Compact, Efficient and CCA-Secure Updatable Encryption from Isogenies

Antonin Leroux[1,2] and Maxime Roméas[3]

[1] DGA-MI, Bruz, France
[2] IRMAR - UMR 6625, Université de Rennes, France
antonin.leroux@polytechnique.org
[3] ANSSI, Paris, France
maxime.romeas@ssi.gouv.fr

Abstract. Updatable Encryption (UE) allows ciphertexts to be updated under new keys without decryption, enabling efficient key rotation. Constructing post-quantum UE with strong security guarantees is challenging: the only known CCA-secure scheme, COM-UE, uses bitwise encryption, resulting in large ciphertexts and high computational costs. We introduce DINE_ℓ, a family of compact and efficient CCA-secure isogeny-based post-quantum UE schemes parametrized by an integer ℓ, achieving increasing security levels for $\ell = 3, 4, 8$. Each encryption, decryption, or update requires only a few power-of-2 isogeny computations in dimension 2. At NIST security level 1, DINE_ℓ supports 28B messages, 480B ciphertexts, and $\ell \times 336$B update tokens, significantly smaller than prior constructions. Our C implementation demonstrates practical performance, with updates in $\ell \times 15$ms, encryptions in 65ms, and decryptions in 155ms. Our design combines high-dimensional isogeny representations with the Deuring correspondence. We also introduce new algorithms for the Deuring correspondence which may be of independent interest. Moreover, the security of our scheme relies on new problems that might open interesting perspectives in isogeny-based cryptography.

Keywords: Updatable Encryption, Isogenies, Deuring Correspondence

1 Introduction

Updatable Encryption (UE), introduced by Boneh *et al.* in 2013 [9], extends symmetric encryption by allowing ciphertexts to be re-encrypted under a new key without first decrypting them. Instead of downloading, decrypting, and re-encrypting large datasets, a client with the current key can generate a compact *update token* that enables an untrusted server to update all ciphertexts to the new key. This is especially valuable in cloud storage, where key rotation is essential to mitigate the impact of key compromise, but bandwidth and computational costs make naïve re-encryption impractical. In this work, we focus on the *ciphertext-independent* variant of UE, where a single token can update all ciphertexts under a given key. Beyond efficiency, UE must ensure *post-compromise security* and *forward secrecy* even if encryption keys and tokens may be exposed.

UE security definitions extend those for symmetric encryption to capture these stronger guarantees. Security is typically modeled over *epochs*, in which keys and update tokens may be adaptively revealed to the adversary. Early works proposed chosen plaintext attack (CPA) notions, such as IND-ENC and IND-UPD [34], which separately require indistinguishability of fresh encryptions and updated ciphertexts. Stronger notions, such as IND-UE [10], require indistinguishability between fresh and updated ciphertexts, thereby ensuring both confidentiality and unlinkability across key updates. In the classical setting, UE schemes secure under chosen ciphertext attacks (CCA) can be obtained by combining CPA-secure UE with ciphertext integrity (CTXT), using generic transforms such as Encrypt-and-MAC or Naor–Yung [32]. Achieving CCA security in the post-quantum setting, however, has proven significantly more challenging.

Lattice-based UE schemes have been built from the Learning with Errors (LWE) assumption, starting with Jiang's LWEUE [29]. Nishimaki's RtR [43] was the first ciphertext-independent UE scheme that is *unidirectional*, meaning that the new key cannot be recovered from the old key and the update token. Nishimaki showed that unidirectional UE schemes have stronger security than plain UE. Unlike RtR, the UE schemes introduced in this work are not unidirectional. Lattice-based UE schemes use key-homomorphic public key encryption to enable ciphertext updates and, when also plaintext-homomorphic, can achieve unidirectional security. However, they face two critical limitations. First, they support only a bounded number of updates, as ciphertext noise grows with each update. Second, their homomorphic properties and knowledge of update tokens allow an adversary to craft related-message ciphertexts, breaking CCA security. Consequently, all known lattice-based UE schemes achieve at most CPA security.

Cryptographic group actions, particularly isogeny-based ones, offer a compelling post-quantum alternative. There is now a wide variety of possible instantiations [11,25,14,18,4,27,7], that are believed to be quantum-resistant. Leroux and Roméas [38] adapted the classical UE scheme SHINE [10] to this setting in their GAINE construction, which could in principle achieve CCA security if instantiated with a group action satisfying both *mappability*—existence of an efficient bijection between the acted-upon set and bit strings—and *weak pseudorandomness* [3]. Unfortunately, no known post-quantum group action possesses both: isogeny-based actions lack mappability, and linear group actions cannot be weakly pseudorandom without unpractical restrictions.

Inspired by Moriya *et al.* SIGAMAL encryption [42], Leroux and Roméas [38] circumvented the mappability requirement by introducing the *Triple Orbital Group Action* (TOGA) abstraction, and instantiated the TOGA-UE scheme from isogeny-based group actions. Among post-quantum UE schemes, TOGA-UE offers the shortest ciphertexts—just an elliptic curve and a single point—and requires only one isogeny computation per encryption, update, or decryption. Despite this efficiency, this scheme is malleable and thus not CCA-secure.

Recently, Meers and Riepel [41] proposed COM-UE, the first CCA-secure post-quantum UE scheme. Derived from GAINE [38] using bitwise encryption and the Tag-then-Encrypt paradigm, COM-UE achieves IND-UE-CCA security

in the Algebraic Group Action Model [23] under an ad hoc assumption implied by the Discrete Logarithm with Auxiliary Input [5]. This resolves the open problem of constructing a CCA-secure post-quantum UE scheme, but at a steep cost: each encryption, decryption, and update requires one isogeny computation per plaintext and integrity tag bit. An overview of existing UE schemes, their hardness assumptions, and achieved security notions is given in Table 1.

Family	Scheme	Security (IND)	Assumption	Model
Discrete Log	RISE [34]	(rand, UE, CPA)	DDH	Standard
	E&M [32]	(det, ENC/UPD, CCA)	DDH	ROM
	SHINE0 [10]	(det, UE, CCA)	DDH, CDH	Ideal Cipher
Pairings	NYUAE [32]	(rand, ENC/UPD, RCCA)	SXDH	Standard
	SS23 [47]	(rand, UE, CPA)$^{+}$	SXDH	Standard
Lattices	Jia20 [29]	(rand, UE, CPA)	LWE	Standard
	Nis22 [43]	(rand, UE, CPA)	LWE	Standard
	GP23 [30]	(rand, UE, CPA)	LWE	Standard
Isogenies	GAINE0* [38]	(det, UE, CCA)	wk-PR, wk-UP	Ideal Cipher
	TOGA-UE [38]	(det, UE, CPA)	P-CSSDDH	Standard
	BIN-UE [41]	(det, UE, CPA)	wk-PR	Standard
	COM-UE [41]	(det, UE, CCA)	wk-PR, DLAI	ROM, AGAM
	DINE (this work)	(det, UE, CCA)	PP,UP	Ideal Cipher

* no secure instantiation known, $^{+}$ satisfies a stronger definition with expiry epochs

Table 1. Overview of existing ciphertext-independent UE constructions updated from [41] with our new construction DINE. We use rand for randomized updates and det for deterministic ones. Our new assumptions, PP and UP are presented in Section 4.

Metric	KeyGen/UpdateKey	Enc	Dec	Upd
Time (ms)	7000	65	155	$15 \times \ell$

Table 2. Average running time (over 10 runs) of the C implementation of DINE_ℓ on an Intel Core i7 at 2.3GHz with turbo-boost disabled, adapted from the Qlapoty code https://github.com/qlapoty/Qlapoty.

Overview of the contributions. We introduce DINE_ℓ, a family of isogeny-based post-quantum UE schemes achieving detIND-UE-CCA security, with each encryption, decryption, and ciphertext update requiring only a small number of power-of-2 isogeny evaluations in dimension 2. The integer parameter $\ell \geq 3$ provides a trade-off between efficiency and security. Its security relies on several new computational assumptions related to well-studied problems in isogeny-based cryptography, some of whose hardness depends on ℓ.

At NIST security level 1, DINE_3 encrypts 28B messages with 480B ciphertexts and 1008B tokens. For the same message size, the CSIDH-512 (which is considered below level 1 because of a subexponential quantum attack based on Kuperberg's algorithm) instantiation of the only other CCA-secure post-quantum UE scheme, COM-UE [41], has ciphertexts and tokens of size 22,528B. Assuming a very generous estimate of 10ms per group action computation for [41], each COM-UE encryption and update would require around 3.5s.

We further demonstrate the practicality of DINE_ℓ through a full C implementation, whose performance results are reported in Table 2. At NIST security level 1, updates take only $\ell \times 15$ms, encryptions 65ms, and decryptions 150ms. To the best of our knowledge, this is the first implementation of a post-quantum ciphertext-independent UE scheme reported in the literature.

To achieve this, we follow the recent trend [19,8,37,6] in isogeny-based cryptography of building cryptographic schemes from arbitrary degree isogenies by using two powerful tools: the high-dimensional (HD) isogeny representation stemming from Kani's Lemma, and the Deuring correspondence.

We also introduce several new algorithms around the Deuring correspondence that may be of independent interest, as well as novel computational problems that could serve as a foundation for future cryptographic constructions.

Technical overview. We now provide an overview of the main technical ideas behind our construction. The detailed description can be found in Section 3.2.

The basic principle behind our encryption mechanism is as follows: given a secret isogeny of public domain, messages can be encrypted by mapping them invertibly to points of the domain, and then evaluating the secret isogeny on these points. The existence of a dual isogeny, that acts almost as an inverse, gives the means to decrypt.

This symmetric encryption scheme can be made updatable by computing tokens as alternate paths between the codomains of two secret keys. A ciphertext is updated by pushing its components (which are image points) through this token isogeny. Such a token can be efficiently computed if one knows the quaternion ideals corresponding to the secret isogenies via the Deuring correspondence, using for instance the KLPT algorithm [33] and the ideal-to-isogeny machinery at the core of the SQIsign signature scheme and its variants [22,19,8].

The only problem is that the resulting scheme is not secure. Indeed, isogenies act linearly on points, and thus any linear relation between some ciphertexts is preserved under updates. In the quantum setting, this leads to an attack against the IND-UE property, since three points always admit a linear relation, which can be efficiently found in quantum polynomial time using Shor's algorithm. To prevent this attack, we replace points with isogenies. Indeed, every subgroup of an elliptic curve defines the kernel of some isogeny, yielding a one-to-one correspondence between isogenies and their kernels. In most cases, one can "evaluate"[4] an isogeny through another isogeny by computing the image of its kernel. Thus, if messages can be mapped to isogenies invertibly, we can translate our previous scheme into a UE scheme where ciphertexts are isogenies instead of points. If it is possible to go from the ciphertext isogenies to their kernels efficiently, then the security of this new scheme is no better than the previous one. However, if the ciphertext isogenies are chosen so that even representing a point of their kernel is infeasible, then our scheme appears resistant to known attacks.

This property is achieved by using isogenies of large prime degree N over a field of characteristic p such that p has large order in $\mathbb{F}_N^\times$. That way, the points of order N (which constitute the kernel of isogenies of degree N) are defined over

[4] this is called a pushforward

an extension field of degree roughly N, making explicitly representing a kernel point infeasible. For random large primes p and N this will be true with overwhelming probability. Recent progress in isogeny-based cryptography has made it possible to efficiently handle and represent large prime degree isogenies using high-dimensional isogenies [19,8], enabling their practical use in our scheme.

Thus, there is some hope for security, but a key challenge remains: mapping messages to isogenies in an invertible way. Unlike mapping messages to points, this is nontrivial for isogenies. Notably, the kernels of our ciphertext isogenies are deliberately hard to compute, so mapping through them is infeasible. Moreover, computing the HD representation of an isogeny without prior knowledge of an efficient representation is hard, ruling out a mapping via this representation.

To address this challenge, we rely on the Deuring correspondence. Indeed, mapping messages to ideals invertibly is straightforward, and using the ideal-to-isogeny algorithm from [8], one can efficiently compute their HD representation, enabling practical encryption. Decryption, however, requires inverting this mapping, *i.e.*, recovering the ideal from the HD representation. While a quantum polynomial-time algorithm exists for this task [13,15], no efficient classical algorithm is known—in fact, this computational problem was even used to build the pSIDH key exchange protocol [35].

Our final idea to overcome this obstacle is to use endomorphisms instead of individual isogenies. Indeed, an endomorphism of degree N^2 can be decomposed into two isogenies of degree N between the same pair of curves. This allows us to construct our scheme as before, but with each ciphertext consisting of two degree N isogenies instead of one. Every endomorphism of a curve can be represented by four integers corresponding to its decomposition in a basis of the endomorphism ring. Crucially, these integers can be efficiently recovered by evaluating the endomorphism on points of sufficiently large order to compute trace pairings with the basis elements (see, *e.g.*, the CheckTrace algorithm [35, Algorithm 9]). Hence, the integers can be extracted from the HD representations—composed of evaluated points—of the two degree N isogenies forming the endomorphism, enabling efficient inversion of the HD representation back to the endomorphism.

If the endomorphism ring is well-chosen, a random string can be mapped to four integers representing an endomorphism of norm N^2. For instance, in the public domain with j-invariant 1728, the endomorphism ring contains a suborder with basis $1, i, j, k$, and the norm of an endomorphism $x+iy+jz+kt$ is $x^2+y^2+p(z^2+t^2)$. Our idea is to map a message together with a random string to a pair of integers $z, t \leq \sqrt{N^2/2p}$. For each choice of z, t, suitable integers x, y can be efficiently computed using Cornacchia's algorithm with probability roughly $1/\log p$. By trying $O(\log p)$ random strings, one can construct an endomorphism $\theta = x+iy+jz+kt$ to encrypt the message. As discussed above, the integers z, t can then be recovered efficiently from the two degree N isogenies composing θ, providing all the ingredients needed to implement our UE scheme.

The security of our scheme relies on several new problems related to the following assumption in characteristic p: the pushforwards of two degree N isogenies through a secret isogeny are computationally indistinguishable from two random

degree N isogenies, provided N and p are chosen such that the N-torsion points of supersingular elliptic curves in characteristic p cannot be efficiently computed.

We conclude our overview by explaining how we efficiently "evaluate" secret and token isogenies on ciphertext isogenies using the HD representation. In particular, Kani's Lemma—the key ingredient of the HD representation—enables the pushforward of an N-isogeny through a $(2^f - N)$-isogeny by computing a 2^f-isogeny in dimension 2. Consequently, secret key and token isogenies of degree $(2^f - N)^\ell$ can be efficiently used by sequentially applying Kani's Lemma ℓ times to push N-isogenies through $(2^f - N)^\ell$-isogenies.

The exponent ℓ may differ between keys and tokens, with smaller values yielding better efficiency. Ideally, one would take $\ell = 1$, but this does not yield a secure instantiation, as we will see. We propose three values, $\ell = 3, 4, 8$, each providing stronger security guarantees than the previous one. However, finding isogenies of prescribed norm between two curves is generally hard, and known algorithms such as KLPT (and variants [33,22]) require sufficiently large norms. Even our conservative choice $\ell = 8$ is too small. Thus, a larger exponent may be necessary for either keys or tokens—for instance, $\ell = 8$ for keys, but then $\ell > 8$ is required for tokens, or vice versa.

Accordingly, we consider two variants of our scheme: one with $\ell \leq 8$ for tokens, optimizing update speed, and one with $\ell \leq 8$ for secret keys, optimizing encryption and decryption speed. We prioritize the former, as update efficiency is more critical and the latter is harder to analyze rigorously.

Outline of the paper. Section 2 reviews UE and isogeny background; Section 3 presents DINE and Section 4 analyzes its security; Section 5 details the algorithmic instantiation, and Section 6 covers its implementation.

2 Preliminaries

Notations. We write $s \leftarrow \mathcal{D}$ to denote sampling from a distribution $\mathcal{D}$. An algorithm $\mathcal{A}$ with oracle access to or is denoted $\mathcal{A}^{\mathsf{or}}$. For a function f depending (possibly implicitly) on a parameter λ, we write $f = \mathsf{negl}(\lambda)$ when $f = o(\lambda^{-n})$ for all $n \in \mathbb{N}$. Throughout, $\mathcal{O}_0$ denotes the special maximal order in the quaternion algebra $B_{p,\infty}$ defined in Section 2.2. For pushforwards of ideals or isogenies (see Section 2.2), we use the shorthand $[I]_*(J, K) := ([I]_*J, [I]_*K)$.

2.1 Updatable Encryption

In this section, we describe the syntax and security definitions of UE, following the presentations of [43,10,34,29]. An UE scheme operates in *epochs*, where each epoch corresponds to an index incremented with every key update. For proof purposes, let n denote the maximum number of epochs.

Definition 1. *An updatable encryption scheme* UE *for message space* $\mathcal{M}$ *consists of a tuple of probabilistic polynomial time algorithms* (UE.Setup, UE.KeyGen, UE.TokenGen, UE.Enc, UE.Dec, UE.Upd) *where:*

- $\mathsf{UE.Setup}(1^\lambda) \to \mathsf{pp}$*: The setup algorithm takes as input the security parameter and outputs a public parameter* pp*.*
- $\mathsf{UE.KeyGen}(\mathsf{pp}) \to k_\mathsf{e}$*: The key generation algorithm takes as input the public parameter* pp *and outputs an epoch key* k_e*.*
- $\mathsf{UE.Enc}(k, m) \to c$*: The encryption algorithm takes as input an epoch key* k*, a message* m*, and outputs a ciphertext* c*.*
- $\mathsf{UE.Dec}(k, c) \to m$*: The decryption algorithm takes as input an epoch key* k*, a ciphertext* c*, and outputs a message* m *or* $\perp$*.*
- $\mathsf{UE.TokenGen}(k_\mathsf{e}, k_{\mathsf{e}+1}) \to \Delta_{\mathsf{e}+1}$*: The token generation algorithm takes as input two keys of consecutive epochs* e *and* $\mathsf{e}+1$ *and outputs a token* $\Delta_{\mathsf{e}+1}$*.*
- $\mathsf{UE.Upd}(\Delta_{\mathsf{e}+1}, c_\mathsf{e}) \to c_{\mathsf{e}+1}$*: The update algorithm takes as input a token* $\Delta_{\mathsf{e}+1}$*, a ciphertext* c_e*, and outputs a ciphertext* $c_{\mathsf{e}+1}$*.*

Definition 2 (Correctness). *For any* $m \in \mathcal{M}$*, for* $1 \le \mathsf{e}_1 \le \mathsf{e}_2 \le n$*, it holds that* $\Pr[\mathsf{UE.Dec}(k_{\mathsf{e}_2}, c_{\mathsf{e}_2}) \neq m] \le \mathsf{negl}(\lambda)$*, where* $\mathsf{pp} \leftarrow \mathsf{UE.Setup}(1^\lambda)$*,* $k_{\mathsf{e}_1}, \ldots, k_{\mathsf{e}_2} \leftarrow \mathsf{UE.KeyGen}(\mathsf{pp})$*,* $c_{\mathsf{e}_1} \leftarrow \mathsf{UE.Enc}(k_{\mathsf{e}_1}, m)$*, and* $\Delta_{i+1} \leftarrow \mathsf{UE.TokenGen}(k_i, k_{i+1})$*,* $c_{i+1} \leftarrow \mathsf{UE.Upd}(\Delta_{i+1}, c_i)$ *for* $i \in [\mathsf{e}_1, \mathsf{e}_2 - 1]$*.*

Security definitions. In all our UE schemes, the Upd algorithm is *deterministic*. We therefore consider only security definitions in the deterministic update setting. An adaptation of the usual indistinguishability security notions for UE is given in Definition 3. We briefly explain the oracles used in this definition; they are formally defined in Fig. 2. The Enc and Dec oracles implement standard encryption and decryption queries. The Next oracle increments the epoch counter and samples a new secret key along its associated token; if a challenge ciphertext exists, it is updated accordingly. The Corr oracle, on input key (resp. token) and an epoch, returns the corresponding key (resp. token). The Chall oracle samples a uniform bit b and returns either an encryption of the input message or an update of the input ciphertext, depending on b. The returned ciphertext is the challenge, and the adversary's goal is to guess b. Finally, the $\mathsf{Upd}\tilde{\mathsf{C}}$ oracle returns the updated version of the challenge ciphertext. All these oracles track the values returned to, or computable by, the adversary; these values are stored in the sets denoted in calligraphic font (see Section 2.1 for details).

Definition 3 ($\mathsf{detIND\text{-}UE\text{-}atk}$ [10])**.** *Let* $\mathsf{UE} = (\mathsf{UE.Setup}, \mathsf{UE.KeyGen}, \mathsf{UE.TokenGen}, \mathsf{UE.Enc}, \mathsf{UE.Dec}, \mathsf{UE.Upd})$ *be an updatable encryption scheme. The* $\mathsf{detIND\text{-}UE\text{-}atk}$ *advantage, for* $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA}\}$*, of an adversary* $\mathcal{A}$ *against* UE *is given by*

$$\mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}atk}}_{\mathsf{UE}}(\mathcal{A}) := |\Pr[\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}atk\text{-}0}}_{\mathsf{UE},\mathcal{A}} = 1] - \Pr[\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}atk\text{-}1}}_{\mathsf{UE},\mathcal{A}} = 1]|$$

where the confidentiality experiment $\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}atk}\text{-}b}_{\mathsf{UE},\mathcal{A}}$ *is given in Fig. 1.*

Next, in the ciphertext integrity (CTXT) game, the adversary is given access to the oracles $\mathcal{O}.\mathsf{Enc}$, $\mathcal{O}.\mathsf{Next}$, $\mathcal{O}.\mathsf{Upd}$, and $\mathcal{O}.\mathsf{Corr}$. At some point, $\mathcal{A}$ attempts to provide a ciphertext forgery via the oracle $\mathcal{O}.\mathsf{Try}$ defined in Fig. 1. The adversary wins if the forgery is valid, *i.e.*, if it decrypts to a message rather than $\perp$.

$\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}atk}\text{-}b}_{\mathsf{UE},\mathcal{A}}(\lambda)$

$\mathbf{Setup}(1^\lambda)$
$\mathtt{ors} \leftarrow \mathcal{O}.\{\mathsf{Enc}, \mathsf{Upd}, \mathsf{Next}, \mathsf{Corr}\}$
if $\mathsf{atk} = \mathsf{CCA}$ **then** $\mathtt{ors} \leftarrow \mathtt{ors} \cup \{\mathcal{O}.\mathsf{Dec}\}$
$(\bar{M}, \bar{C}) \leftarrow \mathcal{A}^{\mathtt{ors}}(1^\lambda)$; $\mathsf{phase} \leftarrow 1$; $\tilde{\mathsf{e}} \leftarrow \mathsf{e}$
$\tilde{C}_{\mathsf{e}} \leftarrow \mathcal{O}.\mathsf{Chall}(\bar{M}, \bar{C})$
$b' \leftarrow \mathcal{A}^{\mathtt{ors}, \mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}}(\tilde{C}_{\mathsf{e}})$
if $\mathcal{K}^* \cap \mathcal{C}^* \neq \emptyset$ **or** $\mathcal{I}^* \cap \mathcal{C}^* \neq \emptyset$ **then** $\mathsf{twf} \leftarrow 1$
if $\mathsf{twf} = 1$ **then** $b' \xleftarrow{\$} \{0,1\}$
return b'

$\mathbf{Exp}^{\mathsf{INT\text{-}CTXT}^{\mathsf{s}}}_{\mathsf{UE},\mathcal{A}}(\lambda)$

$\mathbf{Setup}(1^\lambda)$
$\mathsf{win} \leftarrow 0$
$\mathcal{A}^{\mathcal{O}.\mathsf{Enc}, \mathcal{O}.\mathsf{Next}, \mathcal{O}.\mathsf{Upd}, \mathcal{O}.\mathsf{Corr}, \mathcal{O}.\mathsf{Try}}(\lambda)$
if $\mathsf{twf} = 1$ **then** $\mathsf{win} \leftarrow 0$
return win

$\mathcal{O}.\mathsf{Try}(\tilde{C})$

if $\mathsf{phase} = 1$ **then return** $\perp$
$\mathsf{phase} \leftarrow 1$
if $(\mathsf{e} \in \mathcal{K}^*$ **or** $\tilde{C} \in \mathcal{L}^*)$ **then** $\mathsf{twf} \leftarrow 1$
M **or** $\perp \leftarrow \mathsf{UE.Dec}(k_{\mathsf{e}}, \tilde{C})$
if $M \neq \perp$ **then** $\mathsf{win} \leftarrow 1$

Fig. 1. Description of $\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}atk}\text{-}b}_{\mathsf{UE},\mathcal{A}}$, $\mathbf{Exp}^{\mathsf{INT\text{-}CTXT}^{\mathsf{s}}}_{\mathsf{UE},\mathcal{A}}$ and oracle $\mathcal{O}.\mathsf{Try}$ for UE scheme UE (with deterministic updates) and adversary $\mathcal{A}$, for $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. Oracles are given in Fig. 2. Trivial win conditions are discussed in Section 2.1 and Appendix A.

$\mathbf{Setup}(1^\lambda)$
$\mathsf{pp} \leftarrow \mathsf{UE.Setup}(1^\lambda)$
$k_1 \leftarrow \mathsf{UE.KeyGen}(\mathsf{pp})$
$\Delta_1 \leftarrow \perp$; $\mathsf{e} \leftarrow 1$
$\mathsf{phase}, \mathsf{twf}, \mathsf{c} \leftarrow 0$
$\mathcal{L}, \tilde{\mathcal{L}}, \mathcal{C}, \mathcal{K}, \mathcal{T} \leftarrow \emptyset$

$\mathcal{O}.\mathsf{Enc}(M)$
$C \leftarrow \mathsf{UE.Enc}(k_{\mathsf{e}}, M)$
$\mathsf{c} \leftarrow \mathsf{c} + 1$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathsf{c}, C, \mathsf{e})\}$
return C

$\mathcal{O}.\mathsf{Dec}(C)$
if $\mathsf{phase} = 1$ **and** $(C, \mathsf{e}) \in \tilde{\mathcal{L}}$
  $\mathsf{twf} \leftarrow 1$
M **or** $\perp \leftarrow \mathsf{UE.Dec}(k_{\mathsf{e}}, C)$
return M **or** $\perp$

$\mathcal{O}.\mathsf{Next}$
$\mathsf{e} \leftarrow \mathsf{e} + 1$
$k_{\mathsf{e}} \leftarrow \mathsf{UE.KeyGen}(\mathsf{pp})$
$\Delta_{\mathsf{e}} \leftarrow \mathsf{UE.TokenGen}(k_{\mathsf{e}-1}, k_{\mathsf{e}})$
if $\mathsf{phase} = 1$
  $\tilde{C}_{\mathsf{e}} \leftarrow \mathsf{UE.Upd}(\Delta_{\mathsf{e}}, \tilde{C}_{\mathsf{e}-1})$
  $\tilde{\mathcal{L}} \leftarrow \tilde{\mathcal{L}} \cup \{(\tilde{C}_{\mathsf{e}}, \mathsf{e})\}$

$\mathcal{O}.\mathsf{Upd}(C_{\mathsf{e}-1})$
if $(j, C_{\mathsf{e}-1}, \mathsf{e}-1) \notin \mathcal{L}$
  return $\perp$
$C_{\mathsf{e}} \leftarrow \mathsf{UE.Upd}(\Delta_{\mathsf{e}}, C_{\mathsf{e}-1})$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(j, C_{\mathsf{e}}, \mathsf{e})\}$
return C_{e}

$\mathcal{O}.\mathsf{Corr}(\mathsf{inp}, \hat{\mathsf{e}})$
if $\hat{\mathsf{e}} > \mathsf{e}$ **then return** $\perp$
if $\mathsf{inp} = \mathsf{key}$
  $\mathcal{K} \leftarrow \mathcal{K} \cup \{\hat{\mathsf{e}}\}$
  return $k_{\hat{\mathsf{e}}}$
if $\mathsf{inp} = \mathsf{token}$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\hat{\mathsf{e}}\}$
  return $\Delta_{\hat{\mathsf{e}}}$

$\mathcal{O}.\mathsf{Chall}(\bar{M}, \bar{C})$
if $\mathsf{phase} \neq 1$ **then return** $\perp$
if $(\cdot, \bar{C}, \mathsf{e}-1) \notin \mathcal{L}$ **then return** $\perp$
if $b = 0$ **then** $\tilde{C}_{\mathsf{e}} \leftarrow \mathsf{UE.Enc}(k_{\mathsf{e}}, \bar{M})$
else $\tilde{C}_{\mathsf{e}} \leftarrow \mathsf{UE.Upd}(\Delta_{\mathsf{e}}, \bar{C})$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathsf{e}\}$; $\tilde{\mathcal{L}} \leftarrow \tilde{\mathcal{L}} \cup \{(\tilde{C}_{\mathsf{e}}, \mathsf{e})\}$
return $\tilde{C}_{\mathsf{e}}$

$\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}$
if $\mathsf{phase} \neq 1$ **then return** $\perp$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathsf{e}\}$
$\tilde{\mathcal{L}} \leftarrow \tilde{\mathcal{L}} \cup \{(\tilde{C}_{\mathsf{e}}, \mathsf{e})\}$
return $\tilde{C}_{\mathsf{e}}$

Fig. 2. Oracles in security games for UE with deterministic updates. Computing the leakage sets is discussed in Section 2.1 and Appendix A.

Definition 4 ([10]). *Let* $\mathsf{UE} = \{\mathsf{UE.KeyGen}, \mathsf{UE.TokenGen}, \mathsf{UE.Enc}, \mathsf{UE.Dec}, \mathsf{UE.Upd}\}$ *be an UE scheme. The* $\mathsf{INT\text{-}CTXT^s}$ *advantage of an adversary* $\mathcal{A}$ *against* UE *is defined as* $\mathbf{Adv}^{\mathsf{INT\text{-}CTXT^s}}_{\mathsf{UE}}(\mathcal{A}) := \Pr[\mathbf{Exp}^{\mathsf{INT\text{-}CTXT^s}}_{\mathsf{UE},\mathcal{A}} = 1]$ *where the experiment* $\mathbf{Exp}^{\mathsf{INT\text{-}CTXT^s}}_{\mathsf{UE},\mathcal{A}}$ *is given in Fig. 1.*

Leakage sets. We follow the bookkeeping technique [34,10] to maintain the epoch leakage sets. We denote by $\mathcal{C}$ the set of epochs in which the adversary obtained an updated version of the challenge ciphertext (via $\mathcal{O}.\mathsf{Chall}$ or $\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}$), $\mathcal{K}$ the set of epochs in which the adversary corrupted the encryption key, and $\mathcal{T}$ the set of epochs in which the adversary corrupted the update token.

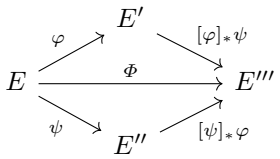
In addition, the adversary may learn ciphertexts and their updates. We denote by $\mathcal{L}$ the set of non-challenge ciphertexts (from $\mathcal{O}.\mathsf{Enc}$ or $\mathcal{O}.\mathsf{Upd}$) and by $\tilde{\mathcal{L}}$ the set of updated versions of the challenge ciphertext (generated by $\mathcal{O}.\mathsf{Next}$ and returned by $\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}$). Extended leakage sets (denoted with a star) account for keys, tokens, and ciphertext updates inferrable from adversarial queries. These sets along with some examples of trivial wins are discussed in Appendix A.

2.2 Isogenies

We assume familiarity with elliptic curves and isogenies up to the level of [26], and recall the necessary background in App. C. For background on quaternion algebras and the Deuring correspondence, see [49]. We also provide a self-contained summary in App. D, adapted from [50].

Below, we recall useful conventions and notions. Let p be a prime congruent to 3 mod 4. Then $B_{p,\infty} = B(-1,-p)$ [45], where $B_{p,\infty}$ denotes the unique quaternion algebra ramified exactly at p and ∞. In this case, $B_{p,\infty}$ contains the maximal order $\mathcal{O}_0 := \langle 1, i, \frac{1+j}{2}, i\frac{1+j}{2}\rangle$. Throughout this paper, p is always a prime congruent to 3 mod 4, and $\mathcal{O}_0$ always denotes this specific maximal order.

Commutative isogeny diagrams. Let E, E', E'' be elliptic curves and $\varphi : E \to E'$, $\psi : E \to E''$ two separable isogenies of coprime degrees d_1 and d_2. Then there exists a fourth elliptic curve E''' together with two *pushforward isogenies* $[\varphi]_*\psi$ and $[\psi]_*\varphi$ going from E' and E'' to E''' and satisfying $\ker([\varphi]_*\psi) = \varphi(\ker(\psi))$, $\deg([\varphi]_*\psi) = d_2$ and $\ker([\psi]_*\varphi) = \psi(\ker(\varphi))$, $\deg([\psi]_*\varphi) = d_1$. This gives rise to the commutative diagram below, where both paths correspond to two decompositions of the same isogeny $\Phi = [\psi]_*\varphi \circ \psi = [\varphi]_*\psi \circ \varphi$.



The dual notion of *pullback isogenies* is defined as follows: given $\varphi : E \to E'$ and $\rho : E' \to E'''$ of coprime degrees, the pullback of ρ by φ is $[\varphi]^*\rho := [\hat{\varphi}]_*\rho$.

Efficient high-dimensional isogeny representation. Building on insights from SIDH attacks, an isogeny $\varphi : E \to E'$ can be efficiently represented using the images

of a basis P, Q of $E[2^f]$, provided $\deg \varphi < 2^f$. The most efficient known method employs dimension 2 isogenies and evaluates P, Q via an auxiliary isogeny $\psi : E \to E''$ of degree $2^f - \deg \varphi$. The main theoretical result behind this way of representing isogenies is called Kani's Lemma [31].

Theorem 1 (Kani's Lemma, [31]). *Let d_1 and d_2 be coprime positive integers. Given a (d_1, d_2)-isogeny commutative diagram as above, where $[\varphi]_*\psi$ is denoted by ψ' and $[\psi]_*\varphi$ is denoted by φ', the isogeny $\Psi : E \times E''' \to E' \times E''$ corresponding to the matrix*

$$\begin{pmatrix} \varphi & \hat{\psi}' \\ -\psi & \hat{\varphi}' \end{pmatrix}$$

is a (d_1+d_2)-isogeny between these products of elliptic curves with their principal product polarisation, with kernel $\ker \Psi = \{(\hat{\varphi}(R), \psi'(R)) \mid R \in E'[d_1 + d_2]\}$.

It is then routine to verify that $\ker \Psi = \{(\deg \varphi)R, (\psi' \circ \varphi)(R) \mid R \in E[2^f]\}$. Hence, knowing $\varphi(P)$ and $\varphi(Q)$, *i.e.* the action of φ on the 2^f-torsion of E, together with the ability to construct an arbitrary isogeny $\psi' : E' \to E'''$ of degree $2^f - \deg \varphi$, suffices to compute Ψ. By [20], Ψ can then be efficiently evaluated on any point of $E \times E'''$. In particular, since $\Psi((R, 0_{E'''})) = (\varphi(R), -\psi(R))$ for any $R \in E$, we can efficiently evaluate φ everywhere. This observation motivates the following definition of a *high-dimensional (HD) representation* of an isogeny φ.

Definition 5. *Let $\varphi : E \to E'$ be an isogeny of (implicitly known) degree $N < 2^f$. An* HD representation *of φ is a tuple $s_{\mathsf{HD}}(\varphi) := E, P, Q, E', \varphi(P), \varphi(Q)$ where P, Q form a basis of $E[2^f]$. If E, E' can be computed deterministically from their isomorphism classes over $\mathbb{F}_{p^2}$, they are called* canonical curves. *If P, Q can be deterministically generated, they form a* canonical basis *of $E[2^f]$. If E, E' and P, Q are both canonical, then P, Q can be omitted from the isogeny representation. Furthermore, if the y-coordinate of $\varphi(P)$ is smaller, in the lexicographical order on $\mathbb{F}_{p^2}$ viewed as $\mathbb{F}_p^2$, than that of $(-\varphi)(P)$, we call the representation* canonical *and write $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi) := E, E', \varphi(P), \varphi(Q)$.*

3 UE from the Deuring Correspondence

Parameters. We consider a finite field of characteristic $p := c2^f - 1$, where the cofactor c is as small as possible. This choice ensures that the 2^f-torsion of supersingular curves is defined over $\mathbb{F}_{p^2}$. Moreover, we select a prime $N < 2^f$ such that p has maximal order in $\mathbb{F}_N^\star$. This ensures that the N-torsion points of elliptic curves lie in a field of maximal degree, which is crucial for security.

3.1 Algorithmic Building Blocks

In this section, we list the key algorithmic building blocks used in our construction. For now, these algorithms are treated as black boxes; their precise instantiation is provided and explained in Section 5.

- $\mathsf{EvalAnyIsoCanBasis}$: given an ideal I corresponding to an isogeny $\varphi_I : E \to E_I$, outputs its canonical HD representation $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_I)$.
- $\mathsf{HDPushForward}$: given $s_{\mathsf{HD}}(\varphi_1), s_{\mathsf{HD}}(\varphi_2)$ for two isogenies φ_1, φ_2 with the same domain and respective degrees N and $2^f - N$, outputs $s_{\mathsf{HD}}([\varphi_2]_*\varphi_1)$.
- $\mathsf{ToCanonicalRep}$: converts an HD representation $s_{\mathsf{HD}}(\varphi)$ into its canonical form $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi)$.
- $\mathsf{PartialRepresentInteger}_{N^2}$: given integers z, t, outputs either $\perp$ or integers $x, y > 0$ such that $x^2 + y^2 + p(z^2 + t^2) = N^2$.
- $\mathsf{IdealPullBack}$: given an ideal I corresponding to $\varphi_I : E \to E_I$ and $s^{\mathsf{can}}_{\mathsf{HD}}(\psi)$ with $\psi : E_I \to E$ of degree N, outputs $s^{\mathsf{can}}_{\mathsf{HD}}([\varphi_I]^*\psi)$.
- $\mathsf{HalfEndoToCoord}$: given $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2)$ for two isogenies with the same domain E_0 and isomorphic codomain, outputs (when possible) the four coordinates x, y, z, t such that the endomorphism $\hat{\varphi}_2 \circ \varphi_1$ corresponds via the Deuring correspondence to the principal ideal $\mathcal{O}_0(x + iy + jz + kt)$.
- $\mathsf{SpecialKLPT}_{2^f - N}$: given an ideal I, computes an equivalent ideal $J \sim_{\mathbb{Z}+N\mathcal{O}_L(I)} I$ of norm $(2^f - N)^e$ for some $e \in \mathbb{N}$.

3.2 The DINE Scheme

In this section, we present our family of scheme DINE_ℓ (Deuring Ideal cipher Nonce-based Encryption). The parameter ℓ is a small integer. As noted in the technical overview, there is a trade-off between the efficiency of $\mathsf{Enc}/\mathsf{Dec}$ and that of Upd. We choose to prioritize faster updates, which requires selecting the update token before generating the updated secret key. Consequently, we slightly modify the standard UE interface. Instead of the usual $\mathsf{TokenGen}$ algorithm, which takes two keys $k_{\mathsf{e}}, k_{\mathsf{e}+1}$ generated via KeyGen and outputs the token $\Delta_{\mathsf{e}+1}$, we introduce an $\mathsf{UpdateKey}$ algorithm. $\mathsf{UpdateKey}$ takes only k_{e} as input and outputs both $k_{\mathsf{e}+1}$ and $\Delta_{\mathsf{e}+1}$. This modification does not affect the overall functionality of the UE scheme.

Remark 1. In Appendix E, we describe a second variant of our scheme, which prioritizes faster Enc and Dec. In this variant, since the updated keys are generated before the update tokens, it adheres to the standard UE framework.

We assume an invertible map $\pi : \{0,1\}^{n(\lambda)+m(\lambda)} \to [0, m]^2$ with $m = N/\sqrt{2p}$, where $\{0,1\}^{m(\lambda)}$ is the message space and $\{0,1\}^{n(\lambda)}$ is the nonce space. We use canonical isogeny representations to prevent malleability attacks against the CCA security of our scheme. A description of our scheme is provided in Fig. 3. For clarity, we also include diagrams illustrating the main operations: key generation and rotation in Fig. 4, encryption in Fig. 5, and ciphertext updates in Fig. 6.

We first prove the correctness of ciphertext updates, and then the correctness of DINE_ℓ as a whole.

Proposition 1 (Correctness of updates). *Let k_{e} be a key corresponding to an ideal I_{e}, and define $C^{(\mathsf{e})} := ([\varphi_{I_{\mathsf{e}}}]_*\varphi_{I_1}, [\varphi_{I_{\mathsf{e}}}]_*\varphi_{I_2})$ for ideals $I_1 := \mathcal{O}_0\langle\theta_{N,M}, N\rangle$ and $I_2 := \mathcal{O}_0\langle\overline{\theta_{N,M}}, N\rangle$, where $N \in \mathbb{N}$ and $\theta_{N,M}$ is a quaternion with $\mathrm{Nrd}(\theta_{N,M}) = N^2$. If $k_{\mathsf{e}+1}, \Delta_{\mathsf{e}+1} := \mathsf{UpdateKey}(k_{\mathsf{e}})$, then $\mathsf{Upd}(\Delta_{\mathsf{e}+1}, C^{(\mathsf{e})}) = (\pm[\varphi_{I_{\mathsf{e}+1}}]_*\varphi_{I_1}, \pm[\varphi_{I_{\mathsf{e}+1}}]_*\varphi_{I_2})$ where $I_{\mathsf{e}+1}$ is the ideal corresponding to $k_{\mathsf{e}+1}$ under the Deuring correspondence.*

$\mathsf{KeyGen}(\mathsf{pp})$

$I \leftarrow$ elt. of a random $(\mathbb{Z} + N\mathcal{O}_0)$-ideal class.
return $\mathsf{SpecialKLPT}_{2^f - N}(I)$

$\mathsf{UpdateKey}(k_\mathsf{e})$

Parse k_e as an ideal $J_{\mathsf{e}+1}^{(0)}$
for $1 \le i \le \ell$
  $J_{\mathsf{e}+1}^{(i)} \leftarrow$ random $\mathcal{O}_R(J_{\mathsf{e}+1}^{(i-1)})$-ideal of norm $2^f - N$
  $s_{\mathsf{HD}}^{\mathsf{can}}(\delta_{\mathsf{e}+1}^{(i)}) \leftarrow \mathsf{EvalAnyIsoCanBasis}(J_{\mathsf{e}+1}^{(i)})$
$\Delta_{\mathsf{e}+1} = (s_{\mathsf{HD}}^{\mathsf{can}}(\delta_{\mathsf{e}+1}^{(i)}))_{1 \le i \le \ell}$
$J_{\mathsf{e}+1} \leftarrow \prod_{i=1}^{\ell} J_{\mathsf{e}+1}^{(i)}$
$I_{\mathsf{e}+1} \leftarrow \mathsf{SpecialKLPT}_{2^f - N}(k_\mathsf{e} \cdot J_{\mathsf{e}+1})$
return $I_{\mathsf{e}+1}, \Delta_{\mathsf{e}+1}$

$\mathsf{Upd}(\Delta_{\mathsf{e}+1}, C_\mathsf{e})$

Parse $\Delta_{\mathsf{e}+1}$ as $(s_{\mathsf{HD}}^{\mathsf{can}}(\delta_{\mathsf{e}+1}^{(i)}))_{1 \le i \le \ell}$
Parse C_e as $s_1, s_2 := s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_1), s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_2)$
for $1 \le i \le \ell$
  $s_1 \leftarrow \mathsf{HDPushForward}(s_1, s_{\mathsf{HD}}^{\mathsf{can}}(\delta_{\mathsf{e}+1}^{(i)}))$
  $s_2 \leftarrow \mathsf{HDPushForward}(s_2, s_{\mathsf{HD}}^{\mathsf{can}}(\delta_{\mathsf{e}+1}^{(i)}))$
$s_1 \leftarrow \mathsf{ToCanonicalRep}(s_1)$
$s_2 \leftarrow \mathsf{ToCanonicalRep}(s_2)$
return $C_{\mathsf{e}+1} := (s_1, s_2)$

$\mathsf{Enc}(k_\mathsf{e}, M)$

do
  Generate a random nonce R
  $(z, t) \leftarrow \pi(R \| M)$
  $s \leftarrow \mathsf{PartialRepresentInteger}_{N^2}(z, t)$
while $s = \perp$
Parse s as two integers x, y
$\theta_{N,M} \leftarrow x + iy + jz + kt$
$I_1 \leftarrow \mathcal{O}_0\langle\theta_{N,M}, N\rangle$; $I_2 \leftarrow \mathcal{O}_0\langle\overline{\theta_{N,M}}, N\rangle$
$I_1 \leftarrow [k_\mathsf{e}]_* I_1$; $I_2 \leftarrow [k_\mathsf{e}]_* I_2$
$s_1 := s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_{I_1}) \leftarrow \mathsf{EvalAnyIsoCanBasis}(I_1)$
$s_2 := s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_{I_2}) \leftarrow \mathsf{EvalAnyIsoCanBasis}(I_2)$
return $C_\mathsf{e} := (s_1, s_2)$

$\mathsf{Dec}(k_\mathsf{e}, C_\mathsf{e})$

Parse C_e as $s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_1), s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_2)$
if $s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_1) \neq \mathsf{ToCanonicalRep}(s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_1))$
 or $s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_2) \neq \mathsf{ToCanonicalRep}(s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_2))$
  return $\perp$
$s_1 \leftarrow \mathsf{IdealPullBack}(k_\mathsf{e}, s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_1))$
$s_2 \leftarrow \mathsf{IdealPullBack}(k_\mathsf{e}, s_{\mathsf{HD}}^{\mathsf{can}}(\varphi_2))$
$x', y', z', t' \leftarrow \mathsf{HalfEndoToCoord}(s_1, s_2)$
$x, y \leftarrow \mathsf{PartialRepresentInteger}_{N^2}(z', t')$
if $(x, y) \neq (x', y')$ **then return** $\perp$
$R' \| M' \leftarrow \pi^{-1}(z', t')$
return M'

Fig. 3. Our updatable encryption scheme DINE_ℓ.

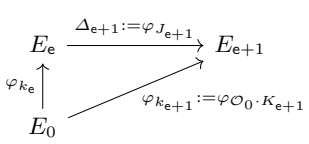


Fig. 4. The commutative diagram representing the key k_e together with the new key $k_{\mathsf{e}+1}$ and the update token $\Delta_{\mathsf{e}+1}$ produced by $\mathsf{UpdateKey}$. We keep the notations of Fig. 3.

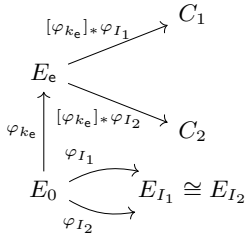


Fig. 5. Encryption of M into a ciphertext (C_1, C_2) under the key k_e. We keep the notations of Fig. 3.

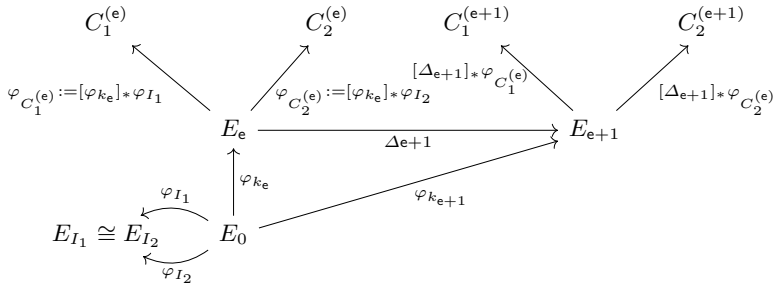


Fig. 6. Update of the ciphertext $(C_1^{(e)}, C_2^{(e)})$ under the key k_e to $(C_1^{(e+1)}, C_2^{(e+1)})$ under the key k_{e+1} using the token Δ_{e+1}. We keep the notations of Fig. 3.

Proof. We prove correctness for the first ciphertext component; the argument for the second one is analogous. The canonical isogeny representation may replace one or both ciphertext isogenies by their opposites, so the final equality holds up to a sign. We omit this sign change for clarity. Retaining the notations of the proposition, let J_{e+1} denote the ideal corresponding to the token isogeny Δ_{e+1}.

It suffices to show that, for any isogeny ψ of degree N with domain E_0, we have $[\Delta_{e+1} \circ \varphi_{k_e}]_* \psi = [\varphi_{k_{e+1}}]_* \psi$. Once this equality holds, we obtain

$$C_1^{(e+1)} := [\Delta_{e+1}]_* C_1^{(e)} = [\Delta_{e+1}]_* ([\varphi_{k_e}]_* \varphi_{I_1}) = [\Delta_{e+1} \circ \varphi_{k_e}]_* \varphi_{I_1} = [\varphi_{k_{e+1}}]_* \varphi_{I_1}$$

which establishes the correctness of the update for any choice of φ_{I_1} and thus proves the proposition. This property is equivalent to requiring that $\Delta_{e+1} \circ \varphi_{k_e}$ and $\varphi_{k_{e+1}}$ have the same codomain, and that both send the cyclic subgroups of $E_0[N]$ to the same cyclic subgroups of $E_{e+1}[N]$. To translate this statement into the language of quaternions, we introduce some notations: let $\mathfrak{O} := \mathcal{O}_0$, $\mathfrak{O}' := \mathbb{Z} + N\mathcal{O}_0$, $\mathfrak{a} := I_e \cdot J_{e+1}$ and $\mathfrak{b} := I_{e+1}$.

By definition of SpecialKLPT and the relation $\sim_{\mathfrak{O}'}$, we have that $K_{e+1} := \mathfrak{O}' \cap \mathfrak{b}$ is equivalent to $\mathfrak{O}' \cap \mathfrak{a}$. Then, [22, Lemma 1] implies the existence of $\alpha \in \mathfrak{O}' \cap \mathfrak{a}$ such that $\mathrm{Nrd}(\alpha) = \mathrm{Nrd}(\mathfrak{a}) \mathrm{Nrd}(\mathfrak{b})$ and $\mathfrak{O}' \cap \mathfrak{b} = (\mathfrak{O}' \cap \mathfrak{a}) \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{a})$. Consequently, $\mathfrak{b} = \mathfrak{a} \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{a})$, and we obtain $\mathcal{O}_R(\mathfrak{a}) = \alpha / \mathrm{Nrd}(\mathfrak{b}) \cdot \mathcal{O}_R(\mathfrak{b}) \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{a})$.

Hence, in the quaternion world, the codomain condition is equivalent to $\mathcal{O}_R(\mathfrak{a})$ and $\mathcal{O}_R(\mathfrak{b})$ being isomorphic—an explicit isomorphism is provided in the previous paragraph. The condition on cyclic subgroups holds if the isogeny $\hat{\varphi}_{k_{e+1}} \circ \Delta_{e+1} \circ \varphi_{k_e}$ associated with the ideal $\mathfrak{a} \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{b}) \cdot \bar{\mathfrak{b}} \cdot \alpha / \mathrm{Nrd}(\mathfrak{a})$ acts as scalar multiplication by an integer coprime to N on $E_0[N]$. From the relation $\mathfrak{b} = \mathfrak{a} \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{a})$ we deduce

$$\begin{aligned} \mathfrak{a} \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{b}) \cdot \bar{\mathfrak{b}} \cdot \alpha / \mathrm{Nrd}(\mathfrak{a}) &= \mathfrak{a} \cdot \bar{\alpha} / \mathrm{Nrd}(\mathfrak{b}) \cdot \alpha / \mathrm{Nrd}(\mathfrak{a}) \cdot \bar{\mathfrak{a}} \cdot \alpha / \mathrm{Nrd}(\mathfrak{a}) \\ &= \mathfrak{a} \cdot \bar{\mathfrak{a}} \cdot \alpha / \mathrm{Nrd}(\mathfrak{a}) \\ &= \mathfrak{O} \, \mathrm{Nrd}(\mathfrak{a}) \cdot \alpha / \mathrm{Nrd}(\mathfrak{a}) \\ &= \mathfrak{O}\alpha, \text{ where } \alpha \in \mathfrak{O}' \cap \mathfrak{a} \subseteq \mathfrak{O}'. \end{aligned}$$

Since $\alpha \in \mathfrak{O}'$ and $\mathrm{Nrd}(\alpha) = (2^f - N)^{\ell+e} \mathrm{Nrd}(\mathfrak{b})$, for some $e \in \mathbb{N}$, is coprime to N (noting that the reduced discriminant of $\mathfrak{O}'$ is pN^3 and, by definition of the

relation $\sim_{\mathfrak{O}'}$, $\mathrm{Nrd}(\mathfrak{b})$ is coprime to it), the cyclic subgroups condition is satisfied. Hence, the updates are indeed correct. □

Proposition 2 (Correctness). *The* DINE *scheme of Fig. 3 is correct.*

Proof. Let $1 \leq \mathsf{e}_1 \leq \mathsf{e}_2 \leq n$ be two epochs, and consider a ciphertext $C^{(\mathsf{e}_2)}$ obtained by successively updating an initial ciphertext $C^{(\mathsf{e}_1)} := \mathsf{Enc}(k_{\mathsf{e}_1}, M)$ through the tokens Δ_{i+1} for $i \in [\mathsf{e}_1, \mathsf{e}_2 - 1]$, as in Definition 2. By definition, $C^{(\mathsf{e}_1)} = (\pm[\varphi_{I_{\mathsf{e}_1}}]_*\varphi_{I_1}, \pm[\varphi_{I_{\mathsf{e}_1}}]_*\varphi_{I_2})$, where $I_1 := \mathcal{O}_0\langle\theta_{N,M}, N\rangle$, $I_2 := \mathcal{O}_0\langle\overline{\theta_{N,M}}, N\rangle$, and $\theta_{N,M} := x + yi + zj + tij$ satisfies $\mathrm{Nrd}(\theta_{N,M}) = N^2$ with $z, t = \pi(R\|M)$.

By Proposition 1, updating $C^{(\mathsf{e}_1)}$ through the sequence of tokens yields $C^{(\mathsf{e}_2)} = (\pm[\varphi_{I_{\mathsf{e}_2}}]_*\varphi_{I_1}, \pm[\varphi_{I_{\mathsf{e}_2}}]_*\varphi_{I_2})$. Applying the IdealPullBack algorithm to I_{e_2} and the HD representations of $C_1^{(\mathsf{e}_2)}$, $C_2^{(\mathsf{e}_2)}$ recovers the HD representations of φ_{I_1} and φ_{I_2}. Then, HalfEndoToCoord applied to these representations returns the coordinates x, y, z, t of $\theta_{N,M}$, from which we recover the original message as $\pi^{-1}(z, t) = \pi^{-1}(\pi(R\|M)) = R\|M$. The sign discrepancy potentially caused by the canonical isogeny representation is corrected by HalfEndoToCoord, which ensures $y \neq 0$ and multiplies x, y, z, t by the sign of y so that x and y are positive, as in the output of SpecialRepresentInteger.

The correctness of IdealPullBack and HalfEndoToCoord is established in Section 5.1 and Section 5.3, respectively. □

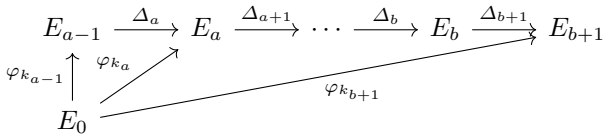
4 Security

In this section, we prove the CPA security and ciphertext integrity of DINE, introduce the new hardness assumptions underlying its security, and deduce CCA security using the generic composition result of Boyd *et al.*

4.1 Security Assumptions

First and foremost, we need to identify the underlying hard problems of our scheme. To this end, we introduce several new problems and briefly discuss their presumed hardness. All of these problems involve computing pushforwards of N-isogenies through a secret $(2^f - N)$-isogeny. Throughout, whenever we refer to an isogeny φ, we implicitly assume access to an efficient representation of φ, which is guaranteed by Kani's Lemma (see Section 2.2).

Definition 6 (Pseudorandom pushforward). *Let $\mathcal{O}_0$ be the special maximal order of $B_{p,\infty}$ with $p := c2^f - 1$, let $\ell \in \mathbb{N}^*$, and let N be prime. The adversary starts the experiment by sending two positive integers $a \leq b$, and a left $\mathcal{O}_0$-ideal k_{a-1}. The experiment then completes the following commutative diagram:*



where the Δ_i are random $(2^f - N)^\ell$-isogenies and $\varphi_{k_{b+1}}$ is a random isogeny such that the endomorphism $\hat{\varphi}_{k_{b+1}} \circ \Delta_{b+1} \circ \ldots \circ \Delta_a \circ \varphi_{k_{a-1}}$ leaves any subgroup of order N in $E_0[N]$ stable, and sends $\Delta_{a+1}, \ldots, \Delta_b$ and k_{b+1} to the adversary. Let $\mathcal{O}_a$ be the maximal order of $B_{p,\infty}$ corresponding to E_a, and let $\mathcal{I}$ be a distribution over pairs of $\mathcal{O}_0$-ideals of norm N.

The experiment samples a random bijection ρ from the set of $\mathcal{O}_0$-ideals of norm N to the set of $\mathcal{O}_a$-ideals of norm N. Define two distributions:

- *$\mathcal{D}_0$: output $((\varphi_{I_1}, \varphi_{I_2}), ([\varphi_{k_a}]_*\varphi_{I_1}, [\varphi_{k_a}]_*\varphi_{I_2}))$, where $(I_1, I_2) \leftarrow \mathcal{I}$.*
- *$\mathcal{D}_1$: output $((\varphi_{I_1}, \varphi_{I_2}), (\varphi_{\rho(I_1)}, \varphi_{\rho(I_2)}))$, where $(I_1, I_2) \leftarrow \mathcal{I}$.*

The advantage of an adversary $\mathcal{A}$ in breaking the pseudorandomness of the $(\mathcal{O}_0, N)$-pushforward is $\mathbf{Adv}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{D}_0} = 1] - \Pr[\mathcal{A}^{\mathcal{D}_1} = 1]|$.

We say that the decomposition (φ_1, φ_2) of a norm N^2 endomorphism $\theta := x + iy + jz + kt$ is *valid* if $\mathsf{PartialRepresentInteger}_{N^2}(z, t) = (x, y)$.

Definition 7 (Unpredictable pushforward). *We keep the same notations and setup as in the first paragraph of Definition 6. The adversary is given access to two oracles* New *and* Push *(initially, $Q = \emptyset$, $L = [\,]$, and $c = 0$):*

$\mathsf{New}()$	$\mathsf{Push}(i)$ $\quad$ (*where* $i \in [0, c-1]$)
$(I_1, I_2) \leftarrow \mathcal{I}$	$(I_1, I_2) \leftarrow L[i]$
$L[c] \leftarrow (I_1, I_2)$	$(C_1, C_2) \leftarrow ([\varphi_{k_a}]_*\varphi_{I_1}, [\varphi_{k_a}]_*\varphi_{I_2})$
$c \leftarrow c + 1$	$Q \leftarrow Q \cup \{(C_1, C_2)\}$
return $(\varphi_{I_1}, \varphi_{I_2})$	**return** (C_1, C_2)

The adversary outputs a pair of isogenies (ψ_1, ψ_2) in canonical form and wins if $(\psi_1, \psi_2) \notin Q$ and $([\varphi_{k_a}]^*\psi_1, [\varphi_{k_a}]^*\psi_2)$ is a valid decomposition of a norm N^2 endomorphism. The success probability of an adversary $\mathcal{A}$ in breaking the unpredictability of the $(\mathcal{O}_0, N)$-pushforward is denoted $\mathbf{Adv}^{\mathsf{UP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{A})$.

The security assumption behind the security of our scheme is the following:

Assumption 1. *For $\lambda > 0$, in prime characteristic p of 3λ bits, if $2^f - N$ and N are primes of 3λ bits, p has maximal order in $\mathbb{F}_N^\times$, $\ell \geq 3$, $\mathcal{I}$ has λ bits of min-entropy, and k_{a-1} is obtained from an honest DINE key chain then, for all probabilistic polynomial time adversaries $\mathcal{A}$, $\mathbf{Adv}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{A})$, and $\mathbf{Adv}^{\mathsf{UP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{A})$ are $\mathsf{negl}(\lambda)$.*

We now justify its plausibility, starting with the most natural strategy, that is recovering k_a. The isogenies Δ_i have prime norm $(2^f - N)^\ell$, so there are at least $2^f - N$ isogenies to test, which is too large to recover k_a through Δ_a or Δ_{a+1}. A second approach is to compute the endomorphism ring of E_a directly. The best known algorithm has complexity $O(p^{1/3+o(1)})$ [51], which we circumvent by taking p to have 3λ bits.

Another concern is the adversary's access to the ideals k_{a-1} and k_{b+1}, which reveals the endomorphism rings of E_{a-1} and E_{b+1}. By definition, the adversary will also know the class of the $\mathbb{Z}+N\mathcal{O}_{a-1}$-ideal containing the ideal corresponding to $\Delta_{b+1} \circ \ldots \circ \Delta_a$. This may be problematic depending on ℓ, as recovering this ideal gives k_a. There are $O(pN^3)$ classes of $\mathbb{Z} + N\mathcal{O}_{a-1}$-ideals. If the class of k_a contains exponentially many ideals of norm $(2^f - N)^{\ell(2+b-a)}$, finding the right one in polynomial time seems infeasible. The case $a = b$ is worst, as the number of ideals of norm M grows linearly with M.

A common and plausible heuristic is that a given class of $\mathfrak{O}$-ideals, where $\mathfrak{O}$ is a quaternion order of reduced discriminant D, contains $O(M/D)$ distinct ideals of norm M (this holds on average by a simple counting argument). We have $D = pN^3 \approx p^4$ and $p \approx N \approx 2^f - N$ in our case. When $\ell = 1$, it can be shown that the ideal we seek will be the smallest in its class with high probability, and can thus be recovered with LLL. When $\ell = 2$, the ideal is not the smallest in its class, and it is unclear whether a polynomial-time algorithm can find it; as far as we know, the best known approach is the SpecialKLPT algorithm presented later, which only succeeds in polynomial time when the required norm is much larger. However, since there are only $O(1)$ possible ideals, the problem reduces to representing $(2^f - N)^4$ by the norm form of a $\mathbb{Z} + N\mathcal{O}_{a-1}$-ideal, a problem that, to our knowledge, has not been used in cryptography.

When $\ell \geq 3$, there should be at least $O(p^2)$ ideals of the correct norm, and so even if they can be found efficiently, there does not seem to be an easy way to identify the correct one. If $\ell = 4$, one can even show, using the mixing property of the isogeny graph with level structure defined by the subgroup of scalar matrices that have been shown to be Ramanujan by [16, Theorem 1.6], that the class of $\overline{k_{b+1}} \cdot k_{a-1}$ should be uniformly random among all possible isogeny classes, and is therefore independent of the remaining information. Hence, knowledge of these ideals cannot help to solve the problem in any meaningful way. Finally, if $\ell = 8$, the same argument shows that the key k_a is uniformly distributed within its $\mathbb{Z}+N\mathcal{O}_0$-ideal class, making PP even harder in this setting. Following the analysis above, we consider $\ell = 3, 4, 8$ as three candidates for securely instantiating our family DINE_ℓ, providing a trade-off between efficiency and security.

Exploiting the oracles. In both the PP and UP experiments, the adversary has access to an oracle computing pushforwards of isogenies of degree N. Our problems are therefore naturally viewed as variants of isogeny problems with torsion information (or level structure), first studied in the context of SIDH [28], as there is a correspondence between isogenies and subgroups. In the computational variants of these problems, the goal is to recover a secret isogeny of known degree given its action on points of the domain. When one has access to the image of the secret isogeny on the full S-torsion for a smooth integer S the problem can be solved in polynomial time using the same techniques we use constructively in this work [46,12,39]. When the torsion information has large prime order, or is not exact, the problem becomes harder. A comprehensive survey is [21].

In our case, the order is a precisely a large prime, and we operate in the regime where it appears hard to relate the isogeny information provided to the

adversary to an explicit torsion information as the N-torsion points are defined only over an extension of $\mathbb{F}_p$ of exponential degree, making it infeasible to even write down a single point of order N. For all these reasons, the variant of the isogeny problem with torsion information considered in this work appears to be hard. We also take $2^f - N$ to be prime as this seems to give the hardest possible version of this problem [40].

Beyond recovering k_a, there does not seem to be a way to attack PP. For UP, there is a bit more to say. In particular, we need to rule out some easy attacks exploiting the malleability of our ciphertexts. The first countermeasure is to make the isogeny representation uniquely determined by the isomorphism class of the isogeny. We take canonical domain and codomain in their isomorphism class, a canonical evaluation basis, and the sign choice between ψ and $-\psi$ is fixed. This should prevent the adversary from maliciously replacing an isogeny representation to trivially win the game. The second countermeasure is to verify that the resulting endomorphism is valid. This prevents the adversary from taking a known pair of valid outputs (ψ_1, ψ_2) corresponding to some endomorphism θ and cheating by providing (ψ_1, ψ_1) or (ψ_2, ψ_1). Indeed, since PartialRepresentInteger ensures that $y \neq 0$, the pair (ψ_1, ψ_1), which would yield $\theta' = N$, is rejected. Similarly, since PartialRepresentInteger ensures that $x, y > 0$, the pair (ψ_2, ψ_1) is rejected, as the corresponding endomorphism would be $\pm\overline{\theta}$, with either $y < 0$ or $x < 0$. The adversary could also try to combine two distinct pairs of output, but then the probability that this yields a valid endomorphism after pulling back through φ_{k_a} should be negligible.

4.2 detIND-UE-CPA Security of DINE

We show that DINE achieves detIND-UE-CPA security in the ideal cipher model, under the hardness of the pseudorandom pushforward problem (Definition 6).

Theorem 2 (DINE is detIND-UE-CPA). *Let* DINE *be the UE scheme described in Fig. 3 for the special maximal order $\mathcal{O}_0$ and $N \in \mathbb{N}$. For any ideal cipher model adversary $\mathcal{A}$, there exists a reduction $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}CPA}}_{\mathsf{DINE}}(\mathcal{A}) \leq 2n^3 \cdot \mathbf{Adv}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{B}) + \mathsf{negl}(\lambda).$$

Proof overview. We provide a high-level overview of our proof here; the full details are given in Appendix F, along with a complete reduction shown in Fig. 10.

We follow the strategy of [10], using a hybrid argument over *insulated regions*. An insulated region is an epoch interval $[\mathsf{e}_1, \mathsf{e}_2]$ in which no keys are compromised, and neither the entering token Δ_{e_1} nor the exiting token Δ_{e_2+1} is revealed. The epochs e_1 and e_2 are referred to as the left and right *firewalls* of the region, respectively. Intuitively, a secure UE scheme should preserve ciphertext confidentiality within insulated regions. The detIND-UE-CPA notion formalizes this: the only epochs in which an adversary may request a challenge ciphertext without immediately triggering a trivial win must lie within such a region.

The hybrid argument goes as follows. In the i-th hybrid, if the detIND-UE-CPA challenge occurs before (resp. after) the i-th insulated region, the challenger always returns an encryption of the plaintext (resp. an update of the ciphertext) supplied by the adversary. If the challenge occurs within the i-th region, a random bit decides whether an encryption or an update is returned.

We define a reduction $\mathcal{B}$ that receives a maximal order $\mathcal{O}_0$, an integer N, and a distribution $\mathcal{I}$. The reduction guesses the i-th insulated region $[a, b]$ and uses the PP experiment to simulate the detIND-UE-CPA experiment for DINE. It first samples $k_1 \leftarrow \mathsf{KeyGen}$ and computes the UpdateKey chain up to k_{a-1}. It then provides k_{a-1} and the interval $[a, b]$ to the PP experiment, which returns isogenies $\Delta_{a+1}, \ldots, \Delta_b$, an ideal k_{b+1}, and access to an oracle BB implementing either the distribution $\mathcal{D}_0$ or $\mathcal{D}_1$ of the PP experiment (cf. Definition 6). The reduction then resumes the UpdateKey chain from k_{b+1}, allowing it to efficiently simulate the key chain outside the insulated region.

Concretely, BB returns either tuples of the form $((\varphi_{I_1}, \varphi_{I_2}), [\varphi_{k_a}]_*(\varphi_{I_1}, \varphi_{I_2}))$ or $((\varphi_{I_1}, \varphi_{I_2}), (\varphi_{J_1}, \varphi_{J_2}))$, where k_a is a fixed left $\mathcal{O}_0$-ideal and right $\mathcal{O}_a$-ideal, $(I_1, I_2) \leftarrow \mathcal{I}$, and J_1, J_2 are random left $\mathcal{O}_a$-ideals of norm N. Inside the insulated region $[a, b]$, the reduction embeds the ideal k_a into the epoch key of the left firewall a. Specifically, $(\varphi_{I_1}, \varphi_{I_2})$ is associated with the nonce-message pair (R, M), and $[\varphi_{k_a}]_*(\varphi_{I_1}, \varphi_{I_2})$ serves as the ciphertext at epoch a.

In DINE, fresh ciphertexts are randomized via a nonce R, while updates are deterministic. The reduction must therefore ensure consistency of ciphertexts, *i.e.*, the nonce value R must remain consistent. The ideal cipher model allows the reduction to program the public permutation π to ensure this consistency. For each encryption query, the reduction associates a nonce-message pair (R, M) with the isogeny pair $(\varphi_{I_1}, \varphi_{I_2})$ supplied by BB, programming π using the coordinates of the associated endomorphism. The reduction carefully retains this association for subsequent updates of the ciphertext.

Finally, we show that if BB implements $\mathcal{D}_0$, then $\mathcal{B}$ perfectly simulates the detIND-UE-CPA experiment. If BB implements $\mathcal{D}_1$, the ciphertexts produced inside the insulated region are random pairs of isogenies and are independent of the challenge bit. Thus, an efficient adversary $\mathcal{A}$ against the detIND-UE-CPA security of DINE allows $\mathcal{B}$ to break the pseudorandomness of the $(\mathcal{O}_0, N)$-pushforward.

4.3 detIND-UE-CCA Security of DINE

We show that DINE achieves detIND-UE-CCA security in the ideal cipher model, under the hardness of the unpredictable pushforward problem (cf. Definition 7).

Theorem 3 (DINE is INT-CTXT^{s}). *Let* DINE *be the UE scheme described in Fig. 3 with the special maximal order $\mathcal{O}_0$ and $N \in \mathbb{N}$. For any ideal cipher model adversary $\mathcal{A}$, there exists a reduction $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT^s}}_{\mathsf{DINE}}(\mathcal{A}) \leq n \cdot \mathbf{Adv}^{\mathsf{UP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{B}) + \mathsf{negl}(\lambda).$$

Corollary 1. *Combining the results of Theorems 2, 3 and 4, we conclude that* DINE *achieves* detIND-UE-CCA *security under Assumption 1.*

Proof. Let $\mathcal{A}$ be an adversary against $\mathsf{INT\text{-}CTXT}^{\mathsf{s}}$ that outputs a valid forgery $\tilde{C}_e$ with probability $\Pr[E]$. We build a reduction $\mathcal{B}$, described in Fig. 12 of App. G, that wins the UP experiment with probability $\Pr[E]/n$.

We follow the simulation strategy of the proof of Th. 2, with one optimization specific to the $\mathsf{INT\text{-}CTXT}^{\mathsf{s}}$ experiment [41]. Since the reduction can abort immediately after the $\mathcal{O}.\mathsf{Try}$ query, it never needs to leave the insulated region containing the forgery. Thus, if a is the epoch at which the UP key is embedded, the tokens $\Delta_{a+1}, \ldots, \Delta_n$ suffice to simulate all epochs from a through n.

The main additional difficulty is to ensure that a valid $\mathsf{INT\text{-}CTXT}^{\mathsf{s}}$ forgery does not correspond to a pushforward previously returned by the UP oracle. Let $\mathcal{K}^*$ and $\mathcal{T}^*$ denote the extended key and token leakage sets. Suppose the forgery attempt $\mathcal{O}.\mathsf{Try}(\tilde{C}_e)$ occurs at epoch e, and let $[\mathsf{fwl}, \mathsf{fwr}]$ be the insulated region containing e. We define $a := \max\{i \in [\mathsf{fwl}, e] \mid i \notin \mathcal{T}^*\}$, where $i \notin \mathcal{T}^*$ means that the token Δ_i is unknown to the adversary. By definition, no keys inside $[a, e]$ are known to the adversary, the token Δ_a is unknown, and all subsequent tokens leading to epoch e are known. The reduction guesses a, incurring a n loss in its success probability, then sends $[a, n]$ and k_{a-1} to the UP experiment.

As in the proof of Th. 2, the reduction embeds the secret ideal k_a of the UP experiment in the key k_a, uses the New oracle to generate ideals associated to nonce-message pairs, programs the ideal cipher to ensure nonce consistency, and uses the Push oracle to get ciphertext values in epoch a when needed.

When $\mathcal{A}$ makes a forgery attempt $\mathcal{O}.\mathsf{Try}(\tilde{C}_e)$, $\mathcal{B}$ checks whether its guess for a is correct. If so, it "downgrades" $\tilde{C}_e$ to epoch a by computing the successive pullbacks $(\psi_1^*, \psi_2^*) := [\bigcirc_{i=e}^{a+1} \Delta_i]^* \tilde{C}_e$. By correctness of DINE, the pullback of (ψ_1^*, ψ_2^*) through φ_{k_a} is a valid decomposition of a norm N^2 endomorphism whenever $\tilde{C}_e$ is a valid ciphertext. The reduction therefore wins the UP experiment provided that $(\psi_1^*, \psi_2^*) \notin Q$, the query set of previously returned pushforwards.

Suppose otherwise, then (ψ_1^*, ψ_2^*) was previously returned by Push in response to an encryption or update query in some epoch $i \in [a, e]$. By definition of a, all tokens $\Delta_{i+1}, \ldots, \Delta_e$ are known to the adversary. Hence, the adversary can compute the corresponding update of (ψ_1^*, ψ_2^*) to epoch e. By correctness of updates, this ciphertext is exactly $\tilde{C}_e$. This contradicts the assumption that $\tilde{C}_e$ is a valid forgery. Thus, $(\psi_1^*, \psi_2^*) \notin Q$, and $\mathcal{B}$ wins the UP experiment with probability $\Pr[E]/n$. □

5 Algorithmic Instantiation

This section provides detailed descriptions of the building blocks required by our UE construction, as listed in Section 3.1. We begin with algorithms that need no full description, as they can be derived from existing building blocks or explained informally.

- $\mathsf{EvalAnyIsoCanBasis}$ follows directly from $\mathsf{AnyIdealToIsogeny}$[8, Algorithm 2].
- $\mathsf{PartialRepresentInteger}$ is obtained from $\mathsf{RepresentInteger}$ of [22] by taking the values z, t from input instead of generating them at random, and then choosing $x, y > 0$ (and aborting if $y = 0$).

– ToCanonicalRep computes a change-of-basis matrix from the domain basis of the input HD representation to the canonical basis, then applies it to the evaluation. The matrix is obtained by solving discrete logarithms in the 2^f-torsion group, which is efficient via the Pohlig–Hellman algorithm.

Moreover, a key tool for our construction is the arbitrary ideal-to-isogeny algorithm of [8], which runs in $O(\log(p \cdot \mathrm{Nrd}(I))$ and provides the following interface.

AnyIdealToIsogeny ([8, Algorithm 2]): given an ideal I of maximal orders in $B_{p,\infty}$, corresponding to an isogeny $\varphi_I : E \to E_I$, and a basis P, Q of $E[2^f]$, the algorithm outputs $\varphi_I(P), \varphi_I(Q)$.

Remark 2. While [8, Algorithm 2] assumes the domain of φ_I is a fixed starting curve E_0, we note that the method extends naturally to any domain. One can apply the algorithm twice—once from E_0 to the domain curve, and once from E_0 to the codomain curve. Constructing ideals connecting $\mathcal{O}_0 \cong \mathrm{End}(E_0)$ to any other maximal order is then straightforward.

The rest of this section is dedicated to the more involved algorithms IdealPullBack, HDPushForward, and HalfEndoToCoord which we construct explicitly.

5.1 Fixed Degree Pushforward and Arbitrary Degree Pullback

The function HDPushForward computes the pushforward of an isogeny of odd degree N through another isogeny of degree $2^f - N$. Thanks to the choice of degrees, this can be done efficiently via a direct application of Kani's Lemma. Consequently, the HD representation of the resulting pushforward isogeny can be obtained from a single 2^f-isogeny computation in dimension 2.

Algorithm 1 HDPushForward$(s_{\mathsf{HD}}(\varphi_1), s_{\mathsf{HD}}(\varphi_2))$

Require: $s_{\mathsf{HD}}(\varphi_1)$ and $s_{\mathsf{HD}}(\varphi_2)$, the canonical representations of two isogenies φ_1 and φ_2 of degrees N and $2^f - N$, respectively.
Ensure: $\perp$ or $s_{\mathsf{HD}}([\varphi_2]_*\varphi_1)$
1: Parse $s_{\mathsf{HD}}(\varphi_1), s_{\mathsf{HD}}(\varphi_2)$ as E, P, Q, E_1, P_1, Q_1 and $E', P', Q', E_2, P_2, Q_2$
2: If $E, P, Q \neq E', P', Q'$, return $\perp$
3: Compute $F : E_1 \times E_2 \to E_3 \times E_4$ of kernel generated by (P_1, P_2) and (Q_1, Q_2)
4: If the computation failed in any way or if $j(E_3) \neq j(E)$, return $\perp$
5: Compute $*, P_4 \leftarrow F(0, P_2)$ and $*, Q_4 \leftarrow F(0, Q_2)$
6: **return** $E_2, P_2, Q_2, E_4, P_4, Q_4$

Now, we provide a description of the IdealPullBack algorithm, which takes as input an isogeny ψ of degree N and domain E (given via its HD representation) and an ideal I of arbitrary norm[5] with right order isomorphic to E, and outputs an HD representation of the pullback of ψ through φ_I. The main challenge for this algorithm is that, unlike in Section 5.1, the degree of φ_I is a power of $2^f - N$.

[5] in practice in our scheme the norm given will already be a power of $2^f - N$, but we can get a more generic version very easily thanks to the SpecialKLPT algorithm

Algorithm 2 $\mathsf{IdealPullBack}(I, s^{\mathsf{can}}_{\mathsf{HD}}(\psi))$

Require: An ideal I of maximal orders in $B_{p,\infty}$, and $s^{\mathsf{can}}_{\mathsf{HD}}(\psi)$, the canonical representation of a N-isogeny ψ with domain's endomorphism ring isomorphic to $\mathcal{O}_R(I)$.
Ensure: $\perp$ or $s^{\mathsf{can}}_{\mathsf{HD}}([\varphi_I]^*\psi)$
1: **if** $n(I) = (2^f - N)^\ell$ **then**
2: $K \leftarrow I$, $e \leftarrow \ell$
3: **else**
4: $K \leftarrow \mathsf{SpecialKLPT}_{2^f-N}(I, N)$ of norm $(2^f - N)^e$
5: Parse $s^{\mathsf{can}}_{\mathsf{HD}}(\psi)$ as E, E', P', Q'
6: Compute P, Q (resp. P_0, Q_0) the canonical basis of E (resp. $E_0[2^f]$)
7: $R, S \leftarrow \mathsf{AnyIdealToIsogeny}(K, P_0, Q_0)$
8: Find the matrix M such that $(R, S) = M(P, Q)$
9: $(R', S') \leftarrow M(P', Q')$
10: $s \leftarrow E, R, S, E', R', S'$
11: **for** $i = 1$ to e **do**
12: Parse s as E, P, Q, E', P', Q'
13: Factor K as $K_1 \cdot K_2$ where $\mathrm{Nrd}(K_2) = 2^f - N$
14: $E_1, R_1, S_1 \leftarrow \mathsf{AnyIdealToIsogeny}(K_1, P_0, Q_0)$
15: $s_1 \leftarrow E, P, Q, E_1, [(2^f - N)^i]R_1, [(2^f - N)^i]S_1$
16: $s \leftarrow \mathsf{HDPushForward}(s, s_1)$
17: $K \leftarrow K_1$
18: **return** $\mathsf{ToCanonicalRep}(s)$

In this more general setting, performing the pullback with only a single 2^f-isogeny computation in dimension 2 via Kani's Lemma becomes highly impractical. Two issues arise. First, the norm $\mathrm{Nrd}(I)$ of the ideal may be arbitrarily large, potentially exceeding 2^f. Second, dimension 2 provides very limited flexibility in choosing suitable degrees, even if $\mathrm{Nrd}(I)$ is small enough. To make a single application of Kani's Lemma feasible, $\mathrm{Nrd}(I)$ would have to belong to a very small list of values, which is extremely unlikely.

One way to overcome the second obstacle is to switch from dimension 2 to dimension 4, a now-classical trick in isogeny-based cryptography (see, *e.g.*, [44,18]). Of course dimension 4 is slower than dimension 2, but this slowdown is not prohibitive, as demonstrated by the implementation provided in [17,18]. However, our first issue remains: what if $\mathrm{Nrd}(I)$ is too large? This question is relevant precisely because, in our application, it will indeed be the case.

Our solution is to replace I with another ideal J that is more convenient for computation while still producing the same output. For correctness, we require an ideal J such that $[\varphi_J]^*\psi = [\varphi_I]^*\psi$. As seen in Proposition 1, this holds when $I \sim_{\mathbb{Z}+N\mathcal{O}_L(I)} J$. Equivalently, we need to find $\beta \in I \cap (\mathbb{Z} + N\mathcal{O}_L(I))$ such that $M = \mathrm{Nrd}(\beta)/\mathrm{Nrd}(I)$ allows for efficient pullback computation. The ideal J is then computed as $J = I\bar{\beta}/\mathrm{Nrd}(I)$. The remaining questions are: how do we choose M, and how do we find such a β?

As explained, the main obstacle is the size of the degree. Thus, it is natural to first consider the expected size of the smallest suitable M. A simple covolume computation of the lattice $I \cap (\mathbb{Z} + N\mathcal{O}_L(I))$ shows that the expected size of

the smallest element in this lattice is roughly $\mathrm{Nrd}(I)\sqrt{pN^3}$, since the reduced discriminant of $\mathbb{Z}+N\mathcal{O}_L(I)$ equals pN^3. Thus, the smallest possible M has size approximately $\sqrt{pN^3}$. Unfortunately, in our scheme we have $N \approx p$ yielding $M \approx p^2$ which is far larger than $2^f \approx p$. Hence, we cannot hope to find an ideal J with $\mathrm{Nrd}(J) < 2^f$ that would allow computing the pullback with a single 2^f-isogeny computation in dimension 2 or 4.

Since we cannot hope to perform the pullback in a single step, we split the computation into smaller, manageable parts. In particular, if we can choose $M = (2^f - N)^e$ for some $e \in \mathbb{N}$, we will be able to compute the pullback via e sequential applications of Kani's Lemma. This approach appears to be the most practical. Finding a suitable β and ideal J can be done using a variant of the KLPT algorithm from [22, Appendix E], which we denote by $\mathsf{SpecialKLPT}_{2^f-N}$.[6]

Next, we give a detailed description of the IdealPullBack algorithm, treating SpecialKLPT as a black box. In Section 5.2, we will describe SpecialKLPT in detail and analyze the expected size of the smallest output that can be computed efficiently. For simplicity, we assume the domain of φ_I is a public curve E_0, which holds in the main variant of IdealPullBack used in DINE. The algorithm can be adapted to the generic setting required by the variant of our scheme in Appendix E by providing the domain of φ_I as input and replacing E_0 with it.

In IdealPullBack, for a pair of points P, Q, we define the action of a matrix $M \in \mathbb{Z}^{2\times 2}$ as $R, S := M(P, Q)$ with $R = [M_{1,1}]P + [M_{1,2}]Q$ and $S = [M_{2,1}]P + [M_{2,2}]Q$. This notation will be reused later in Algorithm 5.

Proposition 3. *(Proof in Appendix H) Under plausible heuristic assumptions, the* IdealPullBack *algorithm is correct and runs in time* $\mathsf{poly}(\log(pN\,\mathrm{Nrd}(I)))$.

5.2 The Special KLPT Algorithm

In this section, we describe the algorithm $\mathsf{SpecialKLPT}_M$, which takes as input an $\mathcal{O}_0$-ideal I and an integer N, and outputs an element $\beta \in I \cap (\mathbb{Z}+N\mathcal{O}_0)$ of norm M^ℓ with ℓ as small as possible. Recall that $\mathcal{O}_0$ is a *special extremal order*, which enables the use of the most efficient KLPT variants (see [33,22,36]). Throughout this section, $\mathcal{O}_0$ is treated as an implicit parameter of the algorithms.

A heuristic algorithm for this task was given in [22, Appendix E], yielding $\ell \log M \approx 4\log p + 5\log N + 2\log(\mathrm{Nrd}(I))$. Without loss of generality, we may assume I has minimal norm in the equivalence class of $I \cap (\mathbb{Z} + N\mathcal{O}_0)$. By the gaussian heuristic, $\log(\mathrm{Nrd}(I)) \approx 1/2\log(p)+3/2\log(N)$, so the expected output size is about $5\log(p) + 8\log(N)$. We now introduce a new heuristic algorithm achieving the same goal, but with expected output size $\approx 3\log(p) + 7\log N$. It has the same generic outline as all the KLPT algorithms:

1. Possibly replace I by a better suited ideal K with $I := K\delta$.
2. Generate $\gamma \in \mathcal{O}_0$ of norm $\mathrm{Nrd}(K)M^{e_0}$ in a suitable suborder of $\mathcal{O}_0$.
3. Find μ_0 of arbitrary norm with $\gamma\mu_0 \in K$ and $\gamma\mu_0\delta$ in the target order.
4. Compute $\mu = \lambda\mu_0 + N\,\mathrm{Nrd}(K)\mu_1$ of norm M^{e_1}, and output $\beta = \gamma\mu\delta$.

[6] the version presented here is an improved version of the one outlined in [22]

Algorithm 3 SpecialRepresentInteger(M, N)

Require: Two integers N, M pairwise coprime and both coprime to p.
Ensure: $\perp$ or $\gamma \in \mathbb{Z}\langle N, Ni, j, k\rangle$ of norm M.

found $\leftarrow \perp$; $x, y, z, t \leftarrow 0, 0, 0, 0$
$c \leftarrow M/N^2 \mod p$
$B \leftarrow M - cN^2$
while found $= \perp$ and $B > 0$ **do**
  if Cornacchia$(c) \neq \perp$ **then**
    $x, y \leftarrow$ Cornacchia(c)
    $s \leftarrow B/p$
    if Cornacchia$(s) \neq \perp$ **then**
      $z, t \leftarrow$ Cornacchia(s)
      found $= \top$
  $B = B - pN^2$
  $c = c + p$
if found $= \perp$ **then return** $\perp$
return $N(x + iy) + jz + kt$

In the algorithm from [22], one takes $K = I$ and $\gamma \in \mathbb{Z}\langle 1, Ni, Nj, Nk\rangle$. Our improvement is to instead choose K as the ideal of smallest prime norm in the equivalence class of $I \cap \mathbb{Z}\langle 1, i, Nj, Nk\rangle$ and $\gamma \in \mathbb{Z}\langle N, Ni, j, k\rangle$. These choices yield K and γ of smaller norms, thereby reducing the exponent ℓ. Correctness hinges on the existence of a solution μ_0 in Step 3, which is established in Lemma 1.

The precise description follows later. We first introduce the necessary building blocks, some classical and others new. We begin with the classical ones.

- Cornacchia: solves $x^2 + y^2 = m$, returning $\perp$ or integers x, y (see, *e.g.*, [36, Algorithm 1]); for primes $m = 1 \bmod 4$, the output is never $\perp$.
- IdealModConstraint: given an ideal K and γ, finds C, D with $\gamma j(C + iD) \in K$.
- StrongApproximation$_{l^\bullet}$: on input integers M, C, D, produces $\mu = \lambda j(C + iD) + M\mu_1$ with $\lambda \in \mathbb{Z}, \mu_1 \in \mathcal{O}_0$, ensuring $\mathrm{Nrd}(\mu) = l^e$ for some e. The factorization of M is assumed known (see, *e.g.*, [36, Algorithm 3]).

Special represent integer. To compute an element γ of prescribed norm in the order $\mathbb{Z}\langle N, Ni, j, k\rangle$, we adapt the classical RepresentInteger algorithm (see, *e.g.*, [36, Algorithm 2]), and denote this variant by SpecialRepresentInteger.

The main change is that, since we seek a solution of the form (Nx, Ny, z, t), the coordinates z, t cannot be chosen at random as in RepresentInteger. Instead, M and $z^2 + t^2$ must satisfy a congruence modulo N^2, so once $z^2 + t^2$ is chosen, the values of z, t are computed using Cornacchia's algorithm.

The algorithm can also start by choosing x, y first, in which case the constraint is on $x^2 + y^2 \bmod p$. Since in our application $p \ll N^2$, selecting x, y before z, t will be more efficient.

Proposition 4. *Whenever* SpecialRepresentInteger *does not output* $\perp$*, its output is correct. Under plausible heuristic assumptions, for a security parameter* λ*, if* $M > 5\lambda \log(p) \log(N) pN^2$*, then the probability that* SpecialRepresentInteger *fails on input* N, M *is* $\mathsf{negl}(\lambda)$.

Proof. Since Cornacchia never outputs $\perp$ when its input is a prime congruent to 1 mod 4, its failure probability on a random input smaller than some bound B is at most $C/(2\log B)$ for some small constant C. Under the plausible assumption that the inputs to Cornacchia within SpecialRepresentInteger behave like random integers of the same size, we can lower-bound the success probability of the while loop for a value c by $1/(4\log(c)\log((M-cN^2)/p))$. Since the first c is smaller than p, and the algorithm runs in time polynomial in $\log p$, we have $\log(c) < (1+\varepsilon)\log p$ for some $\varepsilon > 0$. Therefore, the success probability is lower-bounded by $1/(2(1+\varepsilon)\log(p)\log(M/p))$.

Thus, if $M > N^{2+\varepsilon}p$, the success probability is lower-bounded by $1/(2(1+\varepsilon)(2+\varepsilon)\log(p)\log(N))$. To achieve overwhelming success probability with respect to a parameter λ, it suffices to try roughly $2\lambda(1+\varepsilon)(2+\varepsilon)\log(p)\log(N)$ candidates. Therefore, for $M > 5\lambda\log(p)\log(N)pN^2$, the failure probability of the SpecialRepresentInteger algorithm becomes $\mathsf{negl}(\lambda)$, as claimed. □

Projecting γ to $\mathbb{Z}+N\mathcal{O}_0$. We now focus on Step 3 of SpecialKLPT. The key point is to ensure the existence of a solution μ_0, as established by Lemma 1. If it exists, it can be found efficiently by linear algebra. We denote the resulting algorithm by NonGorensteinModConstraint, which is closely related to IdealModConstraint and EichlerModConstraint (see, *e.g.*, [36, Section 3.2.2]). A full description is omitted.

Lemma 1. *Let $\gamma \in \mathbb{Z}\langle N, Ni, j, k\rangle$ and $\delta \in \mathbb{Z}\langle 1, i, Nj, Nk\rangle$. Then there exists a nonzero element $\mu_0 \in \mathbb{Z}/N\mathbb{Z}\langle j, k\rangle$ such that $\gamma\mu_0\delta \in \mathbb{Z}\langle 1, Ni, Nj, Nk\rangle$.*

Proof. Write $\gamma = N(a+ib)+j(c+id)$, $\mu_0 = j(C+iD)$ and $\delta = x+iy+Nj(z+it)$. Modulo N, the product $\gamma\mu_0\delta$ simplifies to $-p(C+iD)(c-id)(x+iy)$. For this product to lie in $\mathbb{Z}\langle 1, Ni, Nj, Nk\rangle$, it suffices that the coefficient of i is 0 mod N. This coefficient is $-p[C(cy-dx)+D(cx+dy)]$ and there are always non zero solutions C, D to this equation mod N. □

The proof of Lemma 1 implicitly provides a simple method to compute μ_0 from N, γ, and δ. We denote this algorithm by NonGorensteinModConstraint(γ, δ, N) in the description of Algorithm 4.

Equivalent small ideal. The final building block for SpecialKLPT is an algorithm to replace the input ideal I with an equivalent ideal K of smaller norm. By Lem.1, not all ideals suffice: K must be equivalent to I as $\mathbb{Z}\langle 1, i, Nj, Nk\rangle$ ideals. This ensures that the element δ satisfying $I = K\delta/\operatorname{Nrd}(K)$ lies in $\mathbb{Z}\langle 1, i, Nj, Nk\rangle$).

We denote this algorithm by EquivalentSpecialPrimeIdeal. It is similar to the EquivalentPrimeIdeal algorithm ([36, Algorithm 6]). It consists in computing a reduced basis of the lattice $I \cap \mathbb{Z}\langle 1, i, Nj, Nk\rangle$ and enumerating short vectors until finding one $\hat{\delta}$ of norm $\operatorname{Nrd}(I)N'$ for some prime N'. The ideal $K = I\hat{\delta}/\operatorname{Nrd}(I)$ of norm N' is then returned. We omit a full description of the algorithm.

Detailed description of SpecialKLPT. We now give a full description of our heuristic algorithm. We also introduce a constant, `boundSRI`, which is used to select the exponent in the input to SpecialRepresentInteger to ensure overwhelming success probability. We do not provide a full proof of correctness for SpecialKLPT.

Algorithm 4 $\mathsf{SpecialKLPT}_{M^\bullet}(I, N)$

Require: $\mathcal{O}_0$-ideal I and integer N such that $N, \mathrm{Nrd}(I), p$ are all pairwise coprime.
Ensure: $\perp$ or $J \simeq_{\mathbb{Z}+N\mathcal{O}_0} I$ of norm M^e for some $e \in \mathbb{N}$.

$K \leftarrow \mathsf{EquivalentSpecialPrimeIdeal}(I, N)$
Let δ be the element such that $I = K\delta / \mathrm{Nrd}(K)$
Select $M^{e_0} > pN^2 \texttt{boundSRI} / \mathrm{Nrd}(K)$
$\mu \leftarrow \perp$, $\gamma \leftarrow 0$
while $\mu = \perp$ and $\gamma \neq \perp$ **do**
  $\gamma \leftarrow \mathsf{SpecialRepresentInteger}(\mathrm{Nrd}(K)M^{e_0}, N)$
  if $\gamma \neq \perp$ **then**
    $C_N, D_N \leftarrow \mathsf{NonGorensteinModConstraint}(\gamma, \delta, N)$
    $C_K, D_K \leftarrow \mathsf{IdealModConstraint}(K, \gamma)$
    $C \leftarrow \mathsf{CRT}_{N,\mathrm{Nrd}(K)}(C_N, C_K)$, $D \leftarrow \mathsf{CRT}_{N,\mathrm{Nrd}(K)}(D_N, D_K)$
    $\mu \leftarrow \mathsf{StrongApproximation}_{M^\bullet}(N\,\mathrm{Nrd}(K), C, D)$
if $\mu \neq \perp$ **then**
  $\beta \leftarrow \gamma\mu\delta / \mathrm{Nrd}(K)$
  return $J \leftarrow I\bar{\beta} / \mathrm{Nrd}(I)$
else return $\perp$

It follows directly from Lemma 1 and the standard proof of the KLPT algorithm (see [36, Prop. 3.2.7]). The algorithm terminates with overwhelming probability provided `boundSRI` is chosen large enough so that Prop. 4 can be applied.

Regarding the output size, the usual estimate for $\mathsf{StrongApproximation}(x, \cdot, \cdot)$ (see, *e.g.*, [36, Lemma 3.1.6]) gives $\mathrm{Nrd}(\mu) \approx p(N\,\mathrm{Nrd}(K))^3$. We also have $\mathrm{Nrd}(\gamma) \approx pN^2/\mathrm{Nrd}(K)$, ignoring constants and $\log\log$ factors from `boundSRI`. Hence, the final output has norm $\approx p^2N^5\,\mathrm{Nrd}(K)^2$. By the gaussian heuristic, $\mathrm{Nrd}(K)$ is expected to be $\approx \sqrt{p}N$ (as the reduced discriminant of $\mathbb{Z}\langle N, Ni, j, k\rangle$ is $4pN^2$), so the generic expected output norm of $\mathsf{SpecialKLPT}$ is around p^3N^7.

5.3 Recovering the Coefficients of an Endomorphism from an HD Representation

In this section, we introduce the $\mathsf{HalfEndoToCoord}$ algorithm, which recovers the coordinates of an endomorphism $\theta = \hat{\varphi}_2 \circ \varphi_1 \in \mathrm{End}(E_0)$ of norm N^2 in the basis $1, i, j, k$, given the HD representations of the isogenies φ_1, φ_2.

The idea is to recover these coefficients by computing the four integers $\mathrm{Trd}(\theta)$, $\mathrm{Trd}(\theta i)$, $\mathrm{Trd}(\theta j)$, and $\mathrm{Trd}(\theta k)$ modulo 2^f. Indeed, if $\theta = x + iy + jz + kt$, then $\mathrm{Trd}(\theta) = 2x$, $\mathrm{Trd}(\theta i) = -2y$, $\mathrm{Trd}(\theta j) = -2pz$, and $\mathrm{Trd}(\theta k) = -2pt$. If x, y, z, t are small enough, these traces modulo 2^f determine the coefficients over $\mathbb{Z}$, which suffices for our purpose. Modulo 2^f, the trace of an endomorphism θ (which can be evaluated on the 2^f-torsion) is easily computed as $\mathrm{Trd}(\theta) = \theta + \bar{\theta}$.

Proposition 5. *Let $\theta = \hat{\varphi}_2 \circ \varphi_1$ be an endomorphism of norm N^2 corresponding to the quaternion $x+iy+jz+kt \in \mathbb{Z}^+\langle 1, i, j, k\rangle$, where $(x, y, z, t) \in [0, 2^{f-1}[$. Then* $\mathsf{HalfEndoToCoord}$ *successfully computes x, y, z, t on input $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2)$.*

Algorithm 5 $\mathsf{HalfEndoToCoord}(s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2))$

Require: Canonical HD representations $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2)$ of two N-isogenies φ_1, φ_2 with the same domain and isomorphic codomain.

Ensure: $\perp$ if the input is invalid; otherwise integers x, y, z, t such that the endomorphism $\theta = \hat{\varphi}_2 \circ \rho \circ \varphi_1$ corresponds to $x + iy + jz + kt$, where ρ is the isomorphism from the codomain of φ_1 to that of φ_2.

Compute P_0, Q_0, the canonical basis of $E_0[2^f]$
for $\alpha \in \{\iota, \pi, \iota \circ \pi\}$, compute the matrices M_α s.t. $\alpha(P_0, Q_0) = M_\alpha(P_0, Q_0)$
Parse $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_i)$ as E_0, E_i, P_i, Q_i for $i = 1, 2$
if $j(E_1) \neq j(E_2)$, **return** $\perp$
Compute the isomorphism $\rho : E_1 \to E_2$
Find the matrix M s.t. $M(P_2, Q_2) = \rho(P_1, Q_1)M$
Find the integer $0 \leq x \leq 2^{f-1}$ s.t. $2x = N(M_{0,0} + M_{1,1}) \bmod 2^f$
Find the integer $0 \leq y \leq 2^{f-1}$ s.t. $2y = -N(M_{0,0}M_{\iota,0,0} + M_{0,1}M_{\iota,1,0} + M_{0,1}M_{\iota,1,0} + M_{1,1}M_{\iota,1,1}) \bmod 2^f$
$\kappa \leftarrow p^{-1} \bmod 2^f$
Find the integer $0 \leq z \leq 2^{f-1}$ s.t. $2z = -\kappa N(M_{0,0}M_{\pi,0,0} + M_{0,1}M_{\pi,1,0} + M_{0,1}M_{\pi,1,0} + M_{1,1}M_{\pi,1,1}) \bmod 2^f$
Find the integer $0 \leq t \leq 2^f$ s.t. $2t = -\kappa N(M_{0,0}M_{\iota\circ\pi,0,0} + M_{0,1}M_{\iota\circ\pi,1,0} + M_{0,1}M_{\iota\circ\pi,1,0} + M_{1,1}M_{\iota\circ\pi,1,1}) \bmod 2^f$
if $y = 0$, **return** $\perp$
$\varepsilon \leftarrow y/\left|y\right|$
return $\varepsilon x, \varepsilon y, \varepsilon z, \varepsilon t$

Proof. For $\theta = x+iy+jz+kt \in \mathbb{Z}\langle 1, i, j, k\rangle$, we have $\mathrm{Trd}(\theta) = 2x$, $\mathrm{Trd}(\theta i) = -2y$, $\mathrm{Trd}(\theta j) = -2pz$ and $\mathrm{Trd}(\theta k) = -2pt$. Thus, if the values $2x, 2y, 2z, 2t$ computed by $\mathsf{HalfEndoToCoord}$ correspond modulo 2^f to the traces of $\theta, \theta i, \theta j/p, \theta k/p$, then the output is correct. Indeed, since $x, y, z, t \in [0, 2^{f-1}[$, their doubles lie in $[0, 2^f[$. By the definition of the canonical HD representation, the points P_i, Q_i are equal to $\varphi_i(P_0, Q_0)$. If the matrix M satisfies $M(P_2, Q_2) = \rho(P_1, Q_1)$, then $M\varphi_2(P_0, Q_0) = \rho \circ \varphi_1((P_0, Q_0))$ and thus $[N]M(P_0, Q_0) = \theta((P_0, Q_0))$. Consequently, the matrix representing the quaternion θi in the basis P_0, Q_0 is NMM_ι, whose trace is $N(M_{0,0}M_{\iota,0,0} + M_{0,1}M_{\iota,1,0} + M_{0,1}M_{\iota,1,0} + M_{1,1}M_{\iota,1,1})$. Applying the same reasoning to j, k yields the desired result. □

6 Instantiation of the Scheme

In this section, we describe a concrete instantiation of DINE_ℓ with parameter choices targeting NIST security level 1. We also provide a C implementation based on Qlapoty [24] (https://github.com/qlapoty/Qlapoty).

Parameters. Following [51], we use the conservative estimate of $p^{1/3}$ for the best attack against the endomorphism ring problem. This neglects the high memory cost, and the $p^{o(1)}$ factors in the attack from [51], but this is sufficient for our proof of concept. We use the prime characteristic $p = 65 \cdot 2^{376} - 1$ to enable efficient 2^f-isogeny computation in dimension 2. Following the less conservative

estimates provided in [1, Chapter 8], we could consider the prime characteristic $32^{324} - 1$ for better performances.

We choose $N = 2^{375} - 139029$, the largest prime below 2^{375} satisfying all necessary constraints. We search for primes below 2^{375} so that $2^{376} - N$ is not too small. One can verify that N is prime and that p has maximal order in $\mathbb{F}_N^\times$. For good measure, we also chose N such that $2^{376} - N$ is prime, making the computation of $(2^f - N)$-isogenies as hard as possible and, hopefully, making Assumption 1 as hard as possible.

We can now justify the sizes of the various elements in DINE_ℓ. We can encrypt messages of size $m(\lambda) = 8 \cdot 28 = 224$ bits (28 bytes), the largest multiple of 8 that allows us to choose $n(\lambda) > 128$ while leaving a large enough randomization space for PartialRepresentInteger to succeed with overwhelming probability. As with SpecialRepresentInteger, the failure probability of PartialRepresentInteger is $O(\log p)$, so $n(\lambda)$ must be large enough to make encryption failures negligible. For $m(\lambda) = 224$, $n(\lambda) \geq 11$ appears sufficient.

Ciphertexts consist of two curves (the codomains of the two ciphertext isogenies) and two bases (the images of the canonical basis of the domain under the ciphertext isogenies). Tokens consist of ℓ times: two curves (the domain and codomain of the token isogeny) and one basis (the image of the canonical basis of the domain under the token isogeny). In the Montgomery model, each elliptic curve can be represented by a single element over $\mathbb{F}_{p^2}$ (*e.g.*, the coefficient A). Using x-coordinates, a basis can be represented by 3 elements over $\mathbb{F}_{p^2}$. For a 2^f-basis, this representation can be compressed to 3 elements of f bits, roughly halving its size. This is a standard technique [8,19] that relies on pairings to convert between the full and compressed representations. In our setting, encoding elements of $\mathbb{F}_p$ (and f-bit elements) using 48 bytes gives compressed sizes of 480B for ciphertexts and $336 \times \ell$B for tokens, resulting in $1008, 1344$ and 2688B for $\ell = 3, 4, 8$, respectively.

Implementation Details. To improve the efficiency of our scheme, certain values can be precomputed during key generation or key update and stored as part of the secret key. In particular, the chain of 2^f-isogenies used in IdealPullBack during decryption is identical for ciphertexts encrypted with the same key. Thus, this chain can be computed once and stored. Similarly, the evaluation of the secret key ideal I_e on the canonical basis of E_0 can be precomputed to avoid calling AnyIdealToIsogeny on K for computing the initial R, S in IdealPullBack. These precomputations eliminate all calls to AnyIdealToIsogeny in IdealPullBack during decryption, substantially speeding up the process, as shown by the timings reported in Table 2.

Results. Experimental results from our C implementation are reported in Table 2. KeyGen is substantially slower and exhibits significant variability in practice, mainly due to occasional failures in SpecialKLPT. Although SpecialKLPT could likely be optimized, we did not explore this, as key generation time is not critical in our setting.

References

1. Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Giacomo Borin, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Thomas Espitau, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, Riccardo Invernizzi, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Marzio Mula, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, Frederik Vercauteren, Alexandre Wallet, Benjamin Wesolowski, and Wessel van Woerden, *SQIsign*, Tech. report, National Institute of Standards and Technology, 2026.
2. Marius A Aardal, Andrea Basso, Luca De Feo, Sikhar Patranabis, and Benjamin Wesolowski, *A complete security proof of SQIsign*, Annual International Cryptology Conference, Springer, 2025, pp. 190–222.
3. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis, *Cryptographic Group Actions and Applications*, Advances in Cryptology – ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II (Berlin, Heidelberg), Springer-Verlag, 2020, p. 411–439.
4. Bill Allombert, Jean-François Biasse, Jonathan Komada Eriksen, Péter Kutas, Chris Leonardi, Aurel Page, Renate Scheidler, and Márton Tot Bagi, *Pearl-scallop: Parameter extension applicable in real-life scallop*, Cryptology ePrint Archive (2024).
5. Karim Baghery, Daniele Cozzo, and Robi Pedersen, *An Isogeny-Based ID Protocol Using Structured Public Keys*, Cryptography and Coding (Cham) (Maura B. Paterson, ed.), Springer International Publishing, 2021, pp. 179–197.
6. Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski, *Prism: Simple and compact identification and signatures from large prime degree isogenies*, IACR International Conference on Public-Key Cryptography, Springer, 2025, pp. 300–332.
7. Andrea Basso, Giacomo Borin, Ryan Rueger, and Sina Schaeffler, *Coral faster isogeny group action for post-quantum nike*, Annual International Cryptology Conference, Springer, 2026, pp. 506–536.
8. Andrea Basso, Pierrick Dartois, Luca De Feo, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski, *SQIsign2D–West: The Fast, the Small, and the Safer*, Advances in Cryptology – ASIACRYPT 2024: 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings, Part III (Berlin, Heidelberg), Springer-Verlag, 2024, p. 339–370.
9. Dan Boneh, Kevin Lewi, Hart Montgomery, and Ananth Raghunathan, *Key Homomorphic PRFs and Their Applications*, Advances in Cryptology – CRYPTO 2013 (Berlin, Heidelberg) (Ran Canetti and Juan A. Garay, eds.), Springer Berlin Heidelberg, 2013, pp. 410–428.
10. Colin Boyd, Gareth T. Davies, Kristian Gjøsteen, and Yao Jiang, *Fast and Secure Updatable Encryption*, Advances in Cryptology – CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part I, Springer-Verlag, 2020, p. 464–493.

11. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes, *CSIDH: an efficient post-quantum commutative group action*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2018, pp. 395–427.
12. Wouter Castryck, Jana Sotáková, and Frederik Vercauteren, *Breaking the decisional Diffie-Hellman problem for class group actions using genus theory*, Annual International Cryptology Conference, Springer, 2020, pp. 92–120.
13. Mingjie Chen, Muhammad Imran, Gábor Ivanyos, Péter Kutas, Antonin Leroux, and Christophe Petit, *Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of psidh*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2023, pp. 99–130.
14. Mingjie Chen, Antonin Leroux, and Lorenz Panny, *SCALLOP-HD: Group Action from 2-Dimensional Isogenies*, Public-Key Cryptography – PKC 2024 (Cham) (Qiang Tang and Vanessa Teague, eds.), Springer Nature Switzerland, 2024, pp. 190–216.
15. Mingjie Chen and Christophe Petit, *Computing the endomorphism ring of a supersingular elliptic curve from a full rank suborder*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2025, pp. 446–474.
16. Giulio Codogni and Guido Maria Lido, *Spectral theory of isogeny graphs*, Journal of Number Theory **286** (2026), 131–184.
17. Pierrick Dartois, *Fast computation of 2-isogenies in dimension 4 and cryptographic applications*, Journal of Algebra **683** (2025), 449–514.
18. Pierrick Dartois, Jonathan Komada Eriksen, Tako Boris Fouotsa, Arthur Herlédan Le Merdy, Riccardo Invernizzi, Damien Robert, Ryan Rueger, Frederik Vercauteren, and Benjamin Wesolowski, *Pegasis: Practical effective class group action using 4-dimensional isogenies*, Annual International Cryptology Conference, Springer, 2025, pp. 67–99.
19. Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski, *Sqisignhd: new dimensions in cryptography*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2024, pp. 3–32.
20. Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert, *An Algorithmic Approach to (2,2)-Isogenies in the Theta Model and Applications to Isogeny-Based Cryptography*, Advances in Cryptology – ASIACRYPT 2024 (Singapore) (Kai-Min Chung and Yu Sasaki, eds.), Springer Nature Singapore, 2025, pp. 304–338.
21. Luca De Feo, Tako Boris Fouotsa, and Lorenz Panny, *Isogeny problems with level structure*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2024, pp. 181–204.
22. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski, *SQISign: compact post-quantum signatures from quaternions and isogenies*, Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26, Springer, 2020, pp. 64–93.
23. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel, *Generic Models for Group Actions*, Public-Key Cryptography – PKC 2023: 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7–10, 2023, Proceedings, Part I (Berlin, Heidelberg), Springer-Verlag, 2023, p. 406–435.

24. Max Duparc, Antonin Leroux, and Sina Schaeffler, *Qlapoty: Improved analysis and efficiency for quaternionic ideal to isogeny transformation*, Cryptology ePrint Archive (2026).
25. Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski, *Scallop: Scaling the csi-fish*, Public-Key Cryptography – PKC 2023 (Cham) (Alexandra Boldyreva and Vladimir Kolesnikov, eds.), Springer Nature Switzerland, 2023, pp. 345–375.
26. Joseph H. Silverman, *The arithmetic of elliptic curves*, vol. 106, 01 2009.
27. Marc Houben, *Efficient post-quantum commutative group actions from orientations of large discriminant*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2025, pp. 141–173.
28. David Jao and Luca De Feo, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, Post-Quantum Cryptography (Berlin, Heidelberg) (Bo-Yin Yang, ed.), Springer Berlin Heidelberg, 2011, pp. 19–34.
29. Yao Jiang, *The Direction of Updatable Encryption Does Not Matter Much*, Advances in Cryptology – ASIACRYPT 2020 (Cham) (Shiho Moriai and Huaxiong Wang, eds.), Springer International Publishing, 2020, pp. 529–558.
30. Yao Jiang Galteland and Jiaxin Pan, *Backward-Leak Uni-Directional Updatable Encryption from (Homomorphic) Public Key Encryption*, Public-Key Cryptography – PKC 2023: 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7–10, 2023, Proceedings, Part II (Berlin, Heidelberg), Springer-Verlag, 2023, p. 399–428.
31. Ernst Kani, *The number of curves of genus two with elliptic differentials*, Journal für die reine und angewandte Mathematik **485** (1997), 93–122.
32. Michael Klooß, Anja Lehmann, and Andy Rupp, *(R)CCA Secure Updatable Encryption with Integrity Protection*, Advances in Cryptology – EUROCRYPT 2019 (Cham) (Y. Ishai and V. Rijmen, eds.), Springer International Publishing, 2019, pp. 68–99.
33. David Kohel, Kristin E. Lauter, Christophe Petit, and Jean-Pierre Tignol, *On the quaternion ℓ-isogeny path problem*, IACR Cryptology ePrint Archive **2014** (2014), 505.
34. Anja Lehmann and Björn Tackmann, *Updatable Encryption with Post-Compromise Security*, Advances in Cryptology – EUROCRYPT 2018 (Cham) (J. B. Nielsen and V. Rijmen, eds.), Springer International Publishing, 2018, pp. 685–716.
35. Antonin Leroux, *A new isogeny representation and applications to cryptography*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2022, pp. 3–35.
36. ———, *Quaternion algebra and isogeny-based cryptography*, Ph.D. thesis, PhD thesis, Ecole doctorale de l'Institut Polytechnique de Paris, 2022.
37. ———, *Verifiable random function from the deuring correspondence and higher dimensional isogenies*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2025, pp. 167–194.
38. Antonin Leroux and Maxime Roméas, *Updatable Encryption from Group Actions*, Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12–14, 2024, Proceedings, Part II (Berlin, Heidelberg), Springer-Verlag, 2024, p. 20–53.
39. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski, *A direct key recovery attack on sidh*, EUROCRYPT, 2023.
40. Luciano Maino and Christophe Petit, *Private communication*, 2026.

41. Jonas Meers and Doreen Riepel, *CCA Secure Updatable Encryption from Non-mappable Group Actions*, Post-Quantum Cryptography: 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12–14, 2024, Proceedings, Part I (Berlin, Heidelberg), Springer-Verlag, 2024, p. 137–169.
42. T. Moriya, H. Onuki, and T. Takagi, *SiGamal: a supersingular isogeny-based PKE and its application to a PRF*, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2020, pp. 551–580.
43. Ryo Nishimaki, *The Direction of Updatable Encryption Does Matter*, Public-Key Cryptography – PKC 2022: 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8–11, 2022, Proceedings, Part II (Berlin, Heidelberg), Springer-Verlag, 2022, p. 194–224.
44. Aurel Page and Damien Robert, *Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time*, Cryptology ePrint Archive, Paper 2023/1766, 2023.
45. Arnold Pizer, *An algorithm for computing modular forms on $\Gamma_0(N)$*, Journal of Algebra **64** (1980), no. 2, 340–390.
46. Damien Robert, *Breaking sidh in polynomial time*, Advances in Cryptology – EUROCRYPT 2023 (Cham) (Carmit Hazay and Martijn Stam, eds.), Springer Nature Switzerland, 2023, pp. 472–503.
47. Daniel Slamanig and Christoph Striecks, *Revisiting Updatable Encryption: Controlled Forward Security, Constructions and a Puncturable Perspective*, Theory of Cryptography: 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 – December 2, 2023, Proceedings, Part II (Berlin, Heidelberg), Springer-Verlag, 2023, p. 220–250.
48. J. Vélu, *Isogénies entre courbes elliptiques*, Comptes-Rendus de l'Académie des Sciences, Série I **273** (1971), 238–241.
49. John Voight, *Quaternion algebras*, Springer International Publishing, Cham, 2021.
50. Benjamin Wesolowski, *The supersingular isogeny path and endomorphism ring problems are equivalent*, 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), 2022, pp. 1100–1111.
51. Benjamin Wesolowski, *The supersingular isogeny problem in time and memory $p^{1/3+o(1)}$*, Cryptology ePrint Archive, Paper 2026/1486, 2026.

A Background on Updatable Encryption

Trivial wins via keys and ciphertexts. We define extended epoch leakage sets $\mathcal{C}^*$, $\mathcal{K}^*$, and $\mathcal{T}^*$ derived from $\mathcal{C}$, $\mathcal{K}$, and $\mathcal{T}$. These extended sets capture situations where the adversary obtains both a challenge ciphertext update and the means to decrypt it. In particular, if $\mathcal{C}^* \cap \mathcal{K}^* \neq \emptyset$, then there exists an epoch in which the adversary knows both the epoch key and a valid update of the challenge ciphertext, resulting in a trivial win. The challenger computes these extended sets after the adversary halts, with precise definitions given in Appendix A.

Trivial wins via direct updates. We define $\mathcal{I}$ as the set of epochs in which the adversary learned an updated version of the challenge input ciphertext $\bar{C}$. Its extension $\mathcal{I}^*$ also accounts for information inferred through token corruption.

Since Upd is deterministic, an updated ciphertext is uniquely determined by a token and a ciphertext. Hence, if $\mathcal{I}^* \cap \mathcal{C}^* \neq \emptyset$, the adversary trivially wins:

there exists an epoch in which the adversary knows both an updated version of $\bar{C}$ and a challenge-equal ciphertext, and comparing them suffices to win.

Trivial wins in ciphertext integrity games. An adversary who corrupts an epoch key can trivially forge ciphertexts in that epoch. We exclude this case by setting $\mathsf{twf} = 1$ whenever a forgery is attempted in an epoch contained in $\mathcal{K}^*$.

Another trivial win arises if the adversary knows a ciphertext $(C, \mathsf{e}_1) \in \mathcal{L}$ together with all tokens from epoch $\mathsf{e}_1 + 1$ up to some epoch e_2. In this case, it can simply update C to epoch e_2 and present it as a valid forgery. To rule this out, we extend $\mathcal{L}$ to a set $\mathcal{L}^*$ that also accounts for ciphertexts derived via token corruption. If $\mathcal{O}.\mathsf{Try}$ receives a ciphertext from $\mathcal{L}^*$, it sets $\mathsf{twf} = 1$. The algorithm of [10] used to compute $\mathcal{L}^*$ during the game is recalled in Fig. 7 of Appendix A.

Following [34], extended epoch leakage sets $\mathcal{C}^*$, $\mathcal{K}^*$ and $\mathcal{T}^*$ are computed as follows:

$$
\begin{aligned}
\mathcal{K}^* &\leftarrow \{\mathsf{e} \in \{1,\dots,n\} \mid \mathsf{CorrK}(\mathsf{e}) = \mathsf{true}\} \\
&\quad \mathsf{true} \leftarrow \mathsf{CorrK}(\mathsf{e}) \Leftrightarrow (\mathsf{e} \in \mathcal{K}) \vee (\mathsf{CorrK}(\mathsf{e}-1) \wedge \mathsf{e} \in \mathcal{T}) \vee (\mathsf{CorrK}(\mathsf{e}+1) \wedge \mathsf{e}+1 \in \mathcal{T}) \\
\mathcal{T}^* &\leftarrow \{\mathsf{e} \in \{1,\dots,n\} \mid (\mathsf{e} \in \mathcal{T}) \vee (\mathsf{e} \in \mathcal{K}^* \wedge \mathsf{e}-1 \in \mathcal{K}^*)\} \\
\mathcal{C}^* &\leftarrow \{\mathsf{e} \in \{1,\dots,n\} \mid \mathsf{ChallEq}(\mathsf{e}) = \mathsf{true}\} \\
&\quad \mathsf{true} \leftarrow \mathsf{ChallEq}(\mathsf{e}) \Leftrightarrow (\mathsf{e} = \tilde{\mathsf{e}}) \vee (\mathsf{e} \in \mathcal{C}) \vee (\mathsf{ChallEq}(\mathsf{e}-1) \wedge \mathsf{e} \in \mathcal{T}^*) \vee \\
&\qquad\qquad (\mathsf{ChallEq}(\mathsf{e}+1) \wedge \mathsf{e}+1 \in \mathcal{T}^*)
\end{aligned}
$$

Likewise, we extend $\mathcal{I}$ into $\mathcal{I}^*$:

$$
\begin{aligned}
\mathcal{I}^* &\leftarrow \{\mathsf{e} \in \{1,\dots,n\} \mid \mathsf{ChallInpEq}(\mathsf{e}) = \mathsf{true}\} \\
&\quad \mathsf{true} \leftarrow \mathsf{ChallInpEq}(\mathsf{e}) \Leftrightarrow (\mathsf{e} \in \mathcal{I}) \vee (\mathsf{ChallInpEq}(\mathsf{e}-1) \wedge \mathsf{e} \in \mathcal{T}^*) \vee \\
&\qquad\qquad (\mathsf{ChallInpEq}(\mathsf{e}+1) \wedge \mathsf{e}+1 \in \mathcal{T}^*)
\end{aligned}
$$

Moreover, Fig. 7 shows how to extend $\mathcal{L}$ into $\mathcal{L}^*$.

Update $\mathcal{L}^*$

if $\mathcal{O}.\mathsf{Enc}$ **or** $\mathcal{O}.\mathsf{Upd}$ happens
  $\mathcal{L}^* \leftarrow \mathcal{L}^* \cup \{(\cdot, C, \cdot)\}$
if $\mathcal{O}.\mathsf{Corr}(\mathsf{token}, \cdot)$ happens
  for $i \in \mathcal{T}^*$:
    for $(j, C_{i-1}, i-1) \in \mathcal{L}^*$:
      $C_i \leftarrow \mathsf{UE.Upd}(\Delta_i, C_{i-1})$
      $\mathcal{L}^* \leftarrow \mathcal{L}^* \cup \{(j, C_i, i)\}$

Fig. 7. Update procedure of [10] for list $\mathcal{L}^*$.

Composition result for CPA, CTXT and CCA security. We conclude this section by recalling a generic composition result of Boyd *et al.* [10, Theorem 3], which relates CPA security and ciphertext integrity to CCA security for UE schemes.

Theorem 4 ([10]). *For any* detIND-UE-CCA *adversary* $\mathcal{A}$ *against an UE scheme* UE*, there exist a* INT-CTXT^{s} *adversary* $\mathcal{B}$ *and a* detIND-UE-CPA *adversary* $\mathcal{B}'$ *such that* $\mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}CCA}}_{\mathsf{UE}}(\mathcal{A}) \leq 2q \cdot \mathbf{Adv}^{\mathsf{INT\text{-}CTXT^{s}}}_{\mathsf{UE}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}CPA}}_{\mathsf{UE}}(\mathcal{B}')$ *where* q *is the number of decryption queries made by* $\mathcal{A}$*.*

B The Check Algorithm

In our proofs, reductions play hybrid games and guess the location of the i-th insulated region. If the adversary sends a corrupt query inside this insulated region, the guess is wrong and reductions have to abort. We use the algorithm Check of [10], described in Fig. 8, to check if this event happens.

```
Check(inp, ê; e; fwl, fwr)
if ê > e then return ⊥
if inp = key and ê ∈ [fwl, fwr] then return ABORT
if inp = token and ê ∈ {fwl, fwr + 1} then return ABORT
```

Fig. 8. Algorithm Check of [10] used in our proofs. ê is the epoch in the adversary's request and e is the current epoch.

C Background on Isogenies

Let K be a finite field of characteristic $p > 0$ and E_1, E_2 elliptic curves over K. An *isogeny* $\varphi : E_1 \to E_2$ is a non-constant morphism sending the identity of E_1 to that of E_2. Its *degree* is defined as the degree of the corresponding rational map. If $\deg(\varphi) = d$ is coprime to p, then φ is *separable*, with kernel a subgroup of order d. Such an isogeny is completely determined (up to isomorphism of the target curve) by its kernel, yielding a one-to-one correspondence between separable isogenies and finite subgroups of $E_1(\overline{K})$. Given a subgroup G, the isogeny with kernel G is computed using Vélu's formula [48], and written $\varphi : E \to E/G$. The degree of $\varphi \circ \psi$ is equal to $\deg(\varphi)\deg(\psi)$. Each isogeny $\varphi : E_1 \to E_2$ admits a unique *dual isogeny* $\hat{\varphi} : E_2 \to E_1$ such that $\varphi \circ \hat{\varphi} = [\deg(\varphi)]$, the multiplication-by-$\deg(\varphi)$ map on E_2 and $\hat{\varphi} \circ \varphi = [\deg(\varphi)]$ on E_1. An *endomorphism* of E is an isogeny $\theta : E \to E$ or the constant map $[0]$. The set of endomorphisms $\mathrm{End}(E)$ forms a ring under addition and composition. A curve E over K is called *supersingular* if $\mathrm{End}(E)$ is a maximal order inside the quaternion algebra $B_{p,\infty}$ ramified at p and ∞.

Finally, the Frobenius morphism $\pi : (x, y) \to (x^p, y^p)$ sends a curve $E : y^2 = x^3 + ax + b$ to $E^{(p)} : y^2 = x^3 + a^p x + b^p$. Moreover, π is the only isogeny of degree p between any two supersingular curves.

D Background on the Deuring Correspondence

Let $a, b \in \mathbb{Q}^\times$, and let $B(a, b)$ denote the quaternion algebra over $\mathbb{Q}$ with basis $1, i, j, ij$ and relations $i^2 = a$, $j^2 = b$ and $ij = -ji$. There is a *canonical involution* on B mapping $\alpha := x + yi + zj + tij$ to $\bar{\alpha} := x - yi - zj - tij$. The *reduced trace* and *reduced norm* of α are defined as $\mathrm{Trd}(\alpha) := \alpha + \bar{\alpha} = 2x$ and $\mathrm{Nrd}(\alpha) := \alpha\bar{\alpha} = x^2 - ay^2 - bz^2 + abt^2$.

A lattice I in B is a $\mathbb{Z}$-module of the form $I := \mathbb{Z}b_1 + \mathbb{Z}b_2 + \mathbb{Z}b_3 + \mathbb{Z}b_4$, where the b_i's form a basis for the $\mathbb{Q}$-vector space B. The *reduced norm* of I is defined as $\mathrm{Nrd}(I) := \gcd(\{\mathrm{Nrd}(\alpha) \mid \alpha \in I\})$. An *order* $\mathcal{O}$ in B is a lattice that is also a subring of B. An order is *maximal* if it is not contained in any other order. For a lattice I, the *left* and *right orders* of I are defined as:

$$\mathcal{O}_L(I) := \{\alpha \in B \mid \alpha I \subseteq I\} \quad \text{and} \quad \mathcal{O}_R(I) := \{\alpha \in B \mid I\alpha \subseteq I\}$$

If $\mathcal{O}$ is a maximal order and I a left $\mathcal{O}$-ideal, then $\mathcal{O}_R(I)$ is itself a maximal order. Given two maximal orders $\mathcal{O}$ and $\mathcal{O}'$, there exists a lattice I, called a *connecting ideal*, such that $\mathcal{O}_L(I) = \mathcal{O}$ and $\mathcal{O}_R(I) = \mathcal{O}'$.

Two $\mathcal{O}$-ideals I and J are said to be *equivalent* if there exists $\alpha \in B^\times$ such that $I = J\alpha$, which we denote $I \sim J$. The set of equivalence classes under $\sim$ is the *left-ideal class set* of $\mathcal{O}$, denoted $\mathrm{Cls}(\mathcal{O})$. Let $\mathfrak{O} \subset \mathcal{O}$ be a quaternion suborder. If $I \sim J$ are two $\mathcal{O}$-ideals of norms coprime to disc $\mathfrak{O}$, we write $I \sim_{\mathfrak{O}} J$ if the element $\alpha \in B^\times$ such that $I\bar{J} = \mathcal{O}_L(J)\alpha$ belongs to $\mathfrak{O}$.

Let p be a prime congruent to 3 mod 4. Then $B_{p,\infty} = B(-1, -p)$ [45], where $B_{p,\infty}$ denotes the unique quaternion algebra ramified exactly at p and ∞. In this case, $B_{p,\infty}$ contains the maximal order $\mathcal{O}_0 := \langle 1, i, \frac{1+j}{2}, i\frac{1+j}{2} \rangle$. Throughout this paper, p is always a prime congruent to 3 mod 4, and $\mathcal{O}_0$ always denotes this specific maximal order.

The Deuring correspondence associates each maximal order $\mathcal{O}$ in $B_{p,\infty}$ with a supersingular elliptic curve E whose endomorphism ring $\mathrm{End}(E)$ is isomorphic to $\mathcal{O}$. That is, there is a bijection between the sets

$$\left\{ \begin{array}{c} \text{Isomorphism classes of} \\ \text{maximal orders in } B_{p,\infty} \end{array} \right\} \leftrightarrow \left\{ \begin{array}{c} \text{Isomorphism classes of} \\ \text{supersingular elliptic curves} \end{array} \right\} / \mathrm{Gal}(\mathbb{F}_p^2/\mathbb{F}_p)$$

In our setting, the supersingular curve E_0 defined over $\mathbb{F}_p$ by $y^2 = x^3 - x$ corresponds to the maximal order $\mathcal{O}_0$. Let π denote the Frobenius endomorphism $(x, y) \mapsto (x^p, y^p)$ and ι the endomorphism $(x, y) \mapsto (-x, \alpha y)$, where $\alpha \in \mathbb{F}_{p^2}$ satisfies $\alpha^2 = -1$. Then $\mathrm{End}(E_0) = \mathbb{Z} + \mathbb{Z}\iota + \mathbb{Z}\frac{1+\pi}{2} + \mathbb{Z}\iota\frac{1+\pi}{2}$. Since $\iota^2 = [-1]$ and $\pi^2 = [-p]$, we have $\mathrm{End}(E_0) \cong \mathcal{O}_0$.

The Deuring correspondence also preserves morphisms between these categories. To any isogeny $\varphi : E \to E'$, we can associate an ideal I_φ in $B_{p,\infty}$

such that $\mathcal{O}_L(I_\varphi) \cong \mathrm{End}(E)$ and $\mathcal{O}_R(I_\varphi) \cong \mathrm{End}(E')$. In this way, I_φ connects $\mathrm{End}(E)$ to $\mathrm{End}(E')$, just as φ connects E to E'. Moreover, $\mathrm{Nrd}(I_\varphi) = \deg(\varphi)$. Conversely, for any left $\mathrm{End}(E)$-ideal I, there exists an associated isogeny φ_I such that $I_{\varphi_I} = I$ and $\varphi_{I_\varphi} = \varphi$.

E A Variant with Faster Encryption and Decryption but Slower Updates

As noted in Remark 1, we now present a variant of DINE_ℓ that fully conforms to the standard UE interface, since all keys can be generated via KeyGen. The main modifications are highlighted in Fig. 9.

In DINE_ℓ the token is made of ℓ canonical HD representations $(s^{\mathsf{can}}_{\mathsf{HD}}(\delta^i_{\mathsf{e}+1}))_{1\le i\le \ell}$ which all together represent a degree $(2^f - N)^\ell$ isogeny $\delta_{\mathsf{e}+1} : E_{\mathsf{e}} \to E_{\mathsf{e}+1}$. However, when $k_{\mathsf{e}+1}$ is chosen beforehand, we cannot be sure to find an isogeny of exactly this degree. To address this, we instead select $\delta_{\mathsf{e}+1}$ to have degree $(2^f - N)^{\ell'}$ for some $\ell' > \ell$ using the SpecialKLPT algorithm. This isogeny can then be factored as the composition of ℓ' isogenies of degree $2^f - N$, denoted $\delta^{(i)}_{\mathsf{e}+1}$ for $1 \le i \le \ell'$. The actual token consists of the canonical HD representations of these component isogenies.

Concretely, the token can be computed using an algorithm we denote by IdealToSpecialEquivalentIsogeny, which we do not describe in full, as it can be straightforwardly extracted from IdealPullBack. Essentially, it outputs the successive representations s_1 computed internally during the main loop of IdealPullBack.

As noted in our technical overview, the security of the variant presented in this section is harder to analyze. This is because the chain of isogenies produced by SpecialKLPT—which forms the token—follows a highly specific distribution that is challenging to study. In contrast, in DINE_ℓ, the token is uniformly random. Analyzing the output distribution of a KLPT variant was central to the security of the original SQIsign signature scheme [22]; for this reason, SQIsign's main security assumption remains somewhat ad hoc, unlike the newer HD variants [8,19,2], whose security is much cleaner.

Remark 3. This variant of DINE_ℓ requires a slightly more general version of the SpecialKLPT algorithm presented in Section 5.2 that works for any input ideal, not just $\mathcal{O}_0$-ideals. The solution follows [22] to extend the original KLPT algorithm [33]. Let I be the input ideal. The idea is to find β of the desired norm in $\mathcal{O}_0 \cap I$, which can be done via strong approximation modulo $N\,\mathrm{Nrd}(K)\,\mathrm{Nrd}(L)$ (instead of $N\,\mathrm{Nrd}(K)$ in SpecialKLPT), where L is a connecting ideal between $\mathcal{O}_0$ and $\mathcal{O}_L(I)$. Consequently, the output norm increases roughly by a factor $\mathrm{Nrd}(L)^3$, which we can expect to be around $p^{3/2}$ (up to replacing L by an equivalent ideal of smaller norm). For more details, see [36, Section 3.3].

Regarding the implementation results for DINE_ℓ presented in Section 6, in the variant of DINE presented above, encryption and decryption would take roughly the same time as DINE's Upd, while updates would be slightly slower

$\mathsf{TokenGen}(k_e, k_{e+1})$

Parse k_e as an ideal I_e

Parse k_{e+1} as an ideal I_{e+1}

return $\Delta_{e+1} := (s^{\mathsf{can}}_{\mathsf{HD}}(\delta^{(i)}_{e+1}))_{1\le i\le \ell} \leftarrow \mathsf{IdealToSpecialEquivalentIsogeny}(I_e \cdot \overline{I_{e+1}})$

$\mathsf{Upd}(\Delta_{e+1}, C_e)$

Parse Δ_{e+1} as $(s^{\mathsf{can}}_{\mathsf{HD}}(\delta^{(i)}_{e+1}))_{1\le i\le \ell}$

Parse C_e as $s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2)$

for $i = 1$ **to** ℓ:

$s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1) \leftarrow \mathsf{HDPushForward}(s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\delta^{(i)}_{e+1}))$

$s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2) \leftarrow \mathsf{HDPushForward}(s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2), s^{\mathsf{can}}_{\mathsf{HD}}(\delta^{(i)}_{e+1}))$

return $C_{e+1} \leftarrow s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_1), s^{\mathsf{can}}_{\mathsf{HD}}(\varphi_2)$

Fig. 9. A variant with faster encryption and decryption but slower updates. All other algorithms are unchanged compared to Fig. 3.

than DINE's Dec. Token generation would also be slower than DINE's KeyGen, due to the slightly larger output required by the variant of SpecialKLPT.

F Proof of the detIND-UE-CPA Security of DINE

In this section, we study the detIND-UE-CPA security of DINE by giving a full proof of Theorem 2 which we recall below.

Theorem 2 (DINE is detIND-UE-CPA). *Let* DINE *be the UE scheme described in Fig. 3 for the special maximal order $\mathcal{O}_0$ and $N \in \mathbb{N}$. For any ideal cipher model adversary $\mathcal{A}$, there exists a reduction $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}CPA}}_{\mathsf{DINE}}(\mathcal{A}) \le 2n^3 \cdot \mathbf{Adv}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{B}) + \mathsf{negl}(\lambda).$$

We follow the proof strategy of [10,38] and use their hybrid argument across insulated regions. In each hybrid, we can embed at one firewall of the insulated region and simulate all tokens within that insulated region to answer queries to both $\mathcal{O}.\mathsf{Upd}$ and $\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}$. In DINE, fresh ciphertexts are randomized using a nonce R but updates are deterministic. Thus, our reduction will need to provide consistent ciphertexts to the adversary, *i.e.*, the nonce value R must be consistent.

We give a reduction $\mathcal{B}$ which receives $\mathcal{O}_0, N$ and an oracle BB that implements either the distribution $\mathcal{D}_0$ or $\mathcal{D}_1$ of Definition 6, *i.e.*, it returns either tuples of the form $((\varphi_{I_1}, \varphi_{I_2}), [\varphi_K]_*(\varphi_{I_1}, \varphi_{I_2}))$ or $((\varphi_{I_1}, \varphi_{I_2}), (\varphi_{J_1}, \varphi_{J_2}))$ where K is a fixed left $\mathcal{O}_0$-ideal and right $\mathcal{O}_a$-ideal, $(I_1, I_2) \leftarrow \mathcal{I}$, and J_1, J_2 are random left $\mathcal{O}$-ideals of norm N. $\mathcal{B}$ will use the samples of BB to perfectly simulate the

detIND-UE-CPA experiment for DINE when BB implements $\mathcal{D}_0$ (and a random experiment otherwise). The idea is to embed the ideal K to a well chosen epoch key by associating $\varphi_{I_1}, \varphi_{I_2}$ to a nonce-message pair and use $[\varphi_K]_*(\varphi_{I_1}, \varphi_{I_2})$ as ciphertext. Thus, if we know an efficient adversary $\mathcal{A}$ against the detIND-UE-CPA security of DINE, using the hybrid argument of [10], $\mathcal{B}$ can use $\mathcal{A}$ to break the pseudorandomness of the $(\mathcal{O}_0, N)$-pushforward.

Proof. Play hybrid games. We partition the non corrupted key space as follows: $\{1, \ldots, n\} \setminus \mathcal{K}^* = \cup_{(j,\mathsf{fwl}_j,\mathsf{fwr}_j)\in\mathcal{FW}}\{\mathsf{fwl}_j \ldots \mathsf{fwr}_j\}$, where fwl_i and fwr_i are firewalls of the i-th insulated region. For $b \in \{0,1\}$, define game $\mathcal{G}_i^b$ as $\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}CPA}\text{-}b}_{\mathsf{DINE}}(\mathcal{A})$ except for:

1. The game randomly picks $\mathsf{fwl}_i, \mathsf{fwr}_i \xleftarrow{\$} \{1, \ldots, n\}$ and if they are not the i-th firewalls, it aborts and returns a random bit b'. This loss is upper-bounded by n^2.
2. For the challenge (made in epoch $\tilde{\mathsf{e}}$ on input $(\bar{M}, \bar{C})$), the game returns an updated version of $\bar{C}$ if $\tilde{\mathsf{e}} < \mathsf{fwl}_i$ and it returns an encryption of $\bar{M}$ if $\tilde{\mathsf{e}} > \mathsf{fwr}_i$. Finally, if $\mathsf{fwl}_i \leq \tilde{\mathsf{e}} \leq \mathsf{fwr}_i$, the game returns an encryption of $\bar{M}$ if $b = 0$ and an updated version of $\bar{C}$ if $b = 1$.
3. After $\mathcal{A}$ outputs b', the game returns b' if $\mathsf{twf} \neq 1$ or some additional trivial win condition triggers.

If fwl_i, fwr_i are the desired values, then $\mathcal{G}_1^0$ is $\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}CPA}\text{-}0}_{\mathsf{DINE}}(\mathcal{A})$, *i.e.*, all challenges are encryptions of $\bar{M}$. Let ℓ be the total number of insulated regions (bounded by n), such that $\mathcal{G}_\ell^1$ is $\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}CPA}\text{-}1}_{\mathsf{DINE}}(\mathcal{A})$, *i.e.*, all challenges are updates of $\bar{C}$. Let E be the event that fwl_i and fwr_i are the desired values. By definition, for any $1 \leq i \leq n$ and $b \in \{0,1\}$, we have $\Pr[\mathcal{G}_i^b = 1 \mid \neg E] = 1/2$. Then

$$\begin{aligned}
\Pr[\mathcal{G}_\ell^1 = 1] &= \Pr[\mathcal{G}_\ell^1 = 1 \mid E] \cdot \Pr[E] + \Pr[\mathcal{G}_\ell^1 = 1 \mid \neg E] \cdot \Pr[\neg E] \\
&= \Pr[\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}CPA}\text{-}1}_{\mathsf{DINE}}(\mathcal{A}) = 1] \cdot \frac{1}{n^2} + \frac{1}{2} \cdot (1 - \frac{1}{n^2}), \text{and} \\
\Pr[\mathcal{G}_1^0 = 1] &= \Pr[\mathbf{Exp}^{\mathsf{detIND\text{-}UE\text{-}CPA}\text{-}0}_{\mathsf{DINE}}(\mathcal{A}) = 1] \cdot \frac{1}{n^2} + \frac{1}{2} \cdot (1 - \frac{1}{n^2})
\end{aligned}$$

Thus, we have $|\Pr[\mathcal{G}_\ell^1 = 1] - \Pr[\mathcal{G}_1^0 = 1]| = \frac{1}{n^2} \cdot \mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}CPA}}_{\mathsf{DINE}}(\mathcal{A})$.

Notice that the games $\mathcal{G}_{i-1}^1$ and $\mathcal{G}_i^0$ behave in the same way: for the challenge query and $\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}$, in an epoch in the first $i-1$ insulated regions, the reduction returns an update of $\bar{C}$, otherwise it returns an encryption of $\bar{M}$. Thus, for any $\ell \leq n$, $|\Pr[\mathcal{G}_\ell^1 = 1] - \Pr[\mathcal{G}_1^0 = 1]| \leq \sum_{i=1}^{\ell} |\Pr[\mathcal{G}_i^1 = 1] - \Pr[\mathcal{G}_i^0 = 1]|$. In the following, we prove that for any $1 \leq i \leq \ell$, $|\Pr[\mathcal{G}_i^1 = 1] - \Pr[\mathcal{G}_i^0 = 1]| \leq 2\mathbf{Adv}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{B})$ for a reduction $\mathcal{B}$.

In hybrid i. Let $\mathcal{A}_i$ be an adversary trying to distinguish $\mathcal{G}_i^0$ from $\mathcal{G}_i^1$. For all queries concerning epochs outside of the i-th insulated region, the responses of both games are the same. Thus, we assume that $\mathcal{A}_i$ asks for at least one challenge ciphertext in an epoch within the i-th insulated region. This is where we will embed the pseudorandom pushforward samples in our reduction.

We construct a reduction $\mathcal{B}$, presented in Fig. 10, that is trying to break the pseudorandomness of the $(\mathcal{O}_0, N)$-pushforward (Definition 6) and will simulate the responses of queries made by adversary $\mathcal{A}_i$.

Recall that $\mathcal{A}_i$ is an adversary attempting to distinguish $\mathcal{G}_i^0$ from $\mathcal{G}_i^1$. $\mathcal{B}$ will use $\mathcal{A}$ to break the pseudorandomness of the $(\mathcal{O}_0, N)$-pushforward. In $\mathbf{Exp}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{B})$, when BB returns samples of the form $(\varphi_{I_1}, \varphi_{I_2}, [\varphi_K]_*\varphi_{I_1}, [\varphi_K]_*\varphi_{I_2})$ for a left $\mathcal{O}_0$-ideal and right $\mathcal{O}$-ideal K, and $(I_1, I_2) \leftarrow \mathcal{I}$, $\mathcal{B}$ will perfectly simulate the environment of $\mathcal{A}_i$ in $\mathcal{G}_i^b$. When BB returns pairs of the form $(\varphi_{I_1}, \varphi_{I_2}, \varphi_{J_1}, \varphi_{J_2})$ for $(I_1, I_2) \leftarrow \mathcal{I}$ and J_1, J_2 random left $\mathcal{O}$-ideals of norm N, $\mathcal{B}$ will give random inputs to $\mathcal{A}_i$ such that $\mathcal{A}_i$ distinguishes $\mathcal{G}_i^0$ from $\mathcal{G}_i^1$ with advantage 0. We explain how our reduction $\mathcal{B}$ does this without knowing which BB oracle was provided to it.

The reduction $\mathcal{B}$ samples $b \xleftarrow{\$} \{0,1\}$ and simulates $\mathcal{G}_i^b$. Whenever the reduction needs to provide an output of $\pi(\cdot)$ to $\mathcal{A}_i$, it chooses some integers $z, t \in [-N/\sqrt{2p}, N/\sqrt{2p}]$ and programs $\pi(\cdot) = (z, t)$. We simulate the public permutation and its inverse by consistent lazy sampling, aborting on repeated nonce-message inputs or conflicting programming, and absorb the resulting negligible bad-event probability into the $\mathsf{negl}(\lambda)$ term in the bound, since this is standard we omit these checks for brevity. In this setting, computing π^{-1} is simply a lookup to this mapping of the ideal cipher π. We now explain our simulation. Initially,

1. $\mathcal{B}$ guesses the values of fwl_i and fwr_i.
2. $\mathcal{B}$ generates keys and tokens for epochs 1 to $\mathsf{fwl}_i - 1$.
3. $\mathcal{B}$ sends $\mathsf{fwl}_i, \mathsf{fwr}_i$, and the key k_{fwl_i-1} to the PP experiment and receives oracle access to BB, tokens $\Delta_{\mathsf{fwl}_i+1}, \ldots, \Delta_{\mathsf{fwr}_i}$, and the key k_{fwr_i+1}.
4. $\mathcal{B}$ generates keys and tokens for epochs $\mathsf{fwr}_i + 2$ to n using successive calls to $\mathsf{UpdateKey}$ from k_{fwr_i+1}.
5. $\mathcal{B}$ does not know the values of $k_{\mathsf{fwl}_i}, \ldots, k_{\mathsf{fwr}_i}, \Delta_{\mathsf{fwl}_i}, \Delta_{\mathsf{fwr}_i+1}$. If $\mathcal{A}_i$ corrupts these keys and tokens, this means that the firewall guess is wrong and the reduction aborts the game using the Check algorithm of Appendix B.

$\mathcal{B}$ will operate so as to embed the value K used by BB, when it implements $\mathcal{D}_0$, to the key k_{fwl_i}. If BB implements $\mathcal{D}_1$ instead, all the ciphertexts inside insulated region i will be random and independent of the challenge bit. To simulate a non-challenge ciphertext that is:

– An $\mathcal{O}.\mathsf{Enc}$ query in epoch $\mathsf{e} \in \{1, \ldots \mathsf{fwl}_i - 1\} \cup \{\mathsf{fwr}_i + 1, \ldots, n\}$: $\mathcal{B}$ queries BB to get a tuple $(\varphi_{I_1}, \varphi_{I_2}, \varphi_{J_1}, \varphi_{J_2})$. $\mathcal{B}$ uses $\varphi_{I_1}, \varphi_{I_2}$ as a value associated to the nonce-message pair by computing the quaternion $\theta := x + yi + zj + tij \in B_{p,\infty}$ through the $\mathsf{HalfEndoToCoord}$ algorithm called on $\varphi_{I_1}, \varphi_{I_2}$ and programming $\pi(\cdot) \leftarrow (z, t)$ (so the randomness will be consistent with calls that $\mathcal{A}_i$ makes to $\mathcal{O}.\mathsf{Upd}$), computes the ciphertext $C_\mathsf{e} := ([\varphi_{k_\mathsf{e}}]_*\varphi_{I_1}, [\varphi_{k_\mathsf{e}}]_*\varphi_{I_2})$ (the value of k_e is known to $\mathcal{B}$ in these epochs) and stores $(\varphi_{I_1}, \varphi_{I_2}, \varphi_{J_1}, \varphi_{J_2})$ in its memory for later use. To respond to $\mathcal{O}.\mathsf{Upd}$ queries in these epochs, $\mathcal{B}$ computes $C_\mathsf{e} := ([\varphi_{k_\mathsf{e}}]_*\varphi_{I_1}, [\varphi_{k_\mathsf{e}}]_*\varphi_{I_2})$ using the randomness $\varphi_{I_1}, \varphi_{I_2}$ generated during the first encryption of the input ciphertext.

Reduction $\mathcal{B}$ playing $\mathbf{Exp}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}$

receive pp
do Setup
$\bar{M}, \bar{C} \leftarrow \mathcal{A}^{\mathtt{ors}}(\mathsf{pp})$
$\mathsf{phase} \leftarrow 1$
$\tilde{C}_{\tilde{\mathsf{e}}} \leftarrow \mathcal{O}.\mathsf{Chall}(\bar{M}, \bar{C})$
$b' \leftarrow \mathcal{A}^{\mathtt{ors},\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}}(\tilde{C}_{\tilde{\mathsf{e}}})$
if $\mathcal{C}^* \cap \mathcal{K}^* \neq \emptyset$ **or** $\mathcal{I}^* \cap \mathcal{C}^* \neq \emptyset$
  $\mathsf{twf} \leftarrow 1$
if ABORT occurred **or** $\mathsf{twf} = 1$
  $b' \xleftarrow{\$} \{0,1\}$
  return b'
if $(i, \mathsf{fwl}_i, \mathsf{fwr}_i) \notin \mathcal{FW}$
  $b' \xleftarrow{\$} \{0,1\}$
  return b'
if $b' = b$ **then return** 0
else return 1

Setup

$b \xleftarrow{\$} \{0,1\}$
$k_1 \leftarrow \mathsf{DINE.KeyGen}(\mathsf{pp})$
$\mathsf{c}, \mathsf{phase}, \mathsf{twf} \leftarrow 0; \mathsf{e} \leftarrow 1$
$\mathcal{L}, \tilde{\mathcal{L}}, \mathcal{C}, \mathcal{K}, \mathcal{T} \leftarrow \emptyset$
$\mathsf{fwl}_i, \mathsf{fwr}_i \xleftarrow{\$} [1, n]$
for $j \in [2, \mathsf{fwl}_i - 1]$:
  $k_j, \Delta_j \leftarrow \mathsf{DINE.UpdateKey}(k_{j-1})$
send $\mathsf{fwl}_i, \mathsf{fwr}_i, k_{\mathsf{fwl}_i - 1}$ to PP
receive BB, $k_{\mathsf{fwr}_i+1}, \Delta_{\mathsf{fwl}_i+1}, \ldots, \Delta_{\mathsf{fwr}_i}$ from PP
for $j \in [\mathsf{fwr}_i + 2, n]$:
  $k_j, \Delta_j \leftarrow \mathsf{DINE.UpdateKey}(k_{j-1})$

$\mathcal{O}.\mathsf{Next}$

$\mathsf{e} \leftarrow \mathsf{e} + 1$

$\mathcal{O}.\mathsf{Enc}(M)$

$\mathsf{c} \leftarrow \mathsf{c} + 1$
$(\mathsf{inf}_1, \mathsf{inf}_2) \leftarrow \mathsf{BB}$
Sample a nonce R
$\pi(R \| M) \leftarrow \mathsf{HETC}(\mathsf{inf}_1)$
if $\mathsf{e} \in [1, \mathsf{fwl}_i - 1] \cup [\mathsf{fwr}_i + 1, n]$
  $C_{\mathsf{e}} \leftarrow [\varphi_{k_{\mathsf{e}}}]_* \mathsf{inf}_1$
else
  $C_{\mathsf{fwl}_i} \leftarrow \mathsf{inf}_2$
  for $j \in [\mathsf{fwl}_i + 1, \mathsf{e}]$
    $C_j \leftarrow [\Delta_j]_* C_{j-1}$
$\mathsf{inf} \leftarrow (\mathsf{inf}_1, \mathsf{inf}_2)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathsf{c}, C_{\mathsf{e}}, \mathsf{e}; \mathsf{inf})\}$
return C_{e}

$\mathcal{O}.\mathsf{Upd}(C_{\mathsf{e}-1})$

if $(\mathsf{c}, C_{\mathsf{e}-1}, \mathsf{e}-1; \mathsf{inf}) \notin \mathcal{L}$
  return $\perp$
if $\mathsf{e} \in [1, \mathsf{fwl}_i - 1] \cup [\mathsf{fwr}_i + 1, n]$
  $(\mathsf{inf}_1, \mathsf{inf}_2) \leftarrow \mathsf{inf}$
  $C_{\mathsf{e}} \leftarrow [\varphi_{k_{\mathsf{e}}}]_* \mathsf{inf}_1$
else
  $(\mathsf{inf}_1, \mathsf{inf}_2) \leftarrow \mathsf{inf}$
  $C_{\mathsf{fwl}_i} \leftarrow \mathsf{inf}_2$
  for $j \in [\mathsf{fwl}_i + 1, \mathsf{e}]$
    $C_j \leftarrow [\Delta_j]_* C_{j-1}$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathsf{c}, C_{\mathsf{e}}, \mathsf{e}; \mathsf{inf})\}$
return C_{e}

$\mathcal{O}.\mathsf{Corr}(\mathsf{inp}, \hat{\mathsf{e}})$

$\mathsf{Check}(\mathsf{inp}, \hat{\mathsf{e}}; \mathsf{e}; \mathsf{fwl}_i, \mathsf{fwr}_i)$
if $\mathsf{inp} = \mathsf{key}$
  $\mathcal{K} \leftarrow \mathcal{K} \cup \{\hat{\mathsf{e}}\}$
  return $k_{\hat{\mathsf{e}}}$
if $\mathsf{inp} = \mathsf{token}$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\hat{\mathsf{e}}\}$
  return $\Delta_{\hat{\mathsf{e}}}$

Fig. 10. Part 1 of our reduction $\mathcal{B}$ for proof of Th.2 in hybrid i. inf encodes fixed programming information: two isogenies pairs $(\mathsf{inf}_1, \mathsf{inf}_2)$ sampled by BB, where inf_1 is the encryption randomness and inf_2 is the ciphertext value in epoch fwl_i. $\mathtt{ors}$ denotes the set $\{\mathcal{O}.\mathsf{Enc}, \mathcal{O}.\mathsf{Next}, \mathcal{O}.\mathsf{Upd}, \mathcal{O}.\mathsf{Corr}\}$. HETC outputs the last two values returned by HalfEndoToCoord. Part 2 of the reduction is given in Fig. 11

$\mathcal{O}.\mathsf{Chall}(\bar{M}, \bar{C})$

$\tilde{\mathsf{e}} \leftarrow \mathsf{e}$
if $(\mathsf{c}, \bar{C}, \tilde{\mathsf{e}} - 1; \mathsf{inf}) \notin \mathcal{L}$
  return `ABORT`
$(\mathsf{inf}_1, \mathsf{inf}_2) \leftarrow \mathsf{inf}$
Sample a nonce R
if $b = 0$
  $(s_1, s_2) \leftarrow \mathsf{BB}$
  $\pi(R \| \bar{M}) \leftarrow \mathsf{HETC}(s_1)$
  $\tilde{C}_{\mathsf{fwl}_i} \leftarrow s_2$
else
  do
    $(z, t) \xleftarrow{\$} [-N/\sqrt{2p}, N/\sqrt{2p}]^2$
    $x, y \leftarrow \mathsf{PartialRepresentInteger}_{N^2}(z, t)$
  while $x, y = \perp$
  $\pi(R \| \bar{M}) \leftarrow (z, t)$
  $\tilde{C}_{\mathsf{fwl}_i} \leftarrow \mathsf{inf}_2$
for $j \in [1, \mathsf{fwl}_i - 1]$: // left
  $\tilde{C}_j \leftarrow [\varphi_{k_j}]_* \mathsf{inf}_1$
for $j \in [\mathsf{fwl}_i + 1, \mathsf{fwr}_i]$: // embed
  $\tilde{C}_j \leftarrow [\Delta_j]_* \tilde{C}_{j-1}$
$(z, t) \leftarrow \pi(R \| \bar{M})$
$x, y \leftarrow \mathsf{PartialRepresentInteger}_{N^2}(z, t)$
$\theta \leftarrow x + iy + jz + kt \in B_{p,\infty}$
$(I_1, I_2) \leftarrow (\mathcal{O}_0\langle\theta, N\rangle, \mathcal{O}_0\langle\bar{\theta}, N\rangle)$
for $j \in [\mathsf{fwr}_i + 1, n]$: // right
  $\tilde{C}_j \leftarrow [\varphi_{k_j}]_* (I_1, I_2)$
$\tilde{\mathcal{L}} \leftarrow \cup_{j=1}^{n} \{(\tilde{C}_j, j)\}$
return $\tilde{C}_{\mathsf{e}}$

$\mathcal{O}.\mathsf{Upd}\tilde{\mathsf{C}}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathsf{e}\}$
find $(\tilde{C}_{\mathsf{e}}, \mathsf{e}) \in \tilde{\mathcal{L}}$
return $\tilde{C}_{\mathsf{e}}$

Fig. 11. Part 2 of our reduction $\mathcal{B}$ for proof of Th.2 in hybrid i. inf encodes fixed programming information: two isogenies pairs $(\mathsf{inf}_1, \mathsf{inf}_2)$ sampled by BB, where inf_1 is the encryption randomness and inf_2 is the ciphertext value in epoch fwl_i. HETC outputs the last two values returned by $\mathsf{HalfEndoToCoord}$.

– An $\mathcal{O}.\mathsf{Enc}$ query in epoch $\mathsf{e} \in \{\mathsf{fwl}_i, \ldots, \mathsf{fwr}_i\}$: $\mathcal{B}$ queries BB to get a tuple $(\varphi_{I_1}, \varphi_{I_2}, \varphi_{J_1}, \varphi_{J_2})$ and programs $\pi(\cdot)$ using $\varphi_{I_1}, \varphi_{I_2}$ as above. It sets $C_{\mathsf{fwl}_i} = (\varphi_{J_1}, \varphi_{J_2})$ (so that all ciphertexts will be encrypted under the key K in epoch fwl_i if BB returns tuples of the form $(\varphi_{I_1}, \varphi_{I_2}, [\varphi_K]_*\varphi_{I_1}, [\varphi_K]_*\varphi_{I_2})$) and updates C_{fwl_i} to the right epoch e using its simulated tokens (remember that $\mathcal{B}$ does not know the keys inside the i-th insulated region). To respond to $\mathcal{O}.\mathsf{Upd}$ queries in these epochs, $\mathcal{B}$ uses the values $\varphi_{J_1}, \varphi_{J_2}$ (if $\varphi_{J_i} = [\varphi_K]_*\varphi_{I_i}$ then the randomness will still be consistent) generated during the first encryption of the input ciphertext as ciphertext value in epoch fwl_i and updates $\varphi_{J_1}, \varphi_{J_2}$ to the right epoch e using its simulated tokens.

During the challenge call, the adversary will provide a ciphertext $\bar{C}$ which was created during the c-th call to $\mathcal{O}.\mathsf{Enc}$. The adversary cannot ask for an update of the c-th encryption in an epoch $\mathsf{e} \geq \mathsf{fwl}_i$, as this would trigger the trivial win condition $[\mathsf{fwl}_i, \mathsf{fwr}_i] \subseteq \mathcal{I}^* \cap \mathcal{C}^* \neq \emptyset$.

To simulate challenge-equal ciphertext in an epoch that is:

– To the left of the i-th insulated region: $\mathcal{B}$ simulates $\mathsf{DINE.Upd}(\bar{C})$ using tokens that it created itself.
– Within the i-th insulated region: $\mathcal{B}$ simulates $\mathsf{DINE.Upd}(\bar{C})$ if $b = 1$, and simulates $\mathsf{DINE.Enc}(\bar{M})$ if $b = 0$. More precisely, if BB returns tuples of the form $(\varphi_{I_1}, \varphi_{I_2}, [\varphi_K]_*\varphi_{I_1}, [\varphi_K]_*\varphi_{I_2})$, $\mathcal{B}$ embeds K to k_{fwl_i}. If $b = 0$, the reduction samples $(\varphi_{I_1}, \varphi_{I_2}, \varphi_{J_1}, \varphi_{J_2}) \leftarrow \mathsf{BB}$, programs $\pi(R\|\bar{M})$ using the $\mathsf{HalfEndoToCoord}$ algorithm called on $(\varphi_{I_1}, \varphi_{I_2})$ as before and sets $\tilde{C}_{\mathsf{fwl}_i} = (\varphi_{J_1}, \varphi_{J_2})$ (we want $k_{\mathsf{fwl}_i} = K$) since $\tilde{C}_{\mathsf{fwl}_i} = \mathsf{DINE.Enc}(\bar{M}) = [\varphi_{k_{\mathsf{fwl}_i}}]_*(\varphi_{I_1}, \varphi_{I_2})$ by definition of $\pi(R\|\bar{M})$.
  If $b = 1$, assume that $\bar{C}$ is an update of $\bar{C}_{\mathsf{e_c}}$, the output of the c-th $\mathcal{O}.\mathsf{Enc}$ query. $\mathcal{B}$ sampled $(\varphi_{I_1}, \varphi_{I_2}, \varphi_{J_1}, \varphi_{J_2})$ using BB and used $\varphi_{I_1}, \varphi_{I_2}$ as randomness to create $\bar{C}_{\mathsf{e_c}}$ and to update it in epochs $\mathsf{e} < \mathsf{fwl}_i$. The reduction gives value $\varphi_{J_1}, \varphi_{J_2}$ to $\tilde{C}_{\mathsf{fwl}_i}$ since $\tilde{C}_{\mathsf{fwl}_i} = \mathsf{DINE.Upd}(\bar{C}) = ([\Delta_{\mathsf{fwl}_i}]_*([\varphi_{k_{\mathsf{fwl}_i - 1}}]_*\varphi_{I_1}), [\Delta_{\mathsf{fwl}_i}]_*([\varphi_{k_{\mathsf{fwl}_i - 1}}]_*\varphi_{I_2})) = ([\varphi_{k_{\mathsf{fwl}_i}}]_*\varphi_{I_1}, [\varphi_{k_{\mathsf{fwl}_i}}]_*\varphi_{I_2})$ by the correctness of updates of DINE (see Proposition 1). Furthermore, the reduction uses tokens $\Delta_{\mathsf{fwl}_i+1}, \ldots, \Delta_{\mathsf{fwr}_i}$ to update $\tilde{C}_{\mathsf{fwl}_i}$ to simulate all challenge ciphertexts in epochs within the insulated region.
– To the right of the i-th insulated region: $\mathcal{B}$ simulates $\mathsf{DINE.Enc}(\bar{M})$ using the keys that it created itself.

Eventually, $\mathcal{B}$ receives the output bit b' from $\mathcal{A}_i$. If $b' = b$, then $\mathcal{B}$ guesses that BB implements $\mathcal{D}_0$; otherwise, $\mathcal{B}$ guesses that BB implements $\mathcal{D}_1$. If $\mathcal{B}$ received an oracle BB implementing $\mathcal{D}_1$, then $\mathcal{B}$ wins with probability $1/2$, since the ciphertexts are completely random. By the correctness of updates and the definition of the isogenies and ideals returned by the PP experiment, $\mathcal{B}$ perfectly simulates the view of $\mathcal{A}_i$ in $\mathcal{G}_i^b$ when BB implements $\mathcal{D}_0$. A standard advantage computation then yields

$$\mathbf{Adv}^{\mathsf{detIND\text{-}UE\text{-}CPA}}_{\mathsf{DINE}}(\mathcal{A}) \leq 2n^3 \cdot \mathbf{Adv}^{\mathsf{PP}}_{\mathcal{O}_0,N,\mathcal{I}}(\mathcal{B}).$$

G Reduction for the Proof of Theorem 3

The reduction for the proof of Theorem 3 is shown in Fig. 12.

H Proof of Proposition 3

Proof. The running time follows from the polynomial complexity of SpecialKLPT and the complexity of AnyIdealToIsogeny. The matrix M can be computed in time $O(\mathsf{poly}(\log(p)))$ via discrete logarithms, as the points used in the HD representation have orders that are powers of two.

For correctness, we first show that the pullback of ψ through φ_J coincides with the pullback through φ_I. This follows from the fact that I and J are equivalent as $\mathbb{Z} + N\mathcal{O}_0$-ideals, as established in the proof of Proposition 1. This equivalence can be assumed due to the correctness of SpecialKLPT.

Now, factor $\varphi_J = \varphi_e \circ \cdots \circ \varphi_1$, where each φ_i has degree $2^f - N$. The pullback of ψ through this chain of e isogenies equals the iterated pushforward of ψ through $\hat{\varphi}_1, \cdots, \hat{\varphi}_e$. Let ψ_i denote the iterated pushforwards, with $\psi_0 = \psi$ and $\psi_1 = [\hat{\varphi}_e]_* \psi_0$, and so on. We must show that, before executing HDPushForward at step i of the loop, s_1 correctly represents $\hat{\varphi}_e \circ \cdots \circ \hat{\varphi}_{e-i+1}$, and s correctly represents ψ_{i-1}, with both having the same domain basis.

Before the loop, by the correctness of the input representation, we have $P', Q' = \psi(P, Q)$. After applying the matrix M, linearity of the isogeny action ensures that $R', S' = \psi(R, S)$, so at this stage s contains a valid representation of ψ_0, with $R, S = \varphi_J(P_0, Q_0)$. In the first iteration of the loop, the Deuring correspondence implies that the ideal K_2 corresponds to φ_e and K_1 corresponds to $\varphi_{K_1} := \varphi_{e-1} \circ \cdots \circ \varphi_1$. Thus, we have $\hat{\varphi}_e(R, S) = [2^f - N]\varphi_{K_1}(P_0, Q_0)$. This confirms that, during the first iteration of the loop, the representations stored in s and s_1 satisfy the desired property.

After executing HDPushForward, by the correctness of that algorithm, s contains a representation of ψ_1, with the codomain basis given by $[2^f - N]\varphi_{e-1} \circ \cdots \circ \varphi_1(P_0, Q_0)$. The argument for the correctness of the inputs to HDPushForward at each subsequent iteration of the loop follows analogously. □

Reduction $\mathcal{B}$ playing $\mathbf{Exp}^{\mathsf{UP}}_{\mathcal{O}_0,N,\mathcal{I}}$

receive pp
do Setup
$\mathcal{A}^{\mathtt{ors}}(\mathsf{pp})$
if ABORT occurred
  return win

Setup

$k_1 \leftarrow \mathsf{DINE.KeyGen}(\mathsf{pp})$
$\mathsf{phase}, \mathsf{win} \leftarrow 0; \mathsf{e} \leftarrow 1; \mathsf{c} \leftarrow -1$
$\mathcal{L}^*, \mathcal{C}, \mathcal{K}, \mathcal{T} \leftarrow \emptyset$
$a \xleftarrow{\$} [1, n]$
for $j \in [2, a-1]$:
  $k_j, \Delta_j \leftarrow \mathsf{DINE.UpdateKey}(k_{j-1})$
send a, n, k_{a-1} to UP
receive $\mathsf{New}, \mathsf{Push}, \Delta_{a+1}, \ldots, \Delta_n$

$\mathcal{O}.\mathsf{Enc}(M)$

$\mathsf{c} \leftarrow \mathsf{c} + 1$
$\mathsf{inf}_1 \leftarrow \mathsf{New}$
Sample a nonce R
$\pi(R \| M) \leftarrow \mathsf{HETC}(\mathsf{inf}_1)$
if $\mathsf{e} \in [1, a-1]$
  $\mathsf{inf}_2 \leftarrow \perp$
  $C_{\mathsf{e}} \leftarrow [\varphi_{k_{\mathsf{e}}}]_* \mathsf{inf}_1$
else
  $\mathsf{inf}_2 \leftarrow \mathsf{Push}(\mathsf{c})$
  $C_a \leftarrow \mathsf{inf}_2$
  for $j \in [a+1, \mathsf{e}]$:
    $C_j \leftarrow [\Delta_j]_* C_{j-1}$
$\mathsf{inf} \leftarrow (\mathsf{inf}_1, \mathsf{inf}_2)$
$\mathcal{L}^* \leftarrow \mathcal{L}^* \cup \{(\mathsf{c}, C_{\mathsf{e}}, \mathsf{e}; \mathsf{inf})\}$
return C_{e}

$\mathcal{O}.\mathsf{Next}$

$\mathsf{e} \leftarrow \mathsf{e} + 1$
$\mathcal{O}.\mathsf{Upd}(C_{\mathsf{e}-1})$

if $(\mathsf{c}, C_{\mathsf{e}-1}, \mathsf{e}-1; \mathsf{inf}) \notin \mathcal{L}^*$
  return $\perp$
if $\mathsf{e} \in [1, a-1]$
  $(\mathsf{inf}_1, \mathsf{inf}_2) \leftarrow \mathsf{inf}$
  $C_{\mathsf{e}} \leftarrow [\varphi_{k_{\mathsf{e}}}]_* \mathsf{inf}_1$
else
  $(\mathsf{inf}_1, \mathsf{inf}_2) \leftarrow \mathsf{inf}$
  if $\mathsf{inf}_2 = \perp$
    $\mathsf{inf}_2 \leftarrow \mathsf{Push}(\mathsf{c})$
  $C_a \leftarrow \mathsf{inf}_2$
  for $j \in [a+1, \mathsf{e}]$:
    $C_j \leftarrow [\Delta_j]_* C_{j-1}$
$\mathsf{inf} \leftarrow (\mathsf{inf}_1, \mathsf{inf}_2)$
$\mathcal{L}^* \leftarrow \mathcal{L}^* \cup \{(\mathsf{c}, C_{\mathsf{e}}, \mathsf{e}; \mathsf{inf})\}$
return C_{e}

Fig. 12. Part 1 of our reduction $\mathcal{B}$ for the proof of Th.3. ors denotes the set $\{\mathcal{O}.\mathsf{Enc}, \mathcal{O}.\mathsf{Next}, \mathcal{O}.\mathsf{Upd}, \mathcal{O}.\mathsf{Corr}, \mathcal{O}.\mathsf{Try}\}$. inf encodes fixed programming information: two isogenies pairs $(\mathsf{inf}_1, \mathsf{inf}_2)$, where inf_1 is the encryption randomness returned by New and inf_2 is the ciphertext value in epoch a returned by Push. HETC outputs the last two values returned by HalfEndoToCoord. Part 2 of the reduction is given in Fig. 13.

$\mathcal{O}.\mathsf{Corr}(\mathsf{inp}, \hat{\mathsf{e}})$	$\mathcal{O}.\mathsf{Try}(\tilde{C})$
do $\mathsf{Check}(\mathsf{inp}, \hat{\mathsf{e}}; \mathsf{e}; a, n)$	$\tilde{\mathsf{e}} \leftarrow \mathsf{e}$
if $\mathsf{inp} = \mathsf{key}$	**if** $\mathsf{phase} = 1$
$\mathcal{K} \leftarrow \mathcal{K} \cup \{\hat{\mathsf{e}}\}$	**return** $\perp$
return $k_{\hat{\mathsf{e}}}$	$\mathsf{phase} \leftarrow 1$
if $\mathsf{inp} = \mathsf{token}$	**if** $\tilde{C} \in \mathcal{L}^*$
$\mathcal{T} \leftarrow \mathcal{T} \cup \{\hat{\mathsf{e}}\}$	**abort**
for $i \in \mathcal{T}^*$:	**if** $\tilde{\mathsf{e}} < a$ **or** $a \in \mathcal{T}^*$
for $(j, C_{i-1}, i-1; \mathsf{inf}) \in \mathcal{L}^*$:	**abort**
$C_i \leftarrow [\Delta_i]_* C_{i-1}$	**if** $[a+1, \tilde{\mathsf{e}}] \setminus \mathcal{T}^* \neq \emptyset$ **or** $[a, \tilde{\mathsf{e}}] \cap \mathcal{K}^* \neq \emptyset$
$\mathcal{L}^* \leftarrow \mathcal{L}^* \cup \{(j, C_i, i; \mathsf{inf})\}$	**abort**
return $\Delta_{\hat{\mathsf{e}}}$	$\psi \leftarrow [\bigcirc_{j=\tilde{\mathsf{e}}}^{a+1} \Delta_j]^* \tilde{C}$
	output ψ **to** $\mathbf{Exp}^{\mathsf{UP}}_{\mathcal{O}_0, N, \mathcal{I}}(\mathcal{B})$
	get b
	$\mathsf{win} \leftarrow b$
	abort

Fig. 13. Part 2 of our reduction $\mathcal{B}$ for the proof of Th.3.