformly from $[1, 2^a - 1]$, conditioned on $q_1, 2^a - q_1, \dots, q_n, 2^a - q_n$ being distinct primes other than 2, 3, 5. Sample uniform isogenies $\phi_i : E_{i-1} \rightarrow E_i$ of degree $q_i(2^a - q_i)$ for $1 \leq i \leq n$.
- Let $\phi^{(j)} := \phi_j \circ \dots \circ \phi_1$. Choose uniformly at random $\alpha_{2,j}, \beta_{2,j} \in \mathbb{Z}_{2^a}^\times, \gamma_{3,j} \in \mathbb{Z}_{3^b}^\times, \delta_{5,j} \in \mathbb{Z}_{5^c}^\times$, and define $P_j := [\alpha_{2,j}]^{\phi^{(j)}}(P_0), Q_j := [\beta_{2,j}]^{\phi^{(j)}}(Q_0), R_j := [\gamma_{3,j}]^{\phi^{(j)}}(R_0), S_j := [\gamma_{3,j}]^{\phi^{(j)}}(S_0), X_j := [\delta_{5,j}]^{\phi^{(j)}}(X_0), Y_j := [\delta_{5,j}]^{\phi^{(j)}}(Y_0)$.
- Sample $r_3 \xleftarrow{\$} \mathbb{Z}_{3^b}$ and let $\psi_0 : E_0 \rightarrow E'_0$ have kernel $\langle R_0 + [r_3]S_0 \rangle$. For $1 \leq i \leq n$, define $\psi_i := (\phi_i)_* \psi_{i-1}$ and $\phi'_i := (\psi_{i-1})_* \phi_i : E'_{i-1} \rightarrow E'_i$.
- Choose uniformly at random $\omega_2 \in \mathbb{Z}_{2^a}^\times$ and $D_5 \in \text{SL}_2(\mathbb{Z}_{5^c})$ and, for each $i \in \{0, j\}$, define $P'_i := [\omega_2] \psi_i(P_i), Q'_i := [\omega_2^{-1}] \psi_i(Q_i), [X'_i, Y'_i]^\top := D_5[\psi_i(X_i), \psi_i(Y_i)]^\top$.
- Let $\tau \xleftarrow{\$} \{0, 1\}$ and let (X^1, Y^1) be a uniformly random ordered basis of $E'_j[5^c]$ with prescribed pairing $e_{5^c}(X^1, Y^1) = e_{5^c}(X'_j, Y'_j) = e_{5^c}(X_j, Y_j)^{3^b}$. Set $(X^0, Y^0) := (X'_j, Y'_j)$.

Given as input $E_0, \dots, E_n, E'_0, \dots, E'_n, ((\phi_i, q_i))_{i \in \{1, \dots, n\} \setminus \{j\}}, (\phi'_i)_{i \in \{1, \dots, n\} \setminus \{j\}}, (P_0, Q_0, R_0, S_0, X_0, Y_0), (P_j, Q_j, R_j, S_j, X_j, Y_j), (P'_0, Q'_0, X'_0, Y'_0), (P'_j, Q'_j)$, and (X^τ, Y^τ) , compute τ .

Theorem 3. *Suppose that the (j, n) -Ch-D-POKE problem is hard for every $j \in \{1, \dots, n\}$. Then Ch-TPOKÉ is SIM-CPA secure under static maximal corruption.*

Proof. In the maximal-corruption experiment, the adversary outputs a corrupt set $\mathcal{C}$ of size $n - 1$; write j for the unique index outside $\mathcal{C}$. As in the proof of Theorem 2, the stateful simulator initializes honestly and answers EncO and the associated partial-decryption queries honestly. For ChalO, it generates the curve and torsion components exactly as in encryption and samples $C \xleftarrow{\$} \text{SL}_2(\mathbb{Z}_{5^c})$ independently.

The simulator computes corrupted-party partial decryptions honestly. When the uncorrupted party j is queried on a challenge, $|P[\text{ct}] \cup \mathcal{C}| = n$, so the simulator receives M_{msg} . It computes (U, V) as in encryption and sets $M^* := CM_{\text{msg}}^{-1}$. Let $H^* : E'_0[5^c] \rightarrow E'_n[5^c]$ be the unique isomorphism satisfying $[U, V]^\top = M^*[H^*(X'_0), H^*(Y'_0)]^\top$. From the corrupted keys it recovers every g_i for $i \neq j$ and defines $g_j^* := (g_n \circ \dots \circ g_{j+1})^{-1} \circ H^* \circ (g_{j-1} \circ \dots \circ g_1)^{-1}$, where empty compositions are identity maps. It returns the images of $\text{Basis}_{5^c}(E'_{j-1})$ under g_j^* and caches this partial decryption. Since the reconstructed matrix is M^* , reconstruction returns $(M^*)^{-1}C = M_{\text{msg}}$. The simulator maintains g_j^* separately for each challenge ciphertext, so, as in the proof of Theorem 2, switching the challenge ciphertexts one at a time reduces the multi-challenge case to the single-challenge case.

It remains to prove single-challenge security. By averaging, fix an index j^* such that the product of $\Pr[\mathcal{C} = \{1, \dots, n\} \setminus \{j^*\}]$ and the single-challenge advantage conditioned on this event is non-negligible. Let $\mathcal{B}$ be given a (j^*, n) -Ch-D-POKE instance. It runs $\mathcal{A}$ and aborts unless $\mathcal{A}$ outputs $\mathcal{C} = \{1, \dots, n\} \setminus \{j^*\}$. As in the proof of Theorem 2, $\mathcal{B}$ constructs the Ch-TPOKÉ public key and the corrupted parties' secret keys from the instance, sampling their masking scalars. It treats the data at j^* as the output of the first j^* key-generation steps, so it does not apply another prefix mask to (X^τ, Y^τ) . It then propagates the points through the known suffix isogenies. This gives the honest key-generation distribution. It gives the keys to $\mathcal{A}$ and answers EncO queries honestly. For a partial-decryption query on an EncO ciphertext, the stored encryption randomness determines the isomorphism $H : E'_0[5^c] \rightarrow E'_n[5^c]$ satisfying $H(X'_0) = X'_n$ and $H(Y'_0) = Y'_n$; together with the known g_i for $i \neq j^*$, this also determines the missing partial decryption. $\mathcal{B}$ answers challenge partial-decryption queries for corrupted parties using their secret keys. On the unique ChalO(msg) query, $\mathcal{B}$ forms the ciphertext as in Ch-TPOKÉ.Enc. Using the known maps g_i for $i > j^*$, it sets $(\tilde{X}_n^\tau, \tilde{Y}_n^\tau) := ((g_n \circ \dots \circ g_{j^*+1})(X^\tau), (g_n \circ \dots \circ g_{j^*+1})(Y^\tau))$. With (U, V) computed as in encryption, it takes the unique $M_\tau \in \text{SL}_2(\mathbb{Z}_{5^c})$ satisfying $[U, V]^\top = M_\tau[\tilde{X}_n^\tau, \tilde{Y}_n^\tau]^\top$, sets $C := M_\tau \cdot M_{\text{msg}}$, and returns the ciphertext.

The two cases in the Ch-D-POKE instance match the two single-challenge experiments. For the real basis, the ciphertext is an honest Ch-TPOKÉ encryption. For an independent basis with the prescribed Weil pairing, the bijective suffix maps preserve uniformity within the corresponding pairing class; hence M_τ , and therefore C , is uniform in $\text{SL}_2(\mathbb{Z}_{5^c})$, as in the simulated experiment. Until $j^\star$ is queried on the challenge ciphertext, every partial decryption returned for that ciphertext is computed from a corrupted secret key. If $j^\star$ is queried, $\mathcal{B}$ lets $H_\tau : E'_0[5^c] \rightarrow E'_n[5^c]$ be the unique isomorphism sending (X'_0, Y'_0) to $(\tilde{X}_n^\tau, \tilde{Y}_n^\tau)$, and returns the images of $\text{Basis}_{5^c}(E'_{j^\star-1})$ under $g_{j^\star}^\tau := (g_n \circ \dots \circ g_{j^\star+1})^{-1} \circ H_\tau \circ (g_{j^\star-1} \circ \dots \circ g_1)^{-1}$. For $\tau = 0$, we have $H_0 = g_n \circ \dots \circ g_1$, so $g_{j^\star}^0$ is the honest partial-decryption map. For $\tau = 1$, this query makes $|P[\text{ct}] \cup \mathcal{C}| = n$, so the ideal experiment supplies M_{msg} to the simulated partial-decryption procedure. Since $M_\tau = CM_{\text{msg}}^{-1}$ and $[U, V]^\top = M_\tau[H_\tau(X'_0), H_\tau(Y'_0)]^\top$, the uniqueness of $H^\star$ gives $H_\tau = H^\star$. Thus $g_{j^\star}^\tau$ is exactly the programmed partial-decryption map defined above. Consequently, any non-negligible single-challenge distinguisher gives a non-negligible solver for the $(j^\star, n)$ -Ch-D-POKE problem, contradicting the assumed hardness for $j^\star$. $\square$

5 Concrete Attacks against TPOKÉ

The attacks analyzed in the POKÉ paper also apply to TPOKÉ [3, Sections 5 and 6]; here, we focus on attacks identified since the publication of POKÉ and on the security effects of features specific to TPOKÉ.

Generic isogeny attack. Under a heuristic assumption, Wesolowski gives a Las Vegas algorithm for the general supersingular isogeny problem, with expected time and memory both $p^{1/3+o(1)}$ [60, Corollary 1.2 and Section 1.1]. We therefore require $p \geq 2^{3\lambda}$, so that the attack's estimated cost is at least 2^λ .

Exhaustive search for q_i . Unlike POKÉ, TPOKÉ samples each q_i conditioned on both q_i and $2^a - q_i$ being prime. Of the attacks analyzed for POKÉ, this change affects only the exhaustive search for q_i [3, Sections 5 and 6]. As shown in the description of S-TPOKÉ key generation, the conditioning reduces the number of candidates from $\Theta(2^a)$ to $\Theta(2^a/a^2)$. For each candidate, the attack computes a four-dimensional representation of ϕ_i from the published 3^b -torsion data [3, Sections 6.1 and 7.1]. This step costs $\tilde{\Theta}(a^2)$ bit operations [22, Section 4.1 and Appendix E]. The total estimated cost therefore remains $\tilde{\Theta}(2^a)$, so $a \approx \lambda$ still provides λ -bit security against this attack.

Weak-key attack. Fix a divisor $D' \mid 3^b 5^c$ with $D' \geq 2^{a-1}$ and a square-free integer r . From the public parameters and pk_i , Moriya's weak-key attack [45, Theorem 11 and Algorithm 2] recovers ϕ_i in polynomial time if $(D')^2 - \deg \phi_i = ru^2$ for some $u \in \mathbb{Z}$. We assume that Moriya's heuristic remains valid under TPOKÉ's prime-conditioned sampling of q_i . For one party, testing M values of r costs $\tilde{O}(M)$ and succeeds with probability $\varepsilon_M = O(\sqrt{M}/2^a)$. For a corrupt set $\mathcal{C}$,

recovering any of the $n - |\mathcal{C}|$ honest keys has probability at most $(n - |\mathcal{C}|)\varepsilon_M = O((n - |\mathcal{C}|)\sqrt{M}/2^a)$ by a union bound. Breaking S-TPOKÉ instead requires recovering $t - |\mathcal{C}|$ such keys. Even allowing correlations among the parties' weak-key events, Markov's inequality bounds the probability that at least $t - |\mathcal{C}|$ honest keys are recovered by $(n - |\mathcal{C}|)\varepsilon_M/(t - |\mathcal{C}|) = O((n - t + 1)\sqrt{M}/2^a)$. For polynomial n , the asymptotic POKÉ-2D parameter choice remains unchanged [45, Heuristic 1 and Theorem 19].

Threshold-specific attacks. From a single ciphertext, each corrupted party $\mathcal{P}_i$ can reconstruct the pair (ϕ_i, ϕ'_i) , where $\phi'_i = (\psi_0)_*\phi_i$ in S-TPOKÉ and $\phi'_i = (\psi_{i-1})_*\phi_i$ in Ch-TPOKÉ. Thus, corrupting several parties reveals several such pairs. Unless both q_i and $2^a - q_i$ are prime, these pairs enable the attack described for Threshold PRISM, which uses Kani representations to recover ψ_0 or ψ_{i-1} from pushforward information. Threshold PRISM expects that the minimal k such that a nonzero q_i -torsion point is $\mathbb{F}_{p^k}$ -rational satisfies $k \approx q_i$ [37]. Requiring both q_i and $2^a - q_i$ to be prime therefore places the attack out of reach.

The attack of Maino and Petit [41, Problem 3 and Theorem 9] uses pushforwards of attacker-chosen isogenies to construct a suitable endomorphism and recover the secret isogeny underlying the pushforwards— ψ_0 in S-TPOKÉ or, locally, ψ_{i-1} in Ch-TPOKÉ. Threshold PRISM allows the attacker to choose the isogenies and consequently admits polynomial-time key recovery [41, Section 6.2]. In S-TPOKÉ and Ch-TPOKÉ, however, the dealer samples the ϕ_i at random, so an adversary cannot choose the isogenies whose pushforwards are revealed. Applying the Maino–Petit method to these random samples has an estimated quantum cost of $2^{14\lambda/9 - O(\log \lambda)}$. This cost exceeds 2^λ . The estimate is heuristic: their method searches for an auxiliary isogeny of degree about the rank-four scale $s_4 = (3^b)^{3/2}\sqrt{p} \approx 2^{14\lambda/3}$ of the Hom lattice. Quantum claw finding costs $s_4^{1/3}$ queries, as in the SIDH analysis [26, Section 5.1]; polynomially many combinations of the ϕ_i lower that degree only polynomially. See Appendix A for details. For S-TPOKÉ, this also shows that the naive distributed key generation protocol in which each party generates its own ϕ_i is not malicious-secure, since a malicious party could choose ϕ_i .

Analyzing other attacks that exploit threshold-specific data with no analogue in a single POKÉ instance remains open.

Remark 1. The dealer may instead sample E_0 at random and include a defining equation for E_0 in the public key. It can obtain each ϕ_i by pushing an isogeny from a curve E with known $\text{End}(E)$ along a secret $\Phi : E \rightarrow E_0$. This setup hides the trapdoor description of $\text{End}(E_0)$ and therefore removes the free public-endomorphism-ring input required for direct application of attacks such as the Maino–Petit attack [41, Theorem 9]. We do not use this variant because the public parameters no longer determine E_0 , and hence every public key must additionally contain a defining equation for E_0 .

Adaptive attack. Moriya's malformed-torsion attack applies to a TPOKÉ scheme when decryption exposes a checking oracle [44, Section 4.2]. It does not appear

Table 1. Communication costs in direct (n, n) -TPKE at security parameter λ .

	Trivial	S-TPOKÉ	Ch-TPOKÉ
pk size	$20n\lambda$	$20n\lambda$	$20n\lambda$
pk _{cmp} size	$(50/3)n\lambda$	$(50/3)n\lambda$	$((47n + 3)/3)\lambda$
ct size	$(40n + 1)\lambda$	$(20n + 21)\lambda$	$(20n + 21)\lambda$
ct _{cmp} size	$((58n + 3)/3)\lambda$	$((29n + 32)/3)\lambda$	$((29n + 32)/3)\lambda$
Total pd size	$n\lambda$	$n\lambda$	$(40/3)n\lambda$
Total pd _{cmp} size	$n\lambda$	$n\lambda$	$(4/3)n\lambda$

to apply to POKÉ4D, the four-dimensional variant of POKÉ [3, Section 4.4]. Both S-TPOKÉ and Ch-TPOKÉ can also be constructed from POKÉ4D, in which case Moriya’s revised weak-key-resistant parameters are required [45, Section 5.2]. At the security-definition level, the transformation of Section 2.4 would rule out this chosen-ciphertext leakage. Efficient instantiation of its generic primitives for our schemes remains open.

6 Parameter Selection and Comparison

The attacks in Section 5 do not suggest larger parameters, so we use the POKÉ sizes $a \approx \lambda$, $3^b \approx 2^{2\lambda}$, and $5^c \approx 2^{\lambda/3}$ [3, Section 7.1]. These sizes give $p \approx 2^{(10/3)\lambda} > 2^{3\lambda}$ asymptotically, as required by the generic attack.

Following POKÉ [3, Section 7.1], we transmit coprime-order points as their sums, represent a torsion basis with known Weil pairing by three coefficients, and omit a deterministic 5^c -torsion basis.

Ch-TPOKÉ omits the 5^c -torsion bases on E_i for $i < n$, deterministically derives (U, V) from E'_n and $e_{5^c}(X_n, Y_n)^{3^b}$, and encodes each partial decryption by four coefficients. Since $C \in \text{SL}_2(\mathbb{Z}_{5^c})$, three entries suffice; the table rounds $|\text{SL}_2(\mathbb{Z}_{5^c})| < 2^\lambda$ up and counts C as λ bits.

At $t = n$, Table 1 compares the three constructions. S-TPOKÉ has the same public key as the trivial construction, while both proposals use ciphertext torsion data on $n + 1$ rather than $2n$ curves. As n grows, either ciphertext approaches half the trivial construction’s size; Ch-TPOKÉ’s compressed public key is about 6% smaller than S-TPOKÉ’s.

For $t < n$, the trivial construction and S-TPOKÉ use n Shamir shares over $\mathbb{F}_{2^e}$ instead of the single ℓ -bit payload ct' , adding $(n - 1)\ell$ ciphertext bits.

Main operation counts. Compared with the trivial construction, both proposed schemes reduce the number of degree- 3^b isogenies in encryption from $2n$ to $n + 1$. Ch-TPOKÉ additionally requires five discrete logarithms in $\mathbb{Z}_{5^c}$ during encryption; five discrete logarithms, $n + 1$ matrix multiplications, and one inversion during reconstruction; and, for compressed partial-decryption shares, four scalar discrete logarithms per party to express the two output points in the deterministic torsion basis, with no corresponding cost for uncompressed shares.

7 Conclusion

We proposed S-TPOKÉ for general t and Ch-TPOKÉ for $t = n$. Both asymptotically halve the trivial construction's ciphertext size and encryption-isogeny count. For Ch-TPOKÉ, we prove only SIM-CPA security under static maximal corruption. Open problems are further threshold-specific attack analysis, sublinear-size public keys and ciphertexts, Ch-TPOKÉ for $t < n$, dealerless distributed key generation, and an efficient SIM-CPA-to-SIM-CCA instantiation.

A Application of the Maino–Petit Algorithm to TPOKÉ

Maino and Petit show how pushforwards of attacker-chosen isogenies can be used to recover an unknown isogeny. We first recall the inputs and recovery strategy of their attack and then examine whether corrupted TPOKÉ parties obtain the required data. Throughout this section, the attacker is assumed to be quantum, and every assumption we make is chosen in the attacker's favor.

A.1 Overview of the Maino–Petit Attack

Let $\psi : E \rightarrow E'$ be an unknown cyclic isogeny of degree d . For an integer N coprime to d , the pushforward oracle accepts an attacker-chosen N -isogeny $\sigma : E \rightarrow F$ and returns $\sigma' := \psi_*\sigma : E' \rightarrow F'$; we denote this oracle by $\mathcal{O}_{\psi, N}^{\text{PUSH}}$. The computational isogeny pushforward problem asks for a representation of an isogeny $\psi : E \rightarrow E'$ whose restriction to $E[N]$ agrees with that of ψ up to a nonzero scalar [41, Problem 3].

We now assume that $d = \deg(\psi) = \prod_{j=1}^r \ell_j^{a_j}$ is smooth. Given $\theta \in \text{End}(E)$ such that $\theta|_{E[d]} = [\mu]|_{E[d]}$ for some $\mu \in \mathbb{Z}/d\mathbb{Z}$, define the diagonalizable exponent as follows.

Definition 4 ([41, Definition 8]). *Let θ and d be as above. Let d' be the largest integer such that $\alpha_0 := \frac{2\theta - \text{tr}(\theta)}{d \cdot d'}$ is in $\text{End}(E)$, every prime divisor of d' divides $2d$, and $\alpha_0|_{E[u]} \neq [0]|_{E[u]}$ for every $u \mid d$. Let $\alpha := \psi_*([d]\alpha_0)$. The diagonalizable exponent is the largest integer m such that there exist distinct $i_1, \dots, i_m \in \{1, \dots, r\}$ with $\alpha|_{E'[\ell_{i_j}]} = [0]|_{E'[\ell_{i_j}]}$ for every $j = 1, \dots, m$.*

Theorem 4 ([41, Theorem 9]). *Let $N \in O(\text{poly}(p))$ and $d = \prod_{j=1}^r \ell_j^{a_j}$ be two coprime positive integers, and suppose $\ell_j \in O(\text{poly}(\log(p)))$ and $r \in O(\log(p))$. Let E be a supersingular elliptic curve with known endomorphism ring, and let $\theta \in \text{End}(E)$ satisfy $\theta|_{E[d]} = [\mu]|_{E[d]}$ for some $\mu \in \mathbb{Z}/d\mathbb{Z}$, $\theta \in \text{End}(E) \setminus \mathbb{Z}$, and $\deg(\theta) = N^2 e$ for some smooth $e \in O(\text{poly}(p))$. Consider a cyclic d -isogeny $\psi : E \rightarrow E'$. Then Algorithm 1 of Maino and Petit solves the computational isogeny pushforward problem using two calls to $\mathcal{O}_{\psi, N}^{\text{PUSH}}$ with time complexity*

$$O(\max\{2^m, \sqrt{e}\} \cdot \text{poly}(\log(p))),$$

where m is the diagonalizable exponent from Definition 4.

The algorithm first computes $\theta' := \psi_*\theta \in \text{End}(E')$ from two pushforward queries. It then uses (θ, θ') to recover the prime-power components of $\ker \psi$ and combines them to recover ψ .

The formal oracle restricts its inputs to one fixed degree N . The recovery proof, however, only uses the pushforwards needed to compute θ' . If these pushforwards are available directly, their degrees need not coincide.

A.2 Application to TPOKÉ

For S-TPOKÉ, let $\mathcal{C}$ denote the set of corrupted parties. For one challenge ciphertext, the unknown isogeny targeted by the Maino–Petit algorithm is $\psi_0 : E_0 \rightarrow E'_0$, whose degree $d := \deg(\psi_0) = 3^b$ is public. For every $i \in \mathcal{C}$, the adversary uses the secret key and the ciphertext to reconstruct a representation of $\phi_i : E_0 \rightarrow E_i$ and its pushforward $\phi'_i = (\psi_0)_*\phi_i : E'_0 \rightarrow E'_i$; see Section 3.2. No such pair is revealed for an honest party. Thus the adversary obtains exactly $|\mathcal{C}|$ pushforward pairs $(\phi_i, \phi'_i)_{i \in \mathcal{C}}$ and uses them to attempt to recover ψ_0 . Because the sender samples a fresh ψ_0 for each ciphertext, pairs from different ciphertexts cannot be combined in this algorithm.

In Ch-TPOKÉ, a corrupted party reconstructs $\phi_i : E_{i-1} \rightarrow E_i$ and $\phi'_i = (\psi_{i-1})_*\phi_i$; see Section 4.2. If party $i-1$ is also corrupted, then, because key generation makes the degrees of the ϕ_j pairwise coprime, the pullbacks $\phi_{i-1}^*\phi_i = (\widehat{\phi_{i-1}})_*\phi_i$ and $(\phi'_{i-1})^*\phi'_i = (\widehat{\phi'_{i-1}})_*\phi'_i$ are computable in polynomial time [53, Propositions 6.4 and 6.15]. If a party is uncorrupted, the attacker cannot continue this sequence of pullbacks past that party, since computing them requires information about that party's ϕ_j . As the target here is ψ_0 alone, we grant the attacker all pullbacks to E_0 nonetheless. This yields $\tilde{\phi}_i$ and $(\psi_0)_*\tilde{\phi}_i$, both of degree $\deg(\phi_i)$, and reduces to the S-TPOKÉ case analyzed below.

A.3 Endomorphism Construction

For every corrupted index i , the degree of ϕ_i is $q_i(2^a - q_i)$. Under the intended parameter generation, these degrees are squarefree and pairwise coprime. The adversary can therefore compute pushforwards among the ϕ_i in polynomial time [53, Proposition 6.15]. For example, define $\Phi_{i,j} := ((\phi_i)_*\phi_j) \circ \phi_i$ and $\Phi'_{i,j} := ((\phi'_i)_*\phi'_j) \circ \phi'_i$. Then $\Phi'_{i,j} = (\psi_0)_*\Phi_{i,j}$ and $\deg(\Phi_{i,j}) = \deg(\Phi'_{i,j}) = \deg(\phi_i)\deg(\phi_j)$. Iterating this construction over corrupted indices I gives isogenies Φ_I and $\Phi'_I = (\psi_0)_*\Phi_I$ of degree $D_I := \deg(\Phi_I) = \prod_{i \in I} \deg(\phi_i)$.

Let $\Phi := \Phi_I : E_0 \rightarrow E_I$ and $\Phi' := (\psi_0)_*\Phi : E'_0 \rightarrow E'_I$. Write $D := D_I$. Given an efficient representation of Φ_I and a public description of $\text{End}(E_0)$, a quantum attacker can compute $\text{End}(E_I)$ in polynomial time [16, Theorem 1.4].

Using Φ_I , the attacker embeds $\text{End}(E_I)$ into $\text{End}^0(E_0)$ by $\beta \mapsto D^{-1}\widehat{\Phi}_I \circ \beta \circ \Phi_I$. It then represents $\text{Hom}(E_I, E_0)$ as a connecting ideal and computes a basis by standard quaternion-lattice arithmetic. To construct an endomorphism, consider an isogeny $\eta : E_I \rightarrow E_0$ satisfying

$$(\eta \circ \Phi)|_{E_0[d]} = [\mu], \quad \eta \circ \Phi \notin \mathbb{Z}, \quad (1)$$

for some $\mu \in \mathbb{Z}/d\mathbb{Z}$. Then $\theta = \eta \circ \Phi$ satisfies the scalar-action and non-scalar conditions of Theorem 4 and has degree $\deg(\theta) = D \deg(\eta)$. The dual $\widehat{\Phi}$ satisfies the first condition because $\widehat{\Phi} \circ \Phi = [D]$, but it does not satisfy the second condition and must be excluded.

The attacker then needs an isogeny $\eta' : E'_I \rightarrow E'_0$ satisfying $\eta' \circ \Phi' = (\psi_0)_*(\eta \circ \Phi)$. Thus $\theta' := \eta' \circ \Phi' = (\psi_0)_*\theta$, which can be used by the recovery algorithm in Theorem 4. It remains to find η' , whose degree equals $\deg(\eta)$. The following lattice calculation estimates $\deg(\eta)$. We assume that $\deg(\eta)$ is smooth. The standard claw search for an isogeny of known smooth degree e costs $O(\sqrt{e})$ classical and $O(e^{1/3})$ quantum queries [26, Section 5.1]. The estimates below use the quantum cost $O(\deg(\eta)^{1/3})$ for finding η' .

Finding η is a constrained shortest-vector problem. Equip $L = \text{Hom}(E_I, E_0)$ with the degree form. This is a rank-four lattice of covolume $p/4$ in the normalization used here. Reduction modulo d identifies L/dL with the homomorphisms between the d -torsion modules. Since d is coprime to D , the restriction of Φ to $E_0[d]$ is an isomorphism. Composition with this restriction therefore defines an isomorphism

$$\rho_\Phi : L/dL \longrightarrow \text{End}(E_0[d]), \quad \eta + dL \longmapsto (\eta \circ \Phi)|_{E_0[d]}.$$

The first condition in Equation (1) defines the sublattice $L' \subset L$ whose reduction modulo d is $\rho_\Phi^{-1}((\mathbb{Z}/d\mathbb{Z})I_2)$. After choosing a basis of $E_0[d]$, the codomain of ρ_Φ is $M_2(\mathbb{Z}/d\mathbb{Z})$, which has d^4 elements, whereas its scalar submodule $(\mathbb{Z}/d\mathbb{Z})I_2$ has d elements. Hence $[L : L'] = d^3$. These rank and covolume statements follow from the level-structure Hom lattices of Borin et al. [10, Corollary 10 and Section 3.4].

The class of $\widehat{\Phi}$ maps to the invertible scalar $[D]$ on $E_0[d]$. It therefore generates all scalar classes modulo d , which gives $L' = \mathbb{Z}\widehat{\Phi} + dL$ and $\text{covol}(L') = d^3 p/4$. The four-dimensional Minkowski estimate gives a shortest-vector degree of order $\text{covol}(L')^{1/2}$ [10, Corollary 10]. Suppressing absolute constants, we therefore write the rank-four scale as $s_4 := d^{3/2} \sqrt{p}$. When $D < s_4$, $\widehat{\Phi}$ is shorter than this estimate, but $\widehat{\Phi} \circ \Phi = [D]$. Since D is squarefree, there do not exist $\xi \in L$ and an integer $\kappa > 1$ such that $\widehat{\Phi} = [\kappa] \circ \xi$: such an equality would give $D = \kappa^2 \deg(\xi)$. Thus the coordinates of $\widehat{\Phi}$ in a $\mathbb{Z}$ -basis of L have greatest common divisor 1, so elementary unimodular operations extend $\widehat{\Phi}$ to a $\mathbb{Z}$ -basis of L . Because L' has fixed rank four, lattice enumeration finds a minimum-degree element of $L' \setminus \mathbb{Z}\widehat{\Phi}$ in polynomial time in the input length. Thus finding η is efficient once L is known; the calculation below estimates its degree, which determines the cost of finding η' . Let π denote orthogonal projection onto $\widehat{\Phi}^\perp$. The volume of the resulting fundamental parallelepiped is $\|\widehat{\Phi}\|$ times the volume of its projected base. Since $\|\widehat{\Phi}\| = \sqrt{D}$, this gives $\text{covol}(\pi(L)) = (p/4)/\sqrt{D}$. Moreover, $L' = \mathbb{Z}\widehat{\Phi} + dL$ implies $\pi(L') = d\pi(L)$, whose covolume is $d^3(p/4)/\sqrt{D}$. The resulting rank-three shortest-vector degree scale is therefore $s_3(D) := d^2(p/4)^{2/3} D^{-1/3}$. In particular, $s_3(D)/s_4 = 4^{-2/3}(s_4/D)^{1/3}$, so $D \ll s_4$ makes the estimated degree of η asymptotically larger than s_4 . When $D \gg s_4$, $\widehat{\Phi}$ is no longer a short vector of L' , and excluding it does not change the rank-four estimate s_4 .

A.4 Heuristic Security Estimate

The estimates below are heuristic: they estimate the cost of the known Maino–Petit strategy, not of all possible attacks.

Even after constructing Φ_I and Φ'_I , a generic search for η' still has an estimated work factor greater than 2^λ . The POKÉ parameter choices satisfy

$$a \approx \lambda, \quad 3^b \approx 2^{2\lambda}, \quad 5^c \approx 2^{\lambda/3}, \quad p \approx 2^{10\lambda/3},$$

up to lower-order factors [3, Section 7.1]. A typical value of $\deg(\phi_i)$ is approximately $2^{2\lambda}$. For $|I| = r$, therefore, $D_I = \prod_{i \in I} \deg(\phi_i) \approx 2^{2r\lambda}$. Since $\log_2 s_4 \approx 14\lambda/3$, the equality $D_I = s_4$ occurs at $r \approx 7/3$. Hence $D_I \ll s_4$ for $r = 1, 2$, and substitution gives

$$\log_2 s_3(D_I) \approx \frac{56 - 6r}{9} \lambda = \begin{cases} 50\lambda/9, & r = 1, \\ 44\lambda/9, & r = 2. \end{cases}$$

In this case, quantum claw finding costs approximately

$$s_3(D_I)^{1/3} \approx \begin{cases} 2^{50\lambda/27}, & r = 1, \\ 2^{44\lambda/27}, & r = 2. \end{cases}$$

Both costs exceed $s_4^{1/3} \approx 2^{14\lambda/9}$, consistent with $s_3(D_I) > s_4$ in this case. Once θ' is available, the Maino–Petit recovery step has constant diagonalizability overhead because $d = 3^b$. For $r \geq 3$, we have $D_I \gg s_4$, and the estimated degree is s_4 . Thus additional factors reduce the estimated cost only to $s_4^{1/3} \approx 2^{14\lambda/9}$, which remains above 2^λ .

It remains to consider whether the attacker can improve this cost by sampling polynomially many choices of Φ_I . Borin et al. estimate the small-degree tail of the first successive minimum of a scalar-level-structure Hom lattice of covolume C by counting ideal classes. Their estimate gives, for $d^2 \ll x \ll \sqrt{C}$,

$$\Pr[\deg(\eta_{\min}) \leq x] \approx k_2 \left(\frac{x}{\sqrt{C}} \right)^2, \quad C = \frac{d^3 p}{4}, \quad k_2 \in [3/2, 6], \quad (2)$$

where $\eta_{\min}$ denotes the shortest admissible non-scalar isogeny. This is Equation (21) of [10, Section 6.2] with the reciprocal ratio. We use the form above because it agrees with the preceding $\Theta(x^2)$ counting argument and yields their Equation (22). Our analysis below relies only on this scaling.

Assume that Equation (2) applies to each of R choices of Φ_I . In the small-degree range, a union bound shows that the probability that at least one choice gives degree at most x is $O(Rx^2/C)$. Constant success probability therefore requires $x \gtrsim \sqrt{C/R} = s_4/(2\sqrt{R})$, and the quantum claw-finding cost satisfies

$$x^{1/3} \gtrsim C^{1/6} R^{-1/6}. \quad (3)$$

For $R = \text{poly}(\lambda)$, this is $2^{14\lambda/9 - O(\log \lambda)}$; moreover, the resulting degree remains above d^2 by an exponential factor, as required for the estimate above. Polynomially many choices of Φ_I can therefore provide only a polynomial-factor improvement and do not decrease the exponential rate below $14\lambda/9$.

References

1. Aardal, M.A., Adj, G., Aranha, D.F., Basso, A., Canales Martínez, I.A., Chávez-Saab, J., Corte-Real Santos, M., Dartois, P., De Feo, L., Duparc, M., Eriksen, J.K., Fouotsa, T.B., Filho, D.L.G., Hess, B., Kohel, D., Leroux, A., Longa, P., Maino, L., Meyer, M., Nakagawa, K., Onuki, H., Panny, L., Patranabis, S., Petit, C., Pope, G., Reijnders, K., Robert, D., Rodríguez Henríquez, F., Schaeffler, S., Wesolowski, B.: SQIsign. Algorithm specifications and supporting documentation, version 2.0, National Institute of Standards and Technology (2025), available at <https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures>
2. Basso, A., Dartois, P., De Feo, L., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-West - the fast, the small, and the safer. In: Chung, K.M., Sasaki, Y. (eds.) *Advances in Cryptology – ASIACRYPT 2024, Part III*. Lecture Notes in Computer Science, vol. 15486, pp. 339–370. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_11
3. Basso, A., Maino, L.: POKÉ: A compact and efficient PKE from higher-dimensional isogenies. In: Fehr, S., Fouque, P.A. (eds.) *Advances in Cryptology – EUROCRYPT 2025, Part II*. Lecture Notes in Computer Science, vol. 15602, pp. 94–123. Springer, Cham, Switzerland, Madrid, Spain (May 4–8, 2025). https://doi.org/10.1007/978-3-031-91124-8_4
4. Basso, A., Maino, L., Pope, G.: FESTA: Fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) *Advances in Cryptology – ASIACRYPT 2023, Part VII*. Lecture Notes in Computer Science, vol. 14444, pp. 98–126. Springer, Singapore, Singapore, Guangzhou, China (Dec 4–8, 2023). https://doi.org/10.1007/978-981-99-8739-9_4
5. Bendlin, R., Damgård, I.: Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In: Micciancio, D. (ed.) *TCC 2010: 7th Theory of Cryptography Conference*. Lecture Notes in Computer Science, vol. 5978, pp. 201–218. Springer Berlin Heidelberg, Germany, Zurich, Switzerland (Feb 9–11, 2010). https://doi.org/10.1007/978-3-642-11799-2_13
6. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient isogeny based signatures through class group computations. In: Galbraith, S.D., Moriai, S. (eds.) *Advances in Cryptology – ASIACRYPT 2019, Part I*. Lecture Notes in Computer Science, vol. 11921, pp. 227–247. Springer, Cham, Switzerland, Kobe, Japan (Dec 8–12, 2019). https://doi.org/10.1007/978-3-030-34578-5_9
7. Boneh, D., Boyen, X., Halevi, S.: Chosen ciphertext secure public key threshold encryption without random oracles. In: Pointcheval, D. (ed.) *Topics in Cryptology – CT-RSA 2006*. Lecture Notes in Computer Science, vol. 3860, pp. 226–243. Springer Berlin Heidelberg, Germany, San Jose, CA, USA (Feb 13–17, 2006). https://doi.org/10.1007/11605805_15
8. Bonnetain, X., Schrottenloher, A.: Quantum security analysis of CSIDH. In: Canteaut, A., Ishai, Y. (eds.) *Advances in Cryptology – EUROCRYPT 2020, Part II*. Lecture Notes in Computer Science, vol. 12106, pp. 493–522. Springer, Cham, Switzerland, Zagreb, Croatia (May 10–14, 2020). https://doi.org/10.1007/978-3-030-45724-2_17
9. Borin, G., Corte-Real Santos, M., Eriksen, J.K., Invernizzi, R., Mula, M., Schaeffler, S., Vercauteren, F.: Qlapoti: Simple and efficient translation of quaternion ideals to isogenies. In: Hanaoka, G., Yang, B.Y. (eds.) *Advances in Cryptology – ASIACRYPT 2025, Part IV*. Lecture Notes in Computer Science, vol. 16248, pp.

- 174–205. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_6
10. Borin, G., De Feo, L., Lido, G.M., Schaeffler, S.: The SQInstructor: A guide to SQIsign and the Deuring correspondence with level structures. *Cryptology ePrint Archive*, Report 2026/493 (2026). <https://eprint.iacr.org/2026/493>
11. Brandão, L.T.A.N., Peralta, R.: NIST First Call for Multi-Party Threshold Schemes. NIST Internal or Interagency Report (NISTIR) 8214C, National Institute of Standards and Technology, Gaithersburg, MD (Jan 2026). <https://doi.org/10.6028/NIST.IR.8214C>
12. Brzuska, C., Klooß, M., Woo, I.K.Y.: Threshold public-key encryption: Definitions, relations, and CPA-to-CCA transforms. *Cryptology ePrint Archive*, Report 2025/1665 (2025). <https://eprint.iacr.org/2025/1665>
13. Brzuska, C., Klooß, M., Woo, I.K.Y.: Threshold public-key encryption: Definitions, relations, and CPA-to-CCA transforms. In: Bai, S., Persichetti, E. (eds.) PKC 2026: 29th International Conference on Theory and Practice of Public Key Cryptography, Part IV. *Lecture Notes in Computer Science*, vol. 16554, pp. 162–197. Springer, Cham, Switzerland, West Palm Beach, FL, USA (May 25–28, 2026). https://doi.org/10.1007/978-3-032-26740-5_6
14. Cai, S., Chen, M., Lai, Y.F., Lin, K.: PIKE: Faster isogeny-based public key encryption with pairing-assisted decryption. In: Bai, S., Persichetti, E. (eds.) *Public-Key Cryptography – PKC 2026*. pp. 58–91. Springer Nature Switzerland, Cham (2026). https://doi.org/10.1007/978-3-032-26737-5_3
15. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) *Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science*, vol. 14008, pp. 423–447. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_15
16. Chen, M., Imran, M., Ivanyos, G., Kutas, P., Leroux, A., Petit, C.: Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH. In: Guo, J., Steinfeld, R. (eds.) *Advances in Cryptology – ASIACRYPT 2023, Part III. Lecture Notes in Computer Science*, vol. 14440, pp. 99–130. Springer, Singapore, Singapore, Guangzhou, China (Dec 4–8, 2023). https://doi.org/10.1007/978-981-99-8727-6_4
17. Cini, V., Lai, R.W.F., Woo, I.K.Y.: Pilvi: Lattice threshold PKE with small decryption shares and improved security. In: Hanaoka, G., Yang, B.Y. (eds.) *Advances in Cryptology – ASIACRYPT 2025, Part VI. Lecture Notes in Computer Science*, vol. 16250, pp. 539–569. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5119-4_17
18. Cong, K., Cozzo, D., Maram, V., Smart, N.P.: Gladius: LWR based efficient hybrid public key encryption with distributed decryption. In: Tibouchi, M., Wang, H. (eds.) *Advances in Cryptology – ASIACRYPT 2021, Part IV. Lecture Notes in Computer Science*, vol. 13093, pp. 125–155. Springer, Cham, Switzerland, Singapore (Dec 6–10, 2021). https://doi.org/10.1007/978-3-030-92068-5_5
19. Costello, C., Jao, D., Longa, P., Naehrig, M., Renes, J., Urbanik, D.: Efficient compression of SIDH public keys. In: Coron, J.S., Nielsen, J.B. (eds.) *Advances in Cryptology – EUROCRYPT 2017, Part I. Lecture Notes in Computer Science*, vol. 10210, pp. 679–706. Springer, Cham, Switzerland, Paris, France (Apr 30 – May 4, 2017). https://doi.org/10.1007/978-3-319-56620-7_24
20. Couveignes, J.M.: Hard homogeneous spaces. *Cryptology ePrint Archive*, Report 2006/291 (2006). <https://eprint.iacr.org/2006/291>

21. Dao, Q., Jain, A., Jin, Z.: Non-interactive zero-knowledge from LPN and MQ. In: Reyzin, L., Stebila, D. (eds.) *Advances in Cryptology – CRYPTO 2024, Part IX*. Lecture Notes in Computer Science, vol. 14928, pp. 321–360. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68400-5_10
22. Dartois, P.: Fast computation of 2-isogenies in dimension 4 and cryptographic applications. Cryptology ePrint Archive, Report 2024/1180 (2024), <https://eprint.iacr.org/2024/1180>
23. Dartois, P., Eriksen, J.K., Fouotsa, T.B., Merdy, A.H.L., Invernizzi, R., Robert, D., Rueger, R., Vercauteren, F., Wesolowski, B.: PEGASIS: Practical effective class group action using 4-dimensional isogenies. In: Kalai, Y.T., Kamara, S.F. (eds.) *Advances in Cryptology – CRYPTO 2025, Part I*. Lecture Notes in Computer Science, vol. 16000, pp. 67–99. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 17–21, 2025). https://doi.org/10.1007/978-3-032-01855-7_3
24. Dartois, P., Eriksen, J.K., Invernizzi, R., Vercauteren, F.: qt-Pegasis: Simpler and faster effective class group actions. In: Daemen, J., Thomé, E. (eds.) *Advances in Cryptology – EUROCRYPT 2026, Part IV*. Lecture Notes in Computer Science, vol. 16544, pp. 636–665. Springer, Cham, Switzerland, Rome, Italy (May 10–14, 2026). https://doi.org/10.1007/978-3-032-25327-9_22
25. Dartois, P., Maino, L., Pope, G., Robert, D.: An algorithmic approach to $(2, 2)$ -isogenies in the theta model and applications to isogeny-based cryptography. In: Chung, K.M., Sasaki, Y. (eds.) *Advances in Cryptology – ASIACRYPT 2024, Part III*. Lecture Notes in Computer Science, vol. 15486, pp. 304–338. Springer, Singapore, Singapore, Kolkata, India (Dec 9–13, 2024). https://doi.org/10.1007/978-981-96-0891-1_10
26. De Feo, L., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology* **8**(3), 209–247 (2014). <https://doi.org/10.1515/jmc-2012-0015>
27. De Feo, L., Meyer, M.: Threshold schemes from isogeny assumptions. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II*. Lecture Notes in Computer Science, vol. 12111, pp. 187–212. Springer, Cham, Switzerland, Edinburgh, UK (May 4–7, 2020). https://doi.org/10.1007/978-3-030-45388-6_7
28. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) *Advances in Cryptology – CRYPTO’89*. Lecture Notes in Computer Science, vol. 435, pp. 307–315. Springer, New York, Santa Barbara, CA, USA (Aug 20–24, 1989). https://doi.org/10.1007/0-387-34805-0_28
29. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: Abe, M. (ed.) *Advances in Cryptology – ASIACRYPT 2010*. Lecture Notes in Computer Science, vol. 6477, pp. 613–631. Springer Berlin Heidelberg, Germany, Singapore (Dec 5–9, 2010). https://doi.org/10.1007/978-3-642-17373-8_35
30. Dodis, Y., Katz, J.: Chosen-ciphertext security of multiple encryption. In: Kilian, J. (ed.) *TCC 2005: 2nd Theory of Cryptography Conference*. Lecture Notes in Computer Science, vol. 3378, pp. 188–209. Springer Berlin Heidelberg, Germany, Cambridge, MA, USA (Feb 10–12, 2005). https://doi.org/10.1007/978-3-540-30576-7_11
31. Frankel, Y., Gemmell, P., Yung, M.: Witness-based cryptographic program checking and robust function sharing. In: *28th Annual ACM Symposium on Theory of*

- Computing. pp. 499–508. ACM Press, Philadelphia, PA, USA (May 22–24, 1996). <https://doi.org/10.1145/237814.237998>
32. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust and efficient sharing of RSA functions. In: Koblitz, N. (ed.) *Advances in Cryptology – CRYPTO’96. Lecture Notes in Computer Science*, vol. 1109, pp. 157–172. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 1996). https://doi.org/10.1007/3-540-68697-5_13
 33. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalali, A., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Renes, J., Soukharev, V., Urbanik, D., Pereira, G., Karabina, K., Hutchinson, A.: SIKE. Tech. rep., National Institute of Standards and Technology (2020), available at <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions>
 34. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*. pp. 19–34. Springer Berlin Heidelberg, Germany, Taipei, Taiwan (Nov 29 – Dec 2 2011). https://doi.org/10.1007/978-3-642-25405-5_2
 35. Kani, E.: The number of curves of genus two with elliptic differentials. *Journal für die reine und angewandte Mathematik* **485**, 93–122 (1997). <https://doi.org/10.1515/crll.1997.485.93>
 36. Kim, H., Hong, S., Kim, S., Lee, S.: INKE: Fast isogeny-based PKE using intermediate curves. *Cryptology ePrint Archive*, Report 2025/1458 (2025), <https://eprint.iacr.org/2025/1458>
 37. Kim, H., Kim, W., Lee, C.: Threshold PRISM signature schemes via graph-based threshold access structures. *Cryptology ePrint Archive*, Report 2026/945 (2026), <https://eprint.iacr.org/2026/945>
 38. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion ℓ -isogeny path problem. *LMS Journal of Computation and Mathematics* **17**(A), 418–432 (2014). <https://doi.org/10.1112/S1461157014000151>
 39. Kraitsberg, M., Lindell, Y., Osheter, V., Smart, N.P., Talibi Alaoui, Y.: Adding distributed decryption and key generation to a ring-LWE based CCA encryption scheme. In: Jang-Jaccard, J., Guo, F. (eds.) *ACISP 19: 24th Australasian Conference on Information Security and Privacy. Lecture Notes in Computer Science*, vol. 11547, pp. 192–210. Springer, Cham, Switzerland, Christchurch, New Zealand (Jul 3–5, 2019). https://doi.org/10.1007/978-3-030-21548-4_11
 40. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) *Advances in Cryptology – EUROCRYPT 2023, Part V. Lecture Notes in Computer Science*, vol. 14008, pp. 448–471. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_16
 41. Maino, L., Petit, C.: Pushforward problems and applications to isogeny-based cryptography. *Cryptology ePrint Archive*, Report 2026/1030 (2026), <https://eprint.iacr.org/2026/1030>
 42. Micciancio, D., Suhl, A.: Simulation-secure threshold PKE from LWE with polynomial modulus. *IACR Communications in Cryptology* **1**(4), 2 (2025). <https://doi.org/10.62056/a0zogy4e->
 43. Milne, J.S.: Abelian Varieties. In: Cornell, G., Silverman, J.H. (eds.) *Arithmetic Geometry*, pp. 103–150. Springer New York, New York, NY (1986). https://doi.org/10.1007/978-1-4613-8655-1_5

44. Moriya, T.: Adaptive attacks on FESTA variants with masked-degree isogenies. Cryptology ePrint Archive, Report 2026/1240 (2026), <https://eprint.iacr.org/2026/1240>
45. Moriya, T.: On weak keys of POKE. Cryptology ePrint Archive, Report 2026/1002 (2026), <https://eprint.iacr.org/2026/1002>
46. Moriya, T., Stopar, M.: LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram. In: Hanaoka, G., Yang, B.Y. (eds.) *Advances in Cryptology – ASIACRYPT 2025, Part IV*. Lecture Notes in Computer Science, vol. 16248, pp. 275–306. Springer, Singapore, Singapore, Melbourne, VIC, Australia (Dec 8–12, 2025). https://doi.org/10.1007/978-981-95-5113-2_9
47. Naehrig, M., Renes, J.: Dual isogenies and their application to public-key compression for isogeny-based cryptography. In: Galbraith, S.D., Moriai, S. (eds.) *Advances in Cryptology – ASIACRYPT 2019, Part II*. Lecture Notes in Computer Science, vol. 11922, pp. 243–272. Springer, Cham, Switzerland, Kobe, Japan (Dec 8–12, 2019). https://doi.org/10.1007/978-3-030-34621-8_9
48. Nakagawa, K., Onuki, H.: QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) *Advances in Cryptology – CRYPTO 2024, Part V*. Lecture Notes in Computer Science, vol. 14924, pp. 75–106. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68388-6_4
49. Page, A., Robert, D.: Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766 (2023), <https://eprint.iacr.org/2023/1766>
50. Peikert, C.: He gives C-sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) *Advances in Cryptology – EUROCRYPT 2020, Part II*. Lecture Notes in Computer Science, vol. 12106, pp. 463–492. Springer, Cham, Switzerland, Zagreb, Croatia (May 10–14, 2020). https://doi.org/10.1007/978-3-030-45724-2_16
51. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) *Advances in Cryptology – CRYPTO 2019, Part I*. Lecture Notes in Computer Science, vol. 11692, pp. 89–114. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2019). https://doi.org/10.1007/978-3-030-26948-7_4
52. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) *Advances in Cryptology – EUROCRYPT 2023, Part V*. Lecture Notes in Computer Science, vol. 14008, pp. 472–503. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30589-4_17
53. Robert, D.: On the Efficient Representation of Isogenies. In: Dąbrowski, A., Pieprzyk, J., Pomykała, J. (eds.) *Number-Theoretic Methods in Cryptology*. pp. 3–84. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-82380-0_1
54. Shamir, A.: How to share a secret. *Communications of the ACM* **22**(11), 612–613 (Nov 1979). <https://doi.org/10.1145/359168.359176>
55. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: *35th Annual Symposium on Foundations of Computer Science*. pp. 124–134. IEEE Computer Society Press, Santa Fe, NM, USA (Nov 20–22, 1994). <https://doi.org/10.1109/SFCS.1994.365700>
56. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology* **15**(2), 75–96 (Mar 2002). <https://doi.org/10.1007/s00145-001-0020-9>

57. Silverman, J.H.: The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics, vol. 106. Springer, New York, NY (2009). <https://doi.org/10.1007/978-0-387-09494-6>
58. Takahashi, K., Hashimoto, K., Ogata, W.: Chosen-ciphertext secure code-based threshold public key encryptions with short ciphertext. *Designs, Codes and Cryptography* **92**(2), 277–301 (2024). <https://doi.org/10.1007/s10623-023-01309-2>
59. Vélú, J.: Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris* **273**, 238–241 (Jul 1971), <https://gallica.bnf.fr/ark:/12148/bpt6k56191248/f52.item>
60. Wesolowski, B.: The supersingular isogeny problem in time and memory $p^{1/3+o(1)}$. *Cryptology ePrint Archive, Report 2026/1486* (2026), <https://eprint.iacr.org/2026/1486>