NICE-PAKE: On the Security of KEM-Based PAKE Constructions without Ideal Ciphers

Nouri Alnahawi[1,5,6], Jacob Alperin-Sheriff[2], Daniel Apon[3], Gareth T. Davies[4], and Alexander Wiesmaier[1,5,6]

[1] Hochschule Darmstadt. {nouri.alnahawi,alexander.wiesmaier}@h-de.de
[2] Independent Researcher. jacobmas@gmail.com
[3] The MITRE Corporation. crypto@mitre.org
[4] NXP Semiconductors. garethtomas.davies@nxp.com
[5] National Research Center for Applied Cybersecurity ATHENE.
[6] European University of Technology, European Union.

Abstract. Password Authenticated Key Exchange (PAKE) is a fundamental cryptographic component that allows two parties to establish a shared key using only (potentially low-entropy) passwords. The interest in realizing generic KEM-based PAKEs has increased significantly in the last few years as part of the global migration effort to quantum-resistant cryptography. One such PAKE is the CAKE protocol, proposed by Beguinet et al. (ACNS '23). However, despite its simple design based on the well-studied EKE protocol both CAKE and its variant OCAKE do not fully protect against quantum adversaries, as they rely on the Ideal Cipher (IC) model. Related and follow-up works, although touching on that issue, still rely on an IC. Considering the lack of a quantum IC model and the difficulty of using the classical IC to achieve secure instantiations on public keys in general and PQC in particular, we set out to eliminate it from PAKE design. In this paper, we present the **N**o **IC** **E**ncryption (**NICE**)-**PAKE**, a (semi)-generic symmetric PAKE framework providing a quantum-safe alternative for the IC, utilizing simpler cryptographic components for the authentication step. To give a formal proof for our construction, we introduce the notions of **A**-Part Secrecy ($\mathsf{A\text{-}SEC\text{-}CCA}$), Splittable Collision Freeness ($\mathsf{A\text{-}CFR\text{-}CCA}$) and Public Key Uniformity ($\mathsf{SPLIT\text{-}PKU}$) for splittable LWE KEMs. We show the relation of the former to the Non-uniform LWE and the Weak Hint LWE assumptions, as well as its application to ring and module LWE. Notably, this side quest led to some surprising discoveries: the new notion is not directly interchangeable between the LWE variants, at least not in a straightforward manner. Further, we show how to obtain a secure PAKE from our construction with concrete parameter choices for both FrodoKEM and CRYSTALS-Kyber. We also address fundamental issues with common IC usage and identify differences between lattice KEMs (and their public keys) regarding their suitability for generic post-quantum PAKEs.

Keywords: Password Authenticated Key Exchange · PAKE · Key Encapsulation Mechanism · KEM · Post-Quantum Cryptography · PQC · Learning with Errors · LWE · Ideal Cipher Model · IC

1 Introduction

As the looming threat of large-scale quantum computers endangering public-key cryptography becomes more evident, the search for quantum-safe replacement primitives and schemes in cryptographic protocols and constructions has also begun to gain more importance. The efforts made towards this goal can be seen, e.g., in the work done throughout the NIST Post-Quantum Cryptography (PQC) standardization process [58], which mainly focuses on Key Encapsulation Mechanisms (KEM) and Digital Signatures. In 2023, a set of candidates, including CRYSTALS-Kyber [19,59] (now referred to as ML-KEM[1]), were announced, and their standardization was recently finalized[2]. These candidates are mainly meant to replace classical asymmetric schemes based on the discrete logarithm and integer factorization assumptions. KEMs and signatures will be used in security protocols such as Transport Layer Security (TLS) and Internet Key Exchange version 2 (IKEv2), and thus have been analyzed and tested extensively in the last decade for that context [5].

Password Authenticated Key Exchange (PAKE) protocols establish a session key between communication parties over insecure channels using an asymmetric key agreement scheme. Since PAKEs do not use static public keys or certificates, they provide an alternative to static authentication using a low entropy *long-lived-key* (pre-shared secret or password) known only to the communicating parties. Among several methods to perform this task [33], the very first PAKE, Encrypted Key Exchange (EKE) [14], relies on the simple idea of (symmetrically) encrypting the initiator's public key using the password. The encrypted public key can only be decrypted and, hence, correctly used by a receiver who has the same password. Thus, it serves as authentication for both sides of the key agreement. Many PAKE constructions rely on quantum-vulnerable hardness assumptions such as (C)DH [33], which has led to increased attention on PAKEs from generic building blocks like KEMs (cf. Sec. 2).

Motivation. In 2023 and 2024, five generic KEM-based PQC PAKEs were proposed [11,46,4,51,10], with a noticeable focus on lattice-based schemes from the Learning with Errors (LWE) and Module LWE assumptions (Sec. 2). Despite some differences, these works have in common that they rely on the *Ideal Cipher* (IC) model in their formal security analysis. Essentially, it models a random block cipher as being chosen uniformly from the set of all possible block ciphers [15] and allows for defining a deterministic behavior on such ciphers. Modeling the encryption of public keys in PAKE protocols as an IC has significant drawbacks that affect confidence in generic constructions (elaborated further in Sec. 2 and Sec. 4). If the protocol requires cipher output to be statistically close to uniform then ML-KEM—the most viable candidate for instantiation—cannot be

[1] `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf`

[2] `https://www.federalregister.gov/documents/2024/08/14/2024-17956/announcing-issuance-of-federal-information-processing-standards-fips-fips-203-module-lattice-based`

Alice		**Bob**
$Password\ \pi$		$Password\ \pi'$
$sk_a, pk_a \leftarrow \mathsf{KEM.KGen}(1^\kappa)$		
$(\rho, \mathbf{b}) \leftarrow pk_a$		
$z \leftarrow \rho \oplus \mathcal{H}(\pi)$		
	$\xrightarrow{z, \mathbf{b}}$	
		$\rho' \leftarrow z \oplus \mathcal{H}(\pi')$
		$pk'_a \leftarrow (\rho', \mathbf{b})$
		$(C_b, \overline{K'}) \leftarrow \mathsf{KEM.Encap}(pk'_a)$
	$\xleftarrow{C_b}$	
$\overline{K} \leftarrow \mathsf{KEM.Decap}(sk_a, C_b)$		

Fig. 1. The NICE-PAKE protocol utilizing a KEM with splittable keys using a random seed for the uniform sampling of the **A**-part of the public key.

used, since its public keys are only computationally indistinguishable from uniform [34]. Furthermore, it is non-trivial to instantiate an IC as a block cipher over a group domain [10] and there are still no known adaptations for the IC model to deal with adversaries with quantum capabilities [57].

Main Contribution. We present the **N**o **IC** **E**ncryption (**NICE**) PAKE protocol (depicted in Fig. 1) to address the issues arising from the use of the IC model in post-quantum PAKE. We realize the PAKE public key authentication step in the form of a bit-wise XOR operation, instead of symmetric encryption of the public key. Inspired by [51,10], our construction incorporates LWE KEMs with splittable public keys of the form $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$, where the lattice base $\mathbf{A}$ is sampled from a bitstring seed. As the seed is usually either appended or prepended to the plain text representation of the public key, we mask (hide) it using the PAKE password. Thus, we allow perfect secrecy schemes to be applied on random bit strings of fixed length without resorting to block ciphers.

To understand the intuitive security of the protocol, note that without knowledge of the correct password π, it is not possible to retrieve the seed required for sampling the matching lattice base used to invoke the encapsulation routine, since it internally requires both parts of the public key. Ciphertexts generated from a key a with a non-matching **A**-part cannot be decapsulated. Thus, only with a password known by both parties is it possible to successfully execute the KEM. Although the protocol only provides implicit mutual authentication, a version with explicit authentication can nevertheless be obtained through adding a key confirmation round as a final step.

To support our formal analysis, we introduce three new properties for LWE (RLWE and MLWE) KEMs (Sec. 3.6) and refer to them as **A**-Part Secrecy under Chosen Ciphertext Attacks (A-SEC-CCA), Splittable Collision Freeness (A-CFR-CCA), and Public Key Uniformity (SPLIT-PKU) for KEMs with splittable public keys. We provide a formal security analysis for our protocol in the extended RoR (*Real-or-Random*) BPR (Bellare-Pointcheval-Rogaway) model [13] (Sec. 6) and discuss concrete instantiations from secure PQC KEMs (Sec. 7). Our work therefore represents a major step in understanding how secure PAKEs can be built from lattice-based KEMs.

Bonus Contribution. Our proof for the newly introduced A-SEC-CCA property (Sec. 5.2) is based on arguably less famous LWE assumptions. We show that this property requires only a straight-forward reduction to schemes built from plain (unstructured lattices) LWE assumptions, yet comes with a costly penalty leading to a significant loss in bit security for the concrete LWE scheme FrodoKEM. Nevertheless, we present a way to regain a sufficient amount of the lost bits for FrodoKEM using a non-standard value for the Gaussian variance parameter (Sec. 7). Further, we conclude that this method is not directly inter-changeable across other schemes from the Ring or Module LWE variants (e.g., NewHope and CRYSTALS-Kyber), and could lead to a complete break in security using the standard parameters (Sec. 7). Hence, we suggest that with a secret of higher entropy for the encapsulator's randomness, the A-SEC-CCA property could also be applied to MLWE schemes with an acceptable loss in bit security resulting from reducing the dimension of the module lattice.

Idealized Objects (ROM vs. QROM). We note that we make use of only one idealized object, the classical ROM, in our analysis. Hence, we do not require the QROM, as the ROM is only used to model queries on a pre-defined and offline accessible password dictionary. This claim can be justified through observing one simple fact: It is safe to assume that an adversary can execute offline dictionary attacks (even without a quantum computer). Hence, they can easily prepare a list of all hash outputs for all possible passwords in the password dictionary. In PAKE proofs, this attack often corresponds to queries using the password as input to a RO-modeled hash function or to a RO-modeled key derivation function used in combination with the IC. In our construction, an adversary can indeed only make use of such queries before committing to a password guess, which they cannot change adaptively at a later stage.

Extraction of adversarial password guesses. Conducting the formal analysis, we concluded that it is infeasible to extract password guesses from an adversary's interaction with the protocol. Although our construction and its informal security arguments may look similar to well known discrete-log based schemes, it is not possible to rely on interactive assumptions combined with the ROM (or the QROM for that matter) to extract password guesses in this case. Considering the flow of a protocol execution, one may observe the following impossibilities:

1. Extraction from the RO-modeled function $\mathcal{H}$ is not possible, as the oracle implementing this function only takes the password as input, and not a public key. Hence, an adversary can query $\mathcal{H}$ on multiple passwords π_i and later decide which one to use in masking their chosen public key pk.

2. Extraction from an adversary's ciphertext C_b is not possible, as the underlying KEM provides strong anonymity ($\mathsf{ANO\text{-}CCA}$) and hence protects the public key used in the encapsulation routine.
3. Extraction from a contrived RO-modeled final key derivation function in the QROM is impossible, as observing the adversary's input to this function in superposition will cause it to collapse, and ultimately the adversary's password choice will remain unknown.

CPA vs. CCA Security. Initially, we built our proof strategy (cf. App. B) based on the CPA variants of the notions for key security ($\mathsf{IND\text{-}CPA}$) and anonymity ($\mathsf{ANO} - \mathsf{CPA}$). Nevertheless, we opted to use the weaker assumptions $\mathsf{IND\text{-}CCA}$and $\mathsf{ANO\text{-}CCA}$ within the actual formal analysis for two main reasons:

1. The *anonymity* property aims at protecting the identity of the public key used in an encapsulation to generate a ciphertext. Therefore, it should be sufficient to rely on CPA security for the interaction between an honest user and an adversary within one session. However, anonymity restricts both an adversary and the simulation from identifying public keys used to generate ciphertexts. This leads to the simulation's inability to extract password guesses from ciphertexts forwarded by an adversary to sessions that are connected with the instance, with whom the adversary is interacting. This means that the adversary may trick the simulation into decrypting malicious ciphertexts and then revealing the decapsualted keys, which means that CCA security is indeed required.
2. Similarly for *indistinguisgability*, consider an adversary that sees the public key and the ciphertext in an honest or simulated protocol execution. The adversary's view on one connected session can be leveraged to forward values to other sessions using the same password and tricking the simulation into decrypting their adaptively chosen ciphertexts and then revealing the decapsualted keys, which means that CCA security is also required.

2 Related Work

Katz and Vaikuntanathan [37] introduced one of the earliest works on PQC PAKEs in 2009, which may be considered the first quantum-safe PAKE based on the lattice LWE problem and post-quantum Adaptive Smooth Projective Hash (ASPH) systems. The construction of Zhu et al. [35] in 2014 may similarly be considered the first PQC PAKE based on isogenies. However, the Ring (R)LWE PQC-PAK by Ding et al. [27] published in 2017, arguably marks the emergence of several other PAKE proposals from PQC primitives, more precisely based on lattice LWE (e.g. [54,60,32,24]) and isogenies (e.g. [56,55,1]). To our knowledge, the only PQC PAKE with security analysis in the Quantum Random Oracle Model (QROM) so far was given in 2024 by Lyu et al. [42]. The authors proposed the first PAKE protocol form isogeny assumptions in the Universal Composability (UC) framework and the Random Oracle Model (ROM), as well as two PAKE protocols from lattice LWE and the group-action decisional Diffie-Hellman (GA-DDH) in the QROM. Their presented construction is based on lossy public key encryption (LPKE) and is extended to security in the QROM by replacing the basic LPKE with an extractable LPKE (eLPKE). Moreover, the authors apply a Fujisaki-Okamoto (FO) transformation to elevate the security of the chosen LPKE from IND-CPA to IND-CCA. The aforementioned works utilize PQC asymmetric primitives directly, i.e., they do not utilize PQC KEMs in a non-modified black-box manner. In the following, we relate to generic and black-box KEM-based designs in more detail.

In 2023, Beguinet et al. [11] presented the first generic PQC PAKEs, the CAKE and OCAKE protocols, and provided a security analysis of their constructions in the UC framework utilizing the IC and ROM models. The CAKE suite is a generic transformation from KEM to PAKE based on the classical PAKEs EKE and OEKE. The high-level idea is to encrypt the public key and the ciphertext using the password to provide explicit mutual authentication in the CAKE variant. Alternatively, in OCAKE, the ciphertext is authenticated with a key confirmation tag only, which provides explicit authentication for the receiver. CAKE and OCAKE require that the underlying used KEM fulfills the notions of Key Indistinguishability (IND-CPA), public key fuzziness (Fuzzy KEM), and ciphertext anonymity (Anonymous KEM). Pan and Zeng [46] as well as Alnahawi et al. [4] presented further security analysis for CAKE and OCAKE respectively. Unlike the UC proof of [11], the two additional security proofs were presented in the BPR model. Pan and Zeng [46] suggested the notion of Anonymity under Plaintext Checking Oracles (ANO-PCA) for the chosen KEM and extended the security proof to handle multi-user challenges. The authors of [4] also adapted similar anonymity and multi-user notions and formulated the notion of Public Key Uniformity (KEM-PKU) as a replacement for the Fuzzy-KEM property. Additionally, Alnahawi et al. provided mutual explicit authentication by adding a key confirmation round, and showed how to formally handle password guesses in a detailed game-based proof.

Dos Santos et al. [51] presented a new way to construct a UC-secure PAKE protocol under a relaxation of the IC called Half-Ideal Cipher (HIC). Their EKE-KEM protocol utilizes a KEM and a modified 2-round Feistel construction, which they call m2F. The m2F avoids using an IC over a group through defining an IC over a fixed-length bit-string domain. Following that, Arriaga et al. [10] introduced the Compact HIC (CHIC) protocol, which improves the construction of EKE-KEM [51] in computation and communication costs. The CHIC protocol utilizes the m2F construction in a white-box manner by using the public seed of a splittable KEM public key as the ephemeral randomness input used in the m2F construction of [51]. The authors in [10] also address the required KEM properties and define the notions of Passive One-Way Security (OW-CPA), Pseudo-Uniformity of Public Keys (UNI-PK), and Anonymity under Plaintext-Checkable Attacks (ANO-iPCA) for a chosen KEM.

Relying on a quantum equivalent if the IC is not yet an option. In principle it is impossible to utilize the IC capabilities (cf. Sec. 4) due to the quantum nocloning theorem and the infeasibility of rewinding or back-patching. Few works in the literature [2,36,52] address the notion of the *Quantum Ideal Cipher* (QIC) model, yet do not show how to fully match the classical one or the lazy sampling technique. These works mainly focus on one-way functions and non-invertible permutations. The work by Unruh [57] builds upon the idea of compressed function oracles (CFO) [61] and takes a step forward in modeling keyed invertible permutations (i.e., IC) in quantum settings by introducing compressed permutation oracles (CPO). Nevertheless, despite the proposed approach's plausibility, it is not yet formally proven that a CPO is indistinguishable from a truly random permutation [57]. Recent work by Majenz et al. [44] introduces permutation superposition oracles as a generalization of CFOs and shows the possibility of their usage in quantum queries on permutations and their inverses, which may aid in realizing the QIC model in the future.

3 Preliminaries

In this section, we present the definitions of the main building blocks of our protocol and describe their respective security properties.

3.1 Notation

The security parameter is denoted by $\kappa \in \mathbb{N}$, and we use 1^κ for its unary representation. $a \leftarrow \mathsf{Alg}(b)$ represents running (potentially randomized) algorithm Alg on input b (with fresh random coins). We denote assignment of values to other values with $a \leftarrow (b, c)$. Choosing values at random (uniformly, or otherwise if it is clear from context) from a set $\mathcal{S}$ is denoted by $a \leftarrow\!\!\$\, \mathcal{S}$.

We use the notation $\mathsf{Adv}^{\mathsf{PROP}}_{\mathcal{A},\Pi}(1^\kappa)$ to represent adversary $\mathcal{A}$'s advantage in playing the security experiment $\mathsf{Exp}^{\mathsf{PROP}}_{\mathcal{A},\Pi}(1^\kappa)$ for security property PROP with primitive/protocol Π. To ease exposition in inline text we will often omit the security parameter from this experiment representation. For bit-guessing security experiments we follow convention and define two separate experiments parameterized by the choice bit, and define $\mathsf{Adv}^{\mathsf{PROP.BIT}}_{\mathcal{A},\Pi}(1^\kappa) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{PROP.BIT}^0}_{\mathcal{A},\Pi}(1^\kappa) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{PROP.BIT}^1}_{\mathcal{A},\Pi}(1^\kappa) = 1\right]\right|$. For experiments where the adversary tries to find collisions or in some way outputs something that is not a bit, we define in the usual way $\mathsf{Adv}^{\mathsf{PROP.FIND}}_{\mathcal{A},\Pi}(1^\kappa) = \Pr\left[\mathsf{Exp}^{\mathsf{PROP.FIND}}_{\mathcal{A},\Pi}(1^\kappa) = 1\right]$.

3.2 The Random Oracle Model

A random oracle [12] (RO) is an ideal primitive that models a random hash function that responds to each query to a given input value $m \in M$ with a random value $h = \mathcal{H}(m)$. Additionally, a RO keeps a record of all placed queries and responds with the same value for a previously queried input.

Definition 1 (Random Oracle). *RO is a function f that maps elements over the function space $\{0,1\}^* \to \{0,1\}^{poly(\lambda)}$, where:*

$$\textbf{if } f(x) \neq \bot \textbf{ return } f(x) \textbf{ else return } y \leftarrow\!\!\$\, \{0,1\}^{poly(\lambda)}$$

3.3 BPR AKE Security

$\mathcal{A}$'s goal in the BPR security experiment is to distinguish between real and random session keys determined by the bit b for an accepting and fresh user.

Definition 2 (Security of Authenticated Key Exchange (AKE)). *For a protocol Π and an adversary $\mathcal{A}$, we define $\mathcal{A}$'s advantage with respect to Π as $\mathsf{Adv}^{\mathsf{BPR}}_{\mathcal{A},\Pi}(1^\kappa)$. We say that the adversary wins if on issuing a `Test` query for a user $\mathcal{U}_i$ that has terminated in accepting state (i.e. is in possession of a session key K) and no `Reveal` or `Corrupt` query has been issued to this user, $\mathcal{A}$ correctly guesses the bit selected in the `Test` query. We say the protocol Π is secure if the probability of $\mathcal{A}$ winning is bounded to a negligible quantity:*

$$\mathsf{Adv}^{\mathsf{BPR}}_{\mathcal{A},\Pi}(1^\kappa) \leq \frac{q_s}{|\mathcal{D}|} + \varepsilon(1^\kappa)$$

where q_s is the number of sessions $\mathcal{A}$ actively interacts with, $\mathcal{D}$ is a password dictionary and ε is a negligible function.

3.4 Lattice NLWE, eLWE and whLWE

In order to obtain the new KEM security properties mentioned in Sec. 1 and formally defined in Sec. 5.2, we make use of the extended LWE (eLWE), Non-Uniform LWE and Weak-Hint LWE assumptions described in the following. Essentially, we rely on NLWE to show that a ciphertext produced by an LWE encryption scheme can be viewed as an LWE sample from a certain distribution. Analogously, we rely on eLWE and whLWE to determine the amount of leakage on LWE secrets in the presence of so called "hints." As these hints provide some random linear combinations of the secrets, they corresponds to possibly maliciously generated LWE samples in our context. That being said, we refer the reader to App. C for more details on LWE and its Ring and Module variants, and to App. D and E for more details on NMLWE and eMLWE respectively.

Non-uniform Learning with Errors (NLWE) Boneh et al. [17] introduced a variant of the learning with errors (LWE) problem in which the columns of $\mathbf{A}$ (i.e. the LWE sample points) are sampled from a *non-uniform* distribution η over $\mathbb{Z}_q^n$ called the *Non-Uniform Learning with Errors* (NLWE) problem, and showed that for suitable parameters, it is as hard as the basic LWE problem. In what follows, let k denote the dimension of the NLWE problem and let n denote the dimension of the LWE problem. Also, write η^m to denote m independent samples from the distribution η.

Definition 3 (Non-uniform Learning with Errors). *For an integer $q = q(k) \geq 2$, a noise distribution $\chi = \chi(k)$ over $\mathbb{Z}_q$, and a distribution η over $\mathbb{Z}_q^k$, the $\mathsf{NLWE}_{\mathbb{Z}_q,k,\chi,\eta}$ problem is to distinguish between two distributions:*

$$(\mathbf{A}, \mathbf{A}^t\mathbf{s} + \mathbf{e}) \textit{ and } (\mathbf{A}, \mathbf{u})$$

where $m = \text{poly}(k)$, $\mathbf{A} \leftarrow \eta^m : \mathbf{A} \in \mathbb{Z}_q^{k\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^k$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$.

Generally speaking, Boneh et al. [17] showed that, for any Probabilistic Polynomial Time (PPT) adversary, NLWE is as hard as LWE for any distribution η that is *coset samplable* defined as follows.

Definition 4 (Coset Sampleable Distributions [17]). *For integers $q = q(k)$ and $n = n(k)$ we say that a distribution $\eta = \eta(k)$ over $\mathbb{Z}_q^k$ is n-*coset sampleable *if there are two PPT algorithms* $(\mathsf{MatrixGen}, \mathsf{SamplePre})$ *such that:*

- $\mathsf{MatrixGen}(1^k, n, q)$ *outputs a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times k}$ and auxiliary data* $\mathbf{T}$.
- $\mathsf{SamplePre}(\mathbf{z} \in \mathbb{Z}_q^n, \mathbf{T})$ *outputs a $\mathbf{y} \in \mathbb{Z}_q^k$ satisfying* $\mathbf{My} = \mathbf{z}$.

Moreover, if $\mathbf{z}$ is distributed *uniformly* in $\mathbb{Z}_q^n$, then the output of $\mathsf{SamplePre}(\mathbf{z}, \mathbf{T})$ is distributed statistically close to η. That is, we have the following theorem: $\mathsf{NLWE}_{\mathbb{Z}_q,k,\chi,\eta}$ is as hard as $\mathsf{LWE}_{\mathbb{Z}_q,n,\chi}$ for any n-coset samplable distribution η.

Theorem 1 ([17]). *Let $\eta = \eta(k)$ be an n-coset samplable distribution. Suppose there is a PPT algorithm $\mathcal{A}$ that decides $\mathsf{NLWE}_{\mathbb{Z}_q,k,\chi,\eta}$ with advantage $\varepsilon = \varepsilon(k)$. Then, there is a PPT algorithm $\mathcal{B}$ that decides $\mathsf{LWE}_{\mathbb{Z}_q,n,\chi}$ with the* same *advantage $\varepsilon = \varepsilon(k)$.*

Following [17, Remark 4.4], we especially point out that the above holds even when $\mathbf{s}$ is distributed according to the error distribution rather than uniform.

Extended Learning with Errors (eLWE) The extended Learning with Errors assumption (eLWE) was first defined by O'Neill et al. in [45] in the context of proving security of deniable encryption from lattices (cf. Apon et al. [8]). eLWE has additionally been used by Alperin-Sheriff and Peikert [7] to show security theorems for circular and key-dependent message security of lattice identity-based encryption, as well as to show the classical hardness of LWE itself by Brakerski et al. [22]. The basic idea is to *extend* LWE security proofs, like

$$(\mathbf{A}, \mathbf{As} + \mathbf{e}) \overset{\text{comp}}{\approx} (\mathbf{A}, \mathbf{u}),$$

to a setting with additional auxiliary information z; i.e.,

$$(\mathbf{A}, \mathbf{As} + \mathbf{e}, z) \overset{\text{comp}}{\approx} (\mathbf{A}, \mathbf{u}, z).$$

In other words, eLWE claims that standard LWE computational hardness assumptions still hold, up to some very small concrete loss in bit-security — even in the presence of certain, additional (potentially non-uniform) "hints" z on the hidden terms $(\mathbf{s}, \mathbf{e})$ of an LWE equation. The typical form of such "eLWE leakage" z on the hidden values $(\mathbf{s}, \mathbf{e})$ takes the form of a chosen *inner product* $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e}\rangle)$, where some vector $\mathbf{z}$ is *adversarially chosen* and $\langle \mathbf{z}, \mathbf{e}\rangle$ is the derived hint on honestly generated error $\mathbf{e}$. Intuitively, inner products reveal very little information on the LWE secret $(\mathbf{s}, \mathbf{e})$. Define the torus $\mathbb{T} = \mathbb{R}/\mathbb{Z}$, then we have:

Definition 5 (Extended (Plain) Learning with Errors (eLWE)). *For $n, m, q, t \geq 1, \mathcal{Z} \subseteq \mathbb{Z}^m$, and a distribution χ over $\frac{1}{q}\mathbb{Z}^m$, the $\mathsf{eLWE}_{n,m,q,\chi,\mathcal{Z}}$ problem is as follows. The algorithm chooses $\mathbf{z} \in \mathcal{Z}$ and then receives the tuple*

$$(\mathbf{A}, \mathbf{b}, \langle \mathbf{e}, \mathbf{z}\rangle) \in \mathbb{T}_q^{n\times m} \times \mathbb{T}_q^m \times (1/q)\mathbb{Z}.$$

Its goal is to distinguish between two cases: First, $\mathbf{A} \in \mathbb{T}_q^{n \times m}$ *is chosen uniformly,* $\mathbf{e} \in (1/q)\mathbb{Z}^m$ *is chosen from* χ*, and* $\mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e} \bmod 1$ *where* $\mathbf{s} \in \{0, ..., q-1\}^n$ *are chosen uniformly. The second case is identical, except that* $\mathbf{b}$ *is chosen uniformly in* $\mathbb{T}_q^m$ *independently of everything else.*

Theorem 2. *For any* $n \geq 2, q \geq 1, \varepsilon \in (0, 1/2)$*, and* $\alpha, r \geq (\frac{\ln(2m(1+1/\varepsilon))}{\pi})^{1/2}/q$*, there is a reduction from* $\mathsf{LWE}_{n+1,m,q,\alpha}$ *to* $\mathsf{eLWE}_{n,m,q,(\alpha^2\xi^2+r^2)^{1/2},\mathcal{Z}}$ *that reduces the advantage by at most* $33\varepsilon/2$*, where* ξ *is a small constant factor.*

Next, we make a simple observation, that Plain eLWE – originally designated to account for leakage on the error term e – also handles the case of leakage on the secret vector s in certain scenarios.

Definition 6 (Extended Learning With Errors (eLWE, alt version: secret leakage)). *For* $n, q, t \geq 1, \mathcal{Z} \subseteq \mathbb{Z}^n$*, and a distribution* χ *over* $\frac{1}{q}\mathbb{Z}^n$*, the* $\mathsf{eLWE}_{n,q,\chi,\mathcal{Z}}$ *problem is as follows. The algorithm gets to choose* $\mathbf{z} \in \mathcal{Z}$ *and then receives the tuple*

$$(\mathbf{A}, \mathbf{b}, \langle \mathbf{s}, \mathbf{z} \rangle) \in \mathbb{T}_q^{n \times n} \times \mathbb{T}_q^n \times (1/q)\mathbb{Z}.$$

Its goal is to distinguish between two cases: First, $\mathbf{A} \in \mathbb{T}_q^{n \times n}$ *is chosen uniformly (conditioned on being invertible),* $\mathbf{s}, \mathbf{e} \in (1/q)\mathbb{Z}^n$ *is chosen from* χ*, and* $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod 1$*. The second case is identical, except that* $\mathbf{b}$ *is chosen uniformly in* $\mathbb{T}_q^n$ *independently of everything else.*

We claim that this version of eLWE (with leakage on the secret) asymptotically follows from standard eLWE (with leakage on the error) under the condition that $\mathbf{A}$ is invertible (modulo q), which requires $n = m$ in the previous statement of eLWE. In this case, $\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{s} + \mathbf{A}^{-1}\mathbf{e}$ exactly, and the "alt" theorem follows by swapping the roles of $(\mathbf{A}, \mathbf{A}^{-1})$ and $(\mathbf{s}, \mathbf{e})$. The distinction between $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n), \mathbf{e} \leftarrow \chi$ and $\mathbf{e}, \mathbf{s} \leftarrow \chi$ leads to at most a $(1/q)$ multiplicative loss in security (cf. [45]), which is clearly not tight. In instantiations from FrodoKEM, the proofs from the literature give a range of between $\ll 1$ and < 15 bits of security lost, and optimizing this concrete analysis is left to future work.

Weak Hint Learning with Errors (whLWE) Cheon et al. [25] (resp. Lee et al. [39]) defined Hint LWE (hLWE) and Weak-Hint LWE (whLWE) – with Liu et al. [41] providing a valuable discretization step in a recent preprint – as a potential enhancement of eLWE-type leakage-security properties, based on the following statistical insight about conditional Gaussian distributions:

Lemma 1 ([39], Lemma 4.8). *Let* D_s *denote a continuous Gaussian distribution with variance* s*. Let* $D_{c,s}$ *denote a continuous Gaussian distribution with center* c *and variance* s*. Then, for real numbers* $\sigma_1, \sigma_2 > 0$*, let* e *and* f *be random variables distributed as the Gaussian distributions* D_{σ_1} *and* D_{σ_2}*, respectively. Let* $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$*, then the tuple* $(e + f, e|(e + f))$ *is distributed as the joint conditional Gaussian distribution* $\left(D_\sigma, D_{L\sigma_1^2/\sigma^2, \sigma_1\sigma_2/\sigma}\right)$ *with* L *denoting* $e + f$*.*

We define the Weak-Hint LWE (whLWE) problem following [25,39]:

Definition 7 (Weak-Hint LWE). *Let n, q, and k be positive integers, $\sigma_1, \sigma_2 > 0$ be real numbers,* $\mathbf{z}$ *be a vector in* Domain *(with* Domain *arbitrary – but looking forward, we will choose* $\mathsf{Domain} := \mathbb{Z}_q^k$*) and S be a matrix in $\mathbb{Z}_q^{n \times k}$. The Weak-Hint LWE distribution, denoted $A^{\mathsf{whLWE}}_{n,q,\sigma_1,\sigma_2,k}(\mathbf{z}, S)$ is the distribution of* $(\mathbf{a}, S^t\mathbf{a} + \mathbf{e}, \langle \mathbf{z}, \mathbf{e} \rangle + f) \in \mathbb{Z}_q^n \times \mathbb{R}_q^k \times \mathbb{R}_q$*, where* $\mathbf{a} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow D_{\sigma_q}^k$*, and $f \leftarrow D_{\sigma_2}$, and where D is the secret distribution. Then the Weak-Hint LWE problem* $\mathsf{whLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ *is to distinguish, given arbitrarily many independent samples* $\mathbf{z} \leftarrow \mathsf{Domain}$ *chosen by an adversary, between* $A^{\mathsf{whLWE}}_{n,q,\sigma_1,\sigma_2,k}(\mathbf{z}, S)$ *for a fixed $S \leftarrow D$ and the distribution of* $(\mathbf{a}, \mathbf{u}, \langle \mathbf{z}, \mathbf{e} \rangle + f)$ *where* $\mathbf{u} \leftarrow \mathbb{R}_q$*.*

Using Lemma 1, Cheon et al. [25,39] are able to show the following security theorem, which we will leverage in what follows.

Theorem 3. *Let n, q, k be positive integers. Let $\sigma_1, \sigma_1', \sigma_2'$ be positive real numbers that satisfy $\sigma_1 = \sigma_1'\sigma_2'/\sqrt{(\sigma_1')^2 + (\sigma_2')^2}$. Finally, let D be a distribution over $\mathbb{Z}_q^{n \times k}$. Then there exists a polynomial-time reduction from* $\mathsf{LWE}^k_{n,q,\sigma_1}(D)$ *to* $\mathsf{whLWE}^k_{n,q,\sigma_1'\sqrt{k}\sigma_2'}(D)$*, which* exactly *preserves the adversary's advantage.*

3.5 Key Encapsulation Mechanism (KEM)

Generally speaking, a KEM allows a key agreement initiator (encapsulator) to generate and share a symmetric key via a ciphertext to the receiver (decapsulator). It can can be viewed as a special type of a public key encryption (PKE) algorithm [26], where the PKE is used to encrypt and decrypt a payload message, which is then used to derive said shared key. In this paper, we shall first refer to generic (black-box) KEMs, and then narrow down our KEM definition to an important subset that satisfies specific properties, to which our results apply. Ultimately, we aim at a definition encompassing implicitly rejecting KEMs that utilize an internal PKE from lattice LWE primitives (also RLWE and MLWE variants) with splittable public keys, such as FrodoKem and CRYSTALS-Kyber.

KEM Syntax: In line with the work by Cremers et al. [26], we adopt a modified KEM API definition that allows us to explicitly use a public key as input to the decapuslation routine. By doing so, we can easily state whether the same public key was used in both encapsulation and decapsulation.

Definition 8 (Key Encapsulation Mechanism (KEM)). *A key encapsulation mechanism* KEM *is a triple* $(\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$*:*

- $\mathsf{KGen}(1^\kappa) \rightarrow (pk, sk)$*: On input security parameter κ* **return** *key pair pk, sk (probabilistic algorithm),*
- $\mathsf{Encap}(pk) \rightarrow (C, K)$*: On input pk* **return** *ciphertext $C \in \mathcal{C}$ and key $K \in \mathcal{K}$ (probabilistic algorithm).*
- $\mathsf{Decap}(sk, pk, c) \rightarrow K$*: On input sk, pk and ciphertext C:* **return** *key K*

where $\mathcal{C}$ is the ciphertext space and $\mathcal{K}$ is the session key space for KEM*.*

```
$\mathsf{KGen}(1^\kappa)$
    $(pk, sk) \leftarrow \mathsf{PKE.KGen}(1^\kappa)$
    $s \leftarrow\$ \{0,1\}^{|s|}$
    $sk' \leftarrow (s, sk, pk, G_1(pk))$
    **return** $(pk, sk')$
```

```
$\mathsf{Encap}(pk)$
    $m \leftarrow\$ \mathcal{M}$
    $(r, k) \leftarrow G_2(G_1(pk), m)$
    $C \leftarrow \mathsf{PKE.Enc}(pk, m; r)$
    $K \leftarrow F(k, C)$
    **return** $(C, K)$
```

```
$\mathsf{Decap}(sk', pk, C)$
    $m' \leftarrow \mathsf{PKE.Dec}(sk, C)$
    $(r', k') \leftarrow G_2(G_1(pk), m')$
    $C' \leftarrow \mathsf{PKE.Enc}(pk, m'; r')$
    **if** $C' = C$ **then**
        **return** $K' \leftarrow F(k', C)$
    **else return** $K' \leftarrow F(s, C)$
```

Fig. 2. KEM with a public key as decapsulation input using a PKE with message space $\mathcal{M}$ and the hash functions G_1, G_2 and F - Adapted from [18,26,31].

Correctness of KEM: The correctness of a KEM means that the decapsulation algorithm $\mathsf{KEM.Decap}$ recovers the same shared key K produced by the encapsualtion algorithm $\mathsf{KEM.Encap}$ for a public key generated by the key generation algorithm $\mathsf{KEM.KGen}$. That is, however, except for a small probability over the space of key generation and encapsulation.

Definition 9 (KEM Correctness). *For every key pair* $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ *and every encapsulation* $(C, K) \leftarrow \mathsf{Encap}(pk)$*,* KEM *is* $(1 - \delta)$ *correct if:*
$\Pr[K' \neq K | K' \leftarrow \mathsf{Decap}(sk, pk, C)] \leq \delta$. KEM *is perfectly correct if* $\delta = 0$.

FO-KEM with Implicit Rejection: The Fujisaki–Okamoto (FO) transform introduced in [28,29] is a common construction used to elevate the security of an $\mathsf{IND\text{-}CPA}$ PKE to an $\mathsf{IND\text{-}CCA}$ KEM [26]. As opposed to explicitly rejecting KEMs, most NIST PQC KEMs use variants of FO with implicit rejection ($\mathrm{FO}^{\not\perp}_{m}$), where the decryption routine still outputs a random key from the same key space, even if the decryption fails. Throughout the rest of the paper, it is implied that any used KEM is an $\mathrm{FO}^{\not\perp}_{m}$ KEM as depicted in Fig. 2. This type of KEM construction can be viewed as a generalization for most LWE (and variants thereof) KEMs: it is used in CRYSTALS-Kyber, FrodoKEM, NewHope and Korea PQC competition winner SMAUG-T[3].

Security of KEM: The basic security of a KEM is defined in terms of indistinguishability of encapsulated keys from random keys. The anonymity and collision freeness of a KEM also play an essential role in the context of protocol design in general, and PAKEs in particular, as they capture properties beyond key security and protect against other distinguishing and collision finding attacks [26,31]. Further, we adopt the notion of public key uniformity for a KEM, which was introduced in [11] and [46] as the *Fuzziness* property, and later formalized in [4]. We deem this property vital in the context of PAKE dictionary attacks, as it prevents adversaries from distinguishing honestly generated public

[3] Our syntax later on will require Decap to take the public key as an explicit input, but Fig. 2 also reflects how many LWE-based KEMs have the public key as part of the secret key (to perform FO decapsulation).

keys from randomly generated ones, and consequently excluding wrong password guesses within a PAKE execution. The KEM security properties relevant for our construction are described in the experiments $\mathsf{Exp}^{\mathsf{IND\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}$, $\mathsf{Exp}^{\mathsf{PKU}^b}_{\mathcal{A},\mathsf{KEM}}$, $\mathsf{Exp}^{\mathsf{ANO\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}$ and $\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$ respectively in Fig. 3.

We note that KEMs with implicit rejection satisfying the collision freeness property ($\mathsf{SCFR\text{-}CCA}$) cannot satisfy the strong robustness property [31,26] ($\mathsf{SROB\text{-}CCA}$). Although SROB is defined via an almost identical experiment to SCFR, the fact that the latter rules out rejecting a decapsulated key through the provided KEM interface cannot be interpreted the same way as in the former. Whereas SROB guarantees that a ciphertext cannot decapsulate correctly under two different key pairs, SCFR only ensures that a ciphertext cannot produce the exact same (identical) key under two different key pairs.

Definition 10 (Key Uniformity of KEMs). *For a key encapsulation mechanism* KEM *with public key space* $\mathcal{PK}$, *we define the advantage of an adversary* $\mathcal{A}$ *in distinguishing honestly generated public keys from uniform random ones as* $\mathsf{Adv}^{\mathsf{PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *where* $\mathsf{Exp}^{\mathsf{PKU}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *is the security game in Fig. 3.*

Definition 11 ($\mathsf{IND\text{-}CCA}$ security of KEM). *For a key encapsulation mechanism* KEM *with session key space* $\mathcal{K}$, *define the advantage of an adversary* $\mathcal{A}$ *in distinguishing genuinely encapsualted session keys from uniform random ones as* $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *where* $\mathsf{Exp}^{\mathsf{IND\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *is the security game in Fig. 3.*

Definition 12 ($\mathsf{ANO\text{-}CCA}$ security of KEM). *For a key encapsulation mechanism* KEM *with public key space* $\mathcal{PK}$, *define the advantage of an adversary* $\mathcal{A}$ *in distinguishing public keys used to probabilistically encapsulate keys into ciphertexts as* $\mathsf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *where* $\mathsf{Exp}^{\mathsf{ANO\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *is the security game in Fig. 3.*

Definition 13 ($\mathsf{SCFR\text{-}CCA}$ security of KEM). *For a key encapsulation mechanism* KEM *with public key space* $\mathcal{PK}$ *and session key space* $\mathcal{K}$, *define the advantage of an adversary* $\mathcal{A}$ *in probabilistically generating a ciphertext* C *that decapsualtes correctly under two unique public keys and their corresponding secret keys as* $\mathsf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *where* $\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *is the security game in Fig. 3.*

3.6 Lattice KEMs with Splittable Public Keys

KEMs based on the lattice LWE problem and its variants have public keys of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$, where a matrix $\mathbf{A}$ defines the base of the lattice sampled from a random seed (fixed-length bit string usually referred to as ρ) that is appended (or prepended) to the public key. In the following we shall refer to such KEMs as a subset of the previously defined KEMs in Def. 8, which can be obtained from an LWE-based PKE with splittable public keys (Fig. 4).

Recall that in such KEMs, a PKE is used for encryption and decryption (Fig. 2), where the $\mathbf{A}$-part is used by the encapsulator both indirectly to encrypt a message m and directly to encrypt a randomness $\mathbf{r}$ (Fig. 4). When decapsulating a ciphertext C to obtain K, the decapsulator also learns the message m

```
$\mathsf{Exp}^{\mathsf{PKU^b}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $pk_1 \leftarrow\!\!\$\ \mathcal{PK}$
    $b' \leftarrow \mathcal{A}(pk_b)$
    $\textbf{return } b = b'$
```

```
$\mathsf{Exp}^{\mathsf{IND\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(C^*, K_0) \leftarrow \mathsf{Encap}(pk)$
    $K_1 \leftarrow\!\!\$\ \mathcal{K}$
    $b' \leftarrow \mathcal{A}^{\mathsf{D}(sk,pk,.)}(pk, C^*, K_b)$
    $\textbf{return } b = b'$
```

```
$\mathsf{Exp}^{\mathsf{ANO\text{-}CCA}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(pk_1, sk_1) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(C^*, K^*) \leftarrow \mathsf{Encap}(pk_b)$
    $b' \leftarrow \mathcal{A}^{\mathsf{D}(.,.)}(pk_0, pk_1, (C^*, K^*))$
    $\textbf{return } b = b'$
```

```
$\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
    $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
    $(pk_1, sk_1) \leftarrow \mathsf{KGen}(1^\kappa)$
    $C \leftarrow \mathcal{A}^{\mathsf{D}(.,.)}(pk_0, pk_1)$
    $K_0 \leftarrow \mathsf{Decap}(pk_0, sk_0, C)$
    $K_1 \leftarrow \mathsf{Decap}(pk_1, sk_1, C)$
    $\textbf{return } K_0 = K_1 \neq \bot$
```

Fig. 3. KEM Experiments $\mathsf{Exp}^{\mathsf{PKU}}_{\mathcal{A},\mathsf{KEM}}$, $\mathsf{Exp}^{\mathsf{IND\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$, $\mathsf{Exp}^{\mathsf{ANO\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$ and $\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$. The experiments IND-CCA, ANO-CCA and SCFR-CCA are adopted from [31]. The decryption oracle D takes a ciphertext and a bit that selects which secret key to use and disallows queries on the challenge ciphertext where applicable (ANO-CCA and IND-CCA).

encrypted using the internal PKE. This is unavoidable for any PKE-based KEM with splittable public keys, as the ciphertext containing m needs to be decrypted before deriving K from m [26]. Further, the encryption yields a ciphertext of the form $(\mathbf{u}, \mathbf{v})$ (Fig. 4), where $\mathbf{v}$ represents the encrypted message m chosen by the encapsulator using the $\mathbf{b}$-part, and $\mathbf{u}$ is the encryption of the encapsulators own uniformly chosen randomness. To decrypt $(\mathbf{u}, \mathbf{v})$, the decapsulator must use the same $\mathbf{A}$-part as the encapsualtor, otherwise they cannot decrypt the message m.

Adapting the definition in [10] we specify the subset of KEMs that are in scope in the rest of the document:

Definition 14 (LWE KEM with Splittable Public Keys (LS-KEM)). *Let* KEM *be a key encapsulation mechanism and $l \in \mathbb{N}$. We say that* KEM *is as an LWE-based KEM with splittable public keys, and call it* LS-KEM*, if:*

- *it uses the Fujusaki-Okamoto ($\mathsf{FO}^{\not\perp}_m$) transform as defined in Fig. 2,*
- *it utilizes an underlying* PKE *following the syntax defined in Fig. 4 (and therefore has public keys of the form $pk := (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$), and*
- *$\mathbf{A}$ is sampled from $\rho \in \{0, 1\}^l$ using a deterministic function G.*

4 Shortcomings of the IC Model in PAKE Proofs

The ideal cipher model arguably dates back to Shannon [53] and was formalized in PAKE settings in the BPR model [13]. It has been widely used in security

```
PKE.Enc($pk, m; r$)
$(\mathbf{A}, \mathbf{b}) \leftarrow pk$ // Parse $pk$ to lattice base and uniform sample
$(\mathbf{r}, \mathbf{e}_1, e_2) \leftarrow\!\!\$\, r$ // Sample secrets from randomness
$\mathbf{u} := \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$ // Direct usage of $\mathbf{A}$
$\mathbf{v} := \mathbf{b}^T\mathbf{r} + e_2 + m$ // Indirect usage of $\mathbf{A}$ as in $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ (simplified)
return $C := (\mathbf{u}, \mathbf{v})$
```

Fig. 4. Simplified exposition of encryption using an internal PKE in a lattice LWE KEM with splittable public keys of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$. Inputs to PKE.Enc are passed down from KEM as in Fig. 2.

proofs of cryptographic protocols instantiated with block ciphers (cf. Sec. 2). Thus, an IC serves modeling block ciphers (e.g., AES) as idealized objects similar to hash functions in the ROM with some exceptions [15]. Its main advantage is defining the behavior of a cipher, where each encryption maps to an independently random permutation that belongs to the same set of possible inputs.

Definition 15 (Ideal Cipher). *IC is an invertible permutation function $\mathcal{C} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, where each key $k \in \{0,1\}^\kappa$ defines a unique and independent random permutation $\mathcal{C}_k = \mathcal{C}(k, .)$ on $\{0,1\}^n$.*

Similar to a Random Oracle (RO), an IC provides oracle access for forward queries on the function $\mathcal{C}$, where an input of a pre-defined length is deterministically mapped to an output of the same length through a random permutation. However, the main difference here is that an IC also provides access for backward queries on $\mathcal{C}^{-1}$. Thus, it answers queries to both encryption and decryption and models invertible random permutations through an interface accessible in a protocol execution. Queries and responses to and from the IC oracle are modeled via lazy sampling and can be recorded in a list, which can then be used in steps of a protocol analysis to determine further action. This behavior is crucial in formulating sound proofs for three main reasons: 1. Keeping a record of both honestly and maliciously generated public values, ciphertexts, and secrets, 2. replacing values generated in a simulation with random values generated by a challenger, and 3. performing consistency checks across proof steps.

IC Issues in PAKEs: Narrowing down the issue at hand, and ignoring the QIC for a moment, the usage of IC for public key authentication in a PAKE models random permutations as values in the set of all possible public keys, such that $IC : \mathcal{K}_\pi \times \mathcal{PK} \to \mathcal{PK}$. This behavior ensures that decryption always yields a valid (i.e., possibly honestly generated) public key, regardless of the correctness of the password used for encrypting or decrypting this key. Further, the usage of IC in PAKEs ensures well-formed public keys, with the consequence that adversaries cannot use malicious keys to attack the security of the underlying asymmetric scheme. This results in problems when instantiating an IC with real-world ciphers (such as AES) to encrypt structured keys (such as MLWE-based)

Alice		**Bob**
$Password\ \pi$		$Password\ \pi'$
$sk_a, pk_a \leftarrow \mathsf{KEM.KGen}(1^\kappa)$		
$z \leftarrow pk_a \oplus \mathcal{H}_1(\pi)$		
	$\xrightarrow{z}$	
		$pk'_a \leftarrow z \oplus \mathcal{H}_1(\pi')$
		$(C_b, \overline{K'}) = \mathsf{KEM.Encap}(pk'_a)$
	$\xleftarrow{C_b}$	
$\overline{K} \leftarrow \mathsf{KEM.Decap}(sk_a, C_b)$		

Fig. 5. Simple NICE-PAKE

as follows: 1. Encrypting (structured) keys under one key and decrypting them under another key may very well result in values that do not exist in the key space (i.e., invalid keys). 2. Malicious (non-well-formed) keys may very well be used by adversaries to break the underlying scheme.

Practical Example: The following describes these problems in more detail. We devise a very simple PAKE construction as shown in Fig. 5, which resembles EKE or CAKE, but with no IC involved for that matter. In the authentication step, one would simply map the password using a XOF to a fixed length bit string and then XOR the output with the public key. At first glance, such constructions seem very attractive and promise a smooth security proof relying solely on well-studied KEM properties (e.g., IND-CCA and ANO-CPA). However, not relying on IC properties (e.g., bijection and record keeping) quickly reveals that this is at least questionable. We consider a malicious *Bob* who receives a masked pk as z as in Fig. 5. *Bob* may guess a password π' and observe the result of unmasking z. With probability bounded by the password dictionary size, it is likely that *Bob* guessed wrong. Assuming that not all z_i map to a valid pk_i under all π_i, he may observe an invalid pk', which may be detectable, e.g., by checking the (wrong) structure. Then, Bob can exclude this one guess from an offline dictionary and take another guess in the same active session, and thus instantly break the PAKE security. In the specific case of Kyber, the $\mathsf{KEM.KGen}$ algorithm generates a public key that mathematically resembles a vector of polynomials (i.e., the $\mathbf{b}-$part of the public key). Each polynomial has coefficients with values in $\mathbb{Z}_q$ where $q = 3329$ and is stored as 16-bit integer. This vector representation of the public key is then serialized into a byte array that maps two coefficients to three 8-bit integers. Unmasking the **b**-part may however yield a byte array that does not follow this structure, and thus to a polynomial with coefficients outside the range defined by the scheme parameters, which in turn leads to invalid public keys and thus enables detecting wrong password guesses. Considering FrodoKEM, this is not the case, as its public keys are (close to) indistinguishable from random bit strings and do not yield a detectable mathematical structure.

Now, we consider a malicious *Alice* and a KEM where all z_i map to valid pk_i under all π_i. Note that a malicious *Alice* is neither restricted to honestly use $\mathsf{KEM.KGen}$, nor to honestly calculate z. Thus, she may not necessarily be bounded by the security properties of the KEM that rely on honest key generation. Then, she could generate one or many malicious (non-well-formed) key pairs pk_i, sk_i and/or one or many malicious z_i and other data she may use to extract information from C_i. Note that these real-world problems vanish when modeling the key authentication using IC. An approach to prevent both attacks without an IC is restricting the PAKE to KEMs with splittable keys of the form $(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e})$ and enforcing an undetectable mismatch between $\mathbf{A}$ and $\mathbf{b}$ under wrong guesses of π for malicious Bob, and yielding an unusable C_b for malicious Alice. In the following, we investigate how this mismatch can be achieved.

5 Novel Security Properties for Splittable KEMs

In the following, we introduce the security properties for KEMs with splittable public keys ($\mathsf{LS\text{-}KEM}$) of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$, where $\mathbf{A}$ is deterministically sampled from a fixed-length random bit string seed ρ, and is thus chosen from $\mathbb{A}$, the set of uniform distributed lattice matrices (i.e., lattice base space).

5.1 Key Uniformity of KEM with Splittable Public Keys:

Based on the general KEM public key uniformity (cf. Sec. 3.5 Def. 10), we derive a similar notion for our previously defined $\mathsf{LS\text{-}KEM}$, the security experiment of which is shown in Fig. 6. We claim with this extension of the original notion, and the $\mathsf{UNI\text{-}PK}$ notion of Arriaga et al. [10], that an adversary viewing only a $\mathbf{b}$-part of the public key is not able to distinguish the $\mathbf{A}$-part embedded in it as $\mathbf{As} + \mathbf{e}$ from one that is chosen uniformly at random. The proof of $\mathsf{UNI\text{-}PK}$ for MLWE (e.g., Kyber) follows from [10], and can be similarly shown for LWE (e.g., FrodoKEM) via a trivial reduction from the decisional LWE assumptions.

Definition 16 (Key Uniformity of KEM with Splittable Public Keys). *For a key encapsulation mechanism* $\mathsf{LS\text{-}KEM}$ *with public key space* $\mathcal{PK}$ *and* $\mathbf{A}$*-part space* $\mathbb{A}$*, we define the advantage of an adversary* $\mathcal{A}$ *in distinguishing honestly embedded* $\mathbf{A}$*-part in public keys from uniformly random ones as* $\mathsf{Adv}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *where* $\mathsf{Exp}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *is the game in Fig. 6.*

Theorem 4. *Let* $\mathsf{LS\text{-}KEM}$ *be the key encapsulation mechanism defined in Def. 14, where* $\mathbf{A}$ *is uniformly sampled from the* $\mathbf{A}$*-part space* $\mathbb{A}$ *via a random fixed-length bit string seed* ρ *using a hash function* G *(e.g., SHAKE) modeled as a RO. Then the advantage of an adversary against* $\mathsf{SPLIT\text{-}PKU}$ *can be expressed as follows:*

$$\mathsf{Adv}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa) \leq \mathsf{Adv}^{\mathsf{UNI\text{-}PK}}_{\mathcal{B},\mathsf{KEM}}(1^\kappa)$$

```
$\mathrm{Exp}^{\mathsf{UNI\text{-}PK}^b}_{\mathsf{KEM,Split}}(\mathcal{A})$
1 : $(pk_0, -) \leftarrow \mathsf{KGen}(1^\kappa)$
2 : $(r_0, M_0) \leftarrow \mathsf{Split}(pk_0)$
3 : $(r_1, M_1) \leftarrow N_\kappa \times R_\kappa$
4 : $b' \leftarrow \mathcal{A}(r_b, M_b)$
5 : $\mathbf{return}\ b == b'$
```

```
$\mathsf{Exp}^{\mathsf{SPLIT\text{-}PKU}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
1 : $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
2 : $(\mathbf{A_0}, \mathbf{b}_0) \leftarrow pk_0$
3 : $\mathbf{A}_1 \leftarrow\!\!\$\ \mathbb{A}$
4 : $b' \leftarrow \mathcal{A}(\mathbf{A}_b, \mathbf{b}_0)$
5 : $\mathbf{return}\ b == b'$
```

Fig. 6. Key Uniformity for KEM and LS-KEM. The definition UNI-PK is adopted from [10] and preserving the original notation where N maps a bit string of a fixed length r to an element M in R, which corresponds to sampling $\mathbf{A}$ from a random seed.

Proof. Recall that $\mathbf{A}$ is deterministically sampled from a fixed-length bit string ρ using the function G. It is hence obvious that sampling from the exact same bit string will always yield the same $\mathbf{A}$-part using the same function modeled as a RO. It follows that parsing $\mathbf{A}_0$ from pk_0 in line 2 of $\mathsf{Exp}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ is indeed equivalent to parsing r_0 in line 2 of $\mathsf{Exp}^{\mathsf{UNI\text{-}PK}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$, and then sampling $\mathbf{A}_0 \leftarrow G(r_0)$. Similarly, sampling from another unique bit string will yield a different and unique $\mathbf{A}$-part, which respectively corresponds to line 3 of each of said experiments. Now suppose there is an adversary $\mathcal{A}$ that wins in the $\mathsf{Exp}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ game. We show an adversary $\mathcal{B}$ that succeeds against the $\mathsf{Exp}^{\mathsf{UNI\text{-}PK}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ game with approximately the same advantage. The reduction is trivial: $\mathcal{B}$ plays the the $\mathsf{Exp}^{\mathsf{UNI\text{-}PK}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ until line 3, and then samples $\mathbf{A}_0 \leftarrow G(r_0)$ and $\mathbf{A}_1 \leftarrow G(r_1)$ respectively. Then $\mathcal{B}$ forwards these values to $\mathcal{A}$ with either M_0 or M_1. $\mathcal{A}$ then outputs a bit b and forwards their guess to $\mathcal{B}$. Now, as the seed is simply a random bit string, the initial $\mathbf{b}$-part of the key in line 2 can be viewed as pseudo uniform (see [10], Theorem 2), and hence it is equivalent to M_b in line 4 of $\mathsf{Exp}^{\mathsf{UNI\text{-}PK}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$. It follows that if the adversary $\mathcal{A}$ succeeds in deciding SPLIT-PKU, then $\mathcal{B}$ can succeeds in deciding UNI-PK with approximately the same advantage giving Theorem 4. □

5.2 A-Part Secrecy for KEM with Splittable Public Keys:

Based on the definitions in Sect. 3.4, 3.5, and 3.6, we propose the assumption that using a different (non-matching) yet uniform $\mathbf{A}$-part in the encryption routine results in ciphertexts that cannot be decrypted correctly and hence cannot be used to reveal a non-related uniformly sampled $\mathbf{A}$. We formalize this assumption as the notion of KEM $\mathbf{A}$-Part Secrecy (A-SEC-CCA) depicted in Fig. 7, and provide proof of its correctness.

Definition 17 (Splittable KEM A-Part Secrecy). *For a key encapsulation mechanism* LS-KEM *with public key space* $\mathcal{PK}$*, and* $\mathbf{A}$*-part space* $\mathbb{A}$*, we define the advantage of an adversary* $\mathcal{A}$ *in distinguishing a random* $\mathbf{A}$*-part of a known public key used to probabilistically generate a ciphertext as* $\mathsf{Adv}^{\mathsf{A\text{-}SEC\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *where* $\mathsf{Exp}^{\mathsf{A\text{-}SEC\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ *is the game depicted in Fig. 7.*

```
$\mathsf{Exp}^{\mathsf{A\text{-}SEC\text{-}CCA}^{b}}_{\mathcal{A},\mathsf{KEM}}(1^{\kappa})$
    $(pk, sk) \leftarrow \mathcal{A}$
    $\mathbf{A}_0 \leftarrow\!\!\$\ \mathbb{A}$
    $\mathbf{A}_1 \leftarrow\!\!\$\ \mathbb{A}$
    $(C, K) \leftarrow \mathsf{KEM.Encap}'_{\mathbf{A}_b}(pk)$
    $b' \leftarrow \mathcal{A}(C, pk, sk, \mathbf{A}_0, \mathbf{A}_1)$
    $\textbf{return } b == b'$
```

```
$\mathsf{Exp}^{\mathsf{A\text{-}SEC\text{-}CCA}^{(b,\mathcal{D})}}_{\mathcal{A},\mathsf{KEM}}(1^{\kappa})$
    $(pk, sk) \leftarrow \mathcal{A}$
    $\mathbf{A}_0 \leftarrow\!\!\$\ \mathcal{D}$
    $\mathbf{A}_1 \leftarrow\!\!\$\ \mathbb{A}$
    $(C, K) \leftarrow \mathsf{KEM.Encap}'_{\mathbf{A}_b}(pk)$
    $b' \leftarrow \mathcal{A}(C, pk, sk, \mathbf{A}_0, \mathbf{A}_1)$
    $\textbf{return } b == b'$
```

Fig. 7. **A**-Part Secrecy Experiment (Left) and w.r.t. Password Dictionary (Right), where $\mathbb{A}$ and $\mathcal{D}$ denote the lattice matrix space and password dictionary space respectively, and $\mathsf{KEM.Encap}'_{\mathbf{A}}(pk)$ denotes invoking Encap explicitly with a matrix $\mathbf{A}_b$, i.e. ignoring the one embedded in pk (that was initially chosen by $\mathcal{A}$).

Remark on property idea. Unlike the standard notion of anonymity, we explicitly capture the case where an adversary controls the key generation. However, they are restricted by forwarding a masked seed ρ. We show that even with a prepared list of all possible passwords and hash outputs, an adversary cannot leverage a brute-force (offline dictionary) attack to guess the correct password. The trick here is three-fold: First, the masked seed is an unstructured random bit string. Unmasked, it will always lead back to a valid, but not a necessarily correct password. Second, the **A**-part sampled afterwards is used in the encapsulation regardless of the password guess. Assuming the adversary guessed wrong, this will create a mismatch in the encapsulation routine with overwhelming probability. Finally, an adversary cannot determine **A** in an honest ciphertext, and hence cannot relate it to a seed or a corresponding password guess.

Provable Concrete Security: In the following, we show that the **A**-Part Secrecy (A-SEC-CCA) property applies to LS-KEM as previously defined in Sec. 3.6 (Def. 14) that is instantiated with any FrodoKEM-style plain LWE PKE.

Theorem 5. *Let* LS-KEM *be the key encapsulation mechanism defined in Def. 14 where* **A** *is uniformly sampled from a random fixed-length bit string seed ρ using a hash function G (e.g., SHAKE) modeled as a RO, then the advantage of an adversary against* A-SEC-CCA *can be expressed as follows:*

$$\mathsf{Adv}^{\mathsf{A\text{-}SEC\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^{\kappa}) \leq 1/2 + \varepsilon_{\mathsf{whLWE}} + \mathsf{negl}(1^{\kappa})$$

Remark on concrete A-SEC-CCA*-security.* We will show that the concrete security loss using FrodoKEM (cf. Sec. 7), as presented in NIST PQC Round 3, is only a few bits of security. Specifically, it depends on concrete parameters of the associated whLWE problem. Full NIST Category 1, 3, and 5 concrete security levels can be achieved by mild re-parameterizations of FrodoKEM (within its well-defined design space) inducing a minimal loss in practical efficiency.

Remark on terminology. We refer to FrodoKEM in what follows, with the understanding that we either use FrodoKEM as defined but lose a few bits of security, or an alternative "GimliKEM,"[4] with slightly larger keys and ciphertexts, where the security proof equally applies but with no loss in concrete security.

Proof. We show $\mathsf{A\text{-}SEC\text{-}CCA}$ security using the common game-based approach, starting with the original security game $\mathbf{G}_0$, and transitioning through a series of experiments $\mathbf{G}_1, \mathbf{G}_2, ...$ while bounding the adversary's advantage. The final game will be independent of the challenge bit b, giving the theorem. Specifically, we aim to swap the portion of the ciphertext C that depends on $\mathbf{A}_b$ to uniform.

Game $\mathbf{G}_0$*:* This is the original game. The adversary chooses pk, sk arbitrarily. While $pk = (\mathbf{A}, \mathbf{b})$, only $\mathbf{b}$ is used in constructing C. Then, $\mathbf{A}_0$ is sampled from distribution $\mathcal{D}$, which is a well-spread depending on the entropy of the password π with support in the $\mathbf{A}$-space of the KEM, and $\mathbf{A}_1$ is sampled uniformly from the $\mathbf{A}$-space of the KEM. The challenger flips a bit b then constructs the challenge ciphertext C (ignoring the message component) of the form:

$$\begin{aligned} C &= (u, v) \\ u &= \mathbf{A}_b\mathbf{r} + \mathbf{e}' \\ v &= \mathbf{b}\mathbf{r} + \mathbf{e}'' \end{aligned}$$

The adversary's view in $\mathbf{G}_0$ is $(C, pk, sk, \mathbf{A}_0, \mathbf{A}_1)$ and outputs a guess b' as to which matrix $\mathbf{A}_b$ was used in constructing C.

Game $\mathbf{G}_1$*:* In this hybrid, we swap $\mathbf{A}_0$ to be sampled uniformly from the $\mathbf{A}$-space of the KEM. The rest of the experiment is the same.

Lemma 2 ($\mathbf{G}_0 \overset{\text{comp}}{\approx} \mathbf{G}_1$). *If G is modeled as a RO and Non-uniform LWE is $(1 - \mathsf{negl}(1^\kappa))$-hard, then* $\Pr_{\mathbf{G}_0}[\mathcal{A}\,\text{wins}] \leq \Pr_{\mathbf{G}_1}[\mathcal{A}\,\text{wins}] + \mathsf{negl}(1^\kappa)$.

Proof (Proof of Lemma 2). The difference in the games is the distribution of $\mathbf{A}_0$. In the first one, $\mathcal{A}$'s view of $\mathbf{A}_0$ is no worse than having sampled $\mathbf{A}_0$ as the output of a RO, which was given as input a randomly sampled password π. We claim that, conditioned on the entropy of π being well-spread and uniform distributed in $\mathcal{D}$, this distribution is n-coset samplable in a straightforward way. Following [17, Remark 4.4]: even when $\mathbf{r}$ is distributed according to the error distribution, NLWE is hard. Yet, $\mathcal{A}$ additionally has leakage on $\mathbf{r}$ in the form of $v = \mathbf{b}\mathbf{r} + \mathbf{e}''$. However, since $\mathbf{b}$ is chosen by the adversary *non-adaptively* – that is, before seeing $(\mathbf{A}_0, \mathbf{A}_1)$ – this leakage is independent of the n-coset samplability of $\mathbf{A}_0$. Therefore, n-coset samplability of $\mathbf{A}_0$ follows solely from modeling G as a RO, so distinguishing $\mathbf{G}_0$ and $\mathbf{G}_1$ is as hard as NLWE.

Game $\mathbf{G}_2$*:* In this hybrid, we swap the u-part of the challenge ciphertext C to uniform. The rest of the experiment is the same.

[4] Gimli was another member of the Fellowship, who was slightly larger than Frodo but had a nice axe.

Lemma 3 **($\mathbf{G}_1 \overset{\text{comp}}{\approx} \mathbf{G}_2$).** *For $\varepsilon_{\text{whLWE}} < 1$, if Weak-Hint LWE is $(1-\varepsilon_{\text{whLWE}})$-hard, then* $\Pr_{\mathbf{G}_1}[\mathcal{A}\,\text{wins}] \leq \Pr_{\mathbf{G}_2}[\mathcal{A}\,\text{wins}] + \varepsilon_{\text{whLWE}}$.

Proof (Proof of Lemma 3). The difference between the games is the distribution of u. In $\mathbf{G}_1$, we have $u = \mathbf{A}_b\mathbf{r} + \mathbf{e}'$, and in $\mathbf{G}_2$, we have u sampled uniformly at random. The adversary's view of the challenge ciphertext $C = (u, v)$ includes the hint $v = \mathbf{br} + \mathbf{e}''$ on the ciphertext's secret $\mathbf{r}$ that defines the LWE instance considered, where $\mathbf{b}$ is *arbitrarily chosen* by the adversary. The adversary chooses $\mathbf{b}$ and hands it to the challenger. The challenger flips a coin and returns either

$$\begin{aligned} C &= (u, v) \\ u &= \mathbf{A}_b\mathbf{r} + \mathbf{e}' \\ v &= \mathbf{br} + \mathbf{e}'' \end{aligned} \qquad \text{or} \qquad \begin{aligned} C &= (u, v) \\ u &= \text{uniform} \\ v &= \mathbf{br} + \mathbf{e}'' \end{aligned}$$

In the first case, the challenger simulates $\mathbf{G}_1$; in the second case, the challenger simulates $\mathbf{G}_2$. Therefore, the adversary's advantage in distinguishing the hybrids is its advantage $\varepsilon_{\text{whLWE}}$ against the Weak-Hint LWE problem with appropriate parameters (concrete parameters discussed in Sec. 7).

Completing the proof of Theorem 5. Finally, in $\mathbf{G}_2$, we note that the adversary's view is independent of the bit b in the $\mathsf{A\text{-}SEC\text{-}CCA}^{(b,\mathcal{D})}$ security experiment. Therefore, $\Pr_{\mathbf{G}_2}[\mathcal{A}\,\text{wins}] = 1/2$. Combining Lemmas 2 and 3, we have the probability that the adversary wins in the original A-SEC-CCA security game is $\Pr_{\mathbf{G}_0}[\mathcal{A}\,\text{wins}] \leq 1/2 + \mathsf{negl}(1^\kappa) + \varepsilon_{\mathsf{whLWE}}$, which gives Theorem 5. □

Corollary 1 **(A-SEC-CCA Security for MLWE).** *If Non-Uniform MLWE (NMLWE) is $(1-\mathsf{negl}(1^\kappa))$-hard and Extended MLWE (EMLWE) is $(1-\mathsf{negl}(1^\kappa))$-hard (cf. App. D and App. E), and if there exists a reduction from MLWE (MLWE) to Weak-Hint MLWE (whMLWE), which closely preserves the adversary's advantage against (MLWE), then it follows a similar proof for Theorem 5 for any LS-KEM instantiated from MLWE hardness assumptions (e.g., CRYSTALS-Kyber), the concrete parameters of which are discussed in Section 7.*

Collision Freeness for KEM with Splittable Public Keys Analogous to the formerly introduced A-SEC-CCA property, we suggest the notion of A-Part-Collision-Freeness A-CFR-CCA, or Splittable Collision Freeness.

Definition 18 **(Splittable KEM Collision Freeness).** *For a key encapsulation mechanism LS-KEM with public key space $\mathcal{PK}$ and $\mathbf{A}$-part space $\mathbb{A}$, we define the advantage of an adversary $\mathcal{A}$ in probabilistically generating a ciphertext C that decapsulates correctly under two unique $\mathbf{A}$-parts under the same $\mathbf{b}$-part and secret key sk as $\mathsf{Adv}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ where $\mathsf{Exp}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ is the game in Fig. 8.*

Theorem 6. *Let LS-KEM be the key encapsulation mechanism defined in Def. 14, where $\mathbf{A}$ is uniformly sampled from a random fixed-length bit string seed ρ using*

```
$\mathsf{Exp}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
  $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$
  $(\mathbf{A}, \mathbf{b}) \leftarrow pk$
  $\mathbf{A}^* \leftarrow\!\!\$\ \mathbb{A}$
  $C \leftarrow \mathcal{A}^{\mathsf{D}(\cdot,\cdot)}(\mathbf{b}, \mathbf{A}, \mathbf{A}^*)$
  $K \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}, sk, C)$
  $K^* \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}^*, sk, C)$
  return $K = K^* \neq \perp$
```

```
$\mathsf{Exp}^{\mathsf{A\text{-}CFR\text{-}CCA}^{\mathcal{D}}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
  $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$
  $(\mathbf{A}, \mathbf{b}) \leftarrow pk$
  $\mathbf{A}^* \leftarrow\!\!\$\ \mathcal{D}$
  $C \leftarrow \mathcal{A}^{\mathsf{D}(\cdot,\cdot)}(\mathbf{b}, \mathbf{A}, \mathbf{A}^*)$
  $K \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}, sk, C)$
  $K^* \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}^*, sk, C)$
  return $K = K^* \neq \perp$
```

Fig. 8. **A**-Part Collision Freeness (Left) and w.r.t. Password Dictionary (Right), where $\mathbb{A}$ and $\mathcal{D}$ denote the lattice matrix space and password dictionary space respectively, and $\mathsf{KEM.Decap}'(\mathbf{b}, \mathbf{A}, sk, C)$ denotes invoking Decap explicitly with a chosen matrix **A** not necessarily matching the one embedded in **b**.

a hash function G (e.g., SHAKE) modeled as a RO, then the advantage of an adversary against A-CFR-CCA *can be expressed as follows:*

$$\mathsf{Adv}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa) \leq \mathsf{Adv}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{B}_1,\mathsf{KEM}}(1^\kappa) + \mathsf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{B}_2,\mathsf{KEM}}(1^\kappa) + \mathsf{negl}(1^\kappa)$$

LS-KEM could be for example FrodoKEM-style Plain LWE KEM with parameters discussed in Sec. 7.

*Remark on the semantics of Collision Freeness (and its **A**-part variant).* Following Cremers et al. [26], SCFR-CCA for a KEM means that together, the output key and the ciphertext bind the public key; i.e., changing the public key changes the other two components by a sufficient distance, so that they both become statistically independent across different public keys. We argue that the **A**-part of the public key has enough entropy to change the output key in both KEM.Encap and KEM.Decap routines, so that the resulting keys are not systematically relatable to each other. The adversary's goal in the A-CFR-CCA game (Fig. 8) is hence to enforce a collision on ciphertexts so that they decapsualte to matching (i.e., identical) keys although encapsualted with different **A**-parts.

Proof. In order to show A-CFR-CCA security, we use the common game based approach, starting with the original security game $\mathbf{G}_0$ and transitioning to a new experiment $\mathbf{G}_1$ while bounding the adversary's advantage. This latter game will be shown hard based on the hardness of SCFR-CCA.

Game $\mathbf{G}_0$: This is the original game. Here, the challenger first generates pk, sk honestly. **A** is obtained from splitting **b**, and is thus uniformly chosen from the **A**-space of the KEM. Then, $\mathbf{A}^*$ is sampled from distribution $\mathcal{D}$, which is a well-spread distribution depending on the entropy of the password π with support in the **A**-space of the KEM. While $pk = (\mathbf{A}, \mathbf{b})$, only the **b** term is fixed. The adversary can thus freely choose whether to encapsulate using **A** or $\mathbf{A}^*$, and they use one of those with **b** to construct C. The challenger receives C from the adversary and invokes $\mathsf{KEM.Decap}$ twice using both $\mathbf{A}$-parts and obtains K and K^* respectively. The encapsulation routine of the KEM (as shown in Fig. 2) is:

$$\begin{aligned}
&(k, \mathbf{r}) = G_2(G_1(pk)||m) \\
&C \leftarrow \mathsf{PKE.Enc}(m, pk; \mathbf{r}) \\
&K \leftarrow F(C||k) \\
&\textbf{return } (C, K)
\end{aligned}$$

where G_1, G_2 and F are RO-modeled hash functions. The adversary's view in $\mathbf{G}_0$ is $(C, \mathbf{b}, \mathbf{A}, \mathbf{A}^*)$ and the challenger checks: K and K^* are equal and not rejected. For KEMs with implicit rejection the challenger only checks if $K = K^*$.

Game $\mathbf{G}_1$*:* In this step, we swap the $\mathbf{A}$-part in pk to be sampled uniformly from the $\mathbf{A}$-space of the KEM. The rest of the experiment is the same.

Lemma 4 ($\mathbf{G}_0 \overset{\text{comp}}{\approx} \mathbf{G}_1$). *If G is modeled as a RO and Non-uniform LWE is $(1 - \mathsf{negl}(1^\kappa)))$-hard, then*

$$\Pr_{\mathbf{G}_0}[\mathcal{A}\,\text{wins}] \leq \Pr_{\mathbf{G}_1}[\mathcal{A}\,\text{wins}] + \mathsf{negl}(1^\kappa)$$

Proof (Proof of Lemma 4). The difference in the games is the distribution of $\mathbf{A}$. In the first game, $\mathcal{A}$'s view of $\mathbf{b}$ does not allow him to identify the $\mathbf{A}$-part embedded in $\mathbf{b}$ by the hardness of Non-uniform LWE (similar to the proof of Lemma 2). Thus, with the output of the function G_1 being modeled as a RO, inputting a pseudo-random pk as per Definition 16 will yield a statistically independent value, with probability bound by the advantage against $\mathsf{Exp}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{B}_1,\mathsf{KEM}}$ for some adversary $\mathcal{B}_1$. Hence we derive

$$\Pr_{\mathbf{G}_0}[\mathcal{A}\,\text{wins}] \leq \Pr_{\mathbf{G}_1}[\mathcal{A}\,\text{wins}] + \mathbf{Adv}\,^{\mathsf{SPLIT\text{-}PKU}}_{\mathsf{KEM}}(\mathcal{B}_1)$$

Completing the proof of Theorem 6. Now, suppose there is an adversary $\mathcal{A}$ that wins with non-negligible probability in the $\mathsf{A\text{-}CFR\text{-}CCA}^{\mathcal{D}}$ game. We show an adversary $\mathcal{B}_2$ that succeeds against the $\mathsf{SCFR\text{-}CCA}$ security experiment with $(1 - \mathsf{negl}(1^\kappa))$-close to the same probability. The reduction is trivial: $\mathcal{B}_2$ runs $\mathsf{SCFR\text{-}CCA}$ to line 3, then hands $(\mathbf{A}_0, \mathbf{A}_1, \mathbf{b}_0)$ to $\mathcal{A}$. When $\mathcal{A}$ outputs C, $\mathcal{B}_2$ outputs C in line 3 of the $\mathsf{SCFR\text{-}CCA}$ game. From the hardness of standard LWE, their probabilities of success are $\mathsf{negl}(1^\kappa)$ close. From the hardness of $\mathsf{SCFR\text{-}CCA}$ via Grubbs et al. [31] and Cremers et al. [26], we get Theorem 6. □

Corollary 2 ($\mathsf{A\text{-}CFR\text{-}CCA}$ Security for MLWE). *If Non-Uniform MLWE (*NMLWE*) is $(1 - \mathsf{negl}(1^\kappa))$-hard, then it follows a similar proof for Theorem 6 for any* $\mathsf{LS\text{-}KEM}$ *instantiated from MLWE hardness assumptions.*

6 Security Analysis of NICE-PAKE

Definition 19 (The NICE-PAKE Protocol). *Let $\mathcal{D}$ be a password dictionary ($\pi \in \mathcal{D}$),* $\mathsf{LS\text{-}KEM}$ *be a key encapsulation mechanism with session key space $\mathcal{K}$ and seed length l as defined in Def. 14, and $\mathcal{H} : \mathcal{D} \to \{0,1\}^l$. The protocol **NICE-PAKE** denoted by Π is depicted Fig. 1 using these building blocks.*

Correctness: The protocol Π is correct if the final session key matches for every pair of honest and partnered users $(\mathcal{U}_i, \mathcal{U}_j)$ with matching passwords $(\pi_{\mathcal{U}_i}, \pi_{\mathcal{U}_j}) \in \mathcal{D}$ in an honest protocol execution $\Pi(\mathcal{U}_i, \mathcal{U}_j)$. Hence, the correctness of Π depends on the correctness of the underlying KEM and is bounded by the KEM correctness parameter δ as defined in Sec. 3.5.

6.1 Security Model

The security analysis follows the BPR [13] model for authenticated key exchange (AKE) [13]. It models an adversary in full control of the network, but without knowledge of the password, nor the secrets of honest users (e.g., the secret key in a KEM). The goal of an adversary is to distinguish real session keys as established by the protocol from random keys chosen independently, this is controlled by a `Test` query. As we believe most readers are familiar with this model, we defer the full model description to App. A. We also refer non-experts on PAKE security to App. B for a textual proof sketch and remarks on CPA vs. CCA security.

6.2 Formal Security Analysis

Theorem 7. *Let Π be the NICE-PAKE protocol defined in Fig, 1, instantiated over password dictionary $\mathcal{D}$ with a key encapsulation mechanism* $\mathsf{LS\text{-}KEM}$ *with splittable public keys and $\mathcal{H}$ modeled as a random oracle. The* BPR *advantage of an adversary $\mathcal{A}$ can then be bounded as follows, for adversaries $\mathcal{S}$, $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ and $\mathcal{B}_4$ that have similar running time to $\mathcal{A}$:*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{BPR}}_{\mathcal{A},\Pi}(1^\kappa) \leq{}& 2 \cdot (q_e^2 \cdot 2^{-|\rho|}) + |\mathcal{D}|^2 \cdot 2^{-|\rho|} + 2q_e^2 \cdot \mathsf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathcal{B}_1,\mathsf{KEM}}(1^\kappa) \\ &+ 2q_s^2 \cdot |\mathcal{D}|^{-1} \cdot (\mathsf{Adv}^{\mathsf{A\text{-}SEC\text{-}CCA}}_{\mathcal{B}_2,\mathsf{KEM}}(1^\kappa) + \mathsf{Adv}^{\mathsf{SPLIT\text{-}PKU}}_{\mathcal{B}_3,\mathsf{KEM}}(1^\kappa)) \\ &+ 2q_s^2 \cdot \mathsf{Adv}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{B}_4,\mathsf{KEM}}(1^\kappa) + (q_e^2 + q_s^2) \cdot 2\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{S},\mathsf{KEM}}(1^\kappa)\end{aligned}$$

where (q_e, q_s) are the number of $\mathcal{A}$'s `Execute` *and* `Send` *queries respectively, $|\mathcal{D}|$ is the password dictionary size and $|\rho| = \{0,1\}^l$ is the $\mathbf{A}$-part seed size of* KEM.

Proof: We provide a security proof via the common game-based approach. We gradually modify the original security game $\mathbf{G}_0$ through a series of experiments $\mathbf{G}_1$, $\mathbf{G}_2$,... showing that the success probability of an adversary cannot be significantly larger than $\frac{1}{2}$. The final game will be independent of the challenge bit b, giving the theorem. Specifically, we will aim to randomize the values transmitted on the network, which are visible for an adversary $\mathcal{A}$. We say the protocol Π is secure if the probability of $\mathcal{A}$ winning is bounded to a negligible quantity. We denote the probability of $\mathcal{A}$ winning in game $\mathbf{G}_i$ as $\Pr[\mathbf{G}_i]$.

Game $\mathbf{G}_0$*:* The original BPR security game.

Eliminating Collisions ($\mathbf{G}_1$ - $\mathbf{G}_3$):

Game $\mathbf{G}_1$ *(Hash Collisions):* Change Game $\mathbf{G}_0$ declaring the adversary to lose if there are distinct passwords $\pi \neq \pi' \in \mathcal{D}$ such that their hash values collide, $\mathcal{H}(\pi) = \mathcal{H}(\pi')$. The probability of this is bounded by the birthday bound as

$$\Pr[\mathbf{G}_0] \leq \Pr[\mathbf{G}_1] + |\mathcal{D}|^2 \cdot 2^{-|\rho|}.$$

This, in particular, means that we can assume that for each fixed value ρ, the function values $\rho \oplus \mathcal{H}(\pi)$, when varying over π, are all distinct.

Game $\mathbf{G}_2$ *(Unique pk):* Modify Game $\mathbf{G}_1$ by declaring the adversary to lose if there are two honest Alice-sessions (possibly with different passwords) which create the same initial message $(z, \mathbf{b})$. Note that each honest Alice-session picks a fresh random value ρ in each execution. Hence, the probability that this random value (shifted by $H(\pi)$ for Alice's password π in this session) matches the value $\rho' \oplus H(\pi')$ in another honest Alice-session is at most $q_e^2 \cdot 2^{-|\rho|}$. We derive

$$\Pr[\mathbf{G}_1] \leq \Pr[\mathbf{G}_2] + q_e^2 \cdot 2^{-|\rho|}.$$

Game $\mathbf{G}_3$ *(Unique C_b):* Modify Game $\mathbf{G}_2$ by declaring the adversary to lose if there are two honest Bob-sessions (possibly with different passwords) which create the same response message C_b. Note that the ciphertext C_b encapsulates a random message m from $\{0,1\}^{|\rho|}$. This implies that the ciphertext space is of size at least $2^{|\rho|}$ and the ciphertext is chosen uniformly among such encryptions of messages m. Hence, the probability that a ciphertext C_b matches any of the $i \leq q_e$ previously chosen ciphertexts is at most $i \cdot 2^{-|\rho|}$ and thus

$$\Pr[\mathbf{G}_2] \leq \Pr[\mathbf{G}_3] + q_e^2 \cdot 2^{-|\rho|}.$$

Simulating the KEM ($\mathbf{G}_4$ - $\mathbf{G}_5$):
Modify Game $\mathbf{G}_3$ by having any honest Bob-session receiving a message $(z, \mathbf{b})$ from an honest Alice-session use $C_{b,\mathrm{sim}}$ instead of C_b, where $(C_{b,\mathrm{sim}}, \overline{K}'_{\mathrm{sim}}) \leftarrow \mathsf{KEM.Encap}(pk_{\mathrm{sim}})$ for a fresh key pair $(sk_{\mathrm{sim}}, pk_{\mathrm{sim}}) \leftarrow \mathsf{KEM.KGen}()$, for an independently sampled $\mathbf{A}_{\mathrm{sim}}$. When the honest Alice-session which has sent $(z, \mathbf{b})$ (and which is unique by Game $\mathbf{G}_2$) receives this ciphertext $C_{b,\mathrm{sim}}$, it uses the original key $\overline{K}'$ that Bob had encapsualted instead of decapsulating $C_{b,\mathrm{sim}}$; the honest Bob-session also uses the key $\overline{K}'$ for the further steps. The indistinguishability of the hops from $\mathbf{G}_3$ to $\mathbf{G}_5$ is shown in two substeps.

Game $\mathbf{G}_4$ *(Simulate KEM Key):* First, we replace C_b by $C_{b,\mathrm{sim}}$ for honest Bob-sessions (with an honest Alice-session partner) *but use the key* $\overline{K}'_{sim}$ *generated by Bob in the further steps.* We claim that this modification is indistinguishable according to $\mathsf{ANO\text{-}CCA}$. First, note that there can be at most q_s honest Bob-sessions with an honest Alice-session, and we can apply the above modification step-by-step. Second, $\mathsf{ANO\text{-}CCA}$ ensures that adversaries cannot gain information by forwarding ciphertexts to other connected sessions. Via a hybrid argument, we lose a factor q_s in the argument but can, from now on, focus on a single pair of sessions in which we replace Bob's values. It then follows via a straightforward reduction $\mathcal{B}$ to $\mathsf{ANO\text{-}CCA}$, simulating the entire Game $\mathbf{G}_4$ (also picking the passwords of parties), and injecting the challenge key pairs (sk_0, pk_0), (sk_1, pk_1) from the anonymity game into the Alice-session, using pk_0 as pk_a resp. using pk_1 as pk_{sim} in the Bob-session. The latter loses a factor q_s in the security bound to guess the correct Alice-session. If this happens, we immediately know the honest Bob-session communicating with this unique Alice-session. The anonymity game also gives us a challenge ciphertext C^* and key K^*, created either under pk_0 or under pk_1. We use C^* as $C_{b,\text{sim}}$, and both parties use the key K^* as the session key. The reduction against anonymity outputs 1 if and only if adversary $\mathcal{A}$ wins the simulated game. Note that if C^*, K^* in the anonymity game are created under the public key pk_0, then the simulation corresponds perfectly to game $\mathbf{G}_4$. If $(C^*, K*)$ is created under pk_1 then the simulation corresponds precisely to our intermediate game. It follows

$$\Pr[\mathbf{G}_3] \leq \Pr[\mathbf{G}_4] + 2q_e^2 \cdot \mathsf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathcal{B}_1,\mathsf{KEM}}(1^\kappa)$$

for some adversary $\mathcal{B}_1$, where the factor 2 stems from switching from a left-or-right game $\mathsf{ANO\text{-}CCA}$ to a comparison between games $\mathbf{G}_3$ and $\mathbf{G}_4$.

Game $\mathbf{G}_5$ *(Use Original KEM Key):* The next step now is to switch back to the original keys $\overline{K}'$ instead of $\overline{K}'_{\text{sim}}$ in such honest Alice-Bob-interactions. This follows now from the $\mathsf{IND\text{-}CCA}$ security of the KEM, saying that one cannot distinguish which of the two keys is encapsulated in $C_{b,\text{sim}}$. It follows

$$\Pr[\mathbf{G}_4] \leq \Pr[\mathbf{G}_5] + 2q_e^2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{S},\mathsf{KEM}}(1^\kappa)$$

for some adversary $\mathcal{B}$. Now, in honest Alice-Bob interactions, the key $\overline{K}$ is entirely independent of the communication transcript.

Randomizing the Simulation ($\mathbf{G}_6$ - $\mathbf{G}_{10}$):
Sofar, the exchanged messages between Alice and Bob in honest sessions should already be distributed independently of the password π of both parties: Alice sends a random seed string and Bob responds with a ciphertext computed under a fresh public key. It remains to look into sessions where (a) Alice is malicious and interacts with an honest Bob-session, and (b) Bob is malicious but communicates with an honest Alice-session. Therefore, we aim at randomizing messages in response to adversarial query, which may be malicious and / or containing a trivial correct password guess. Here, we consider two *bad* events the simulation cannot prevent. The first one corresponds to the case where an adversary $\mathcal{A}$ forwards a pair $(z, \mathbf{b})$ that contains a correct password guess on z. Intuitively,

an adversary who guessed the password correctly would be able to distinguish between messages created by honest instances and ones randomized by the simulation. However, we rely on A-SEC-CCA to show that $\mathcal{A}$'s view on a pk used by an honest instance for encapsulating a key is indistinguishable from random. Therefore, we continue the simulation of the protocol and derive that $\mathcal{A}$'s advantage is bound to a factor of the number of queried sessions and random guessing over the password dictionary size $|\mathcal{D}|$. The second event corresponds to a Corrupt query placed by an adversary on connected honest instances.

Game $\mathbf{G}_6$ *(Randomize C_b):* Alter Game $\mathbf{G}_5$ by letting an honest Bob-session receiving $(z, \mathbf{b})$ which has not been created by an honest Alice-session use $C_{b,\mathrm{sim}}$ instead of ciphertext C_b, where $(C_{b,\mathrm{sim}}, \overline{K}'_{\mathrm{sim}}) \leftarrow \mathsf{KEM.Encap}_{\mathbf{A}_{\mathrm{sim}}}(pk)$ for an independently sampled $\mathbf{A}_{\mathrm{sim}}$. The honest Bob-session also uses the key $\overline{K}'$ for further steps. We claim that this modification is indistinguishable according to A-SEC-CCA. It then follows via a reduction $\mathcal{B}_2$ to A-SEC-CCA simulating the entire game $\mathbf{G}_{6a}$ and injecting the challenge $\mathbf{A}$-parts $(\mathbf{A}_0, \mathbf{A}_1)$ for the received $\mathbf{b}$ from the $\mathbf{A}$-Part Secrecy game into the Alice-session, using $\mathbf{A}_0$ as $\mathbf{A}_a$ resp. using $\mathbf{A}_1$ as $\mathbf{A}_{sim}$ in the Bob-session. Here, we lose the factor of ignoring trivial password guessing, as the Alice-session has an sk that actually decrypts under an encapsulation invoked by the $\mathbf{A}$-part obtained from the correct password. By doing so, an Alice-session cannot decapsulate the randomized ciphertext $C_{b,\mathrm{sim}}$, which in turn does not reveal the randomized key $\overline{K}'_{\mathrm{sim}}$, as $\mathsf{KEM.Decap}$ rejects implicitly by outputting a random key. On the other hand, the A-SEC-CCA game does not reveal the $\mathbf{A}$-part used for the encapsulation. The reduction against $\mathbf{A}$-Part Secrecy outputs 1 if and only if adversary $\mathcal{A}$ wins the simulated game. Hence, if the pair (C^*, K^*) in the $\mathbf{A}$-Part Secrecy game is created under the public key with $\mathbf{A}_0$, then the simulation corresponds perfectly to game $\mathbf{G}_5$. If (C^*, K^*) is created under the public key with $\mathbf{A}_1$, then the simulation corresponds to this game. It follows

$$\Pr[\mathbf{G}_5] \le \Pr[\mathbf{G}_6] + 2q_s^2 \cdot \mathsf{Adv}^{\mathsf{A\text{-}SEC\text{-}CCA}}_{\mathcal{B}_2,\mathsf{KEM}}(1^\kappa) \cdot |\mathcal{D}|^{-1}$$

for some adversary $\mathcal{B}_2$, where we lose a factor 2 for switching from a left-or-right game A-SEC-CCA to the comparison between games $\mathbf{G}_5$ and $\mathbf{G}_6$. Note that a Reveal query on an honest Bob-session does not carry any significance in this scenario, since $\mathcal{A}$ cannot decapsulate $C_{b,\mathrm{sim}}$, and did not guess the correct password. Thus they cannot compare Bob's key $\overline{K'}$ to a meaningful value.

Game $\mathbf{G}_7$ *(Corrupt Bob):* Declare the adversary to lose, if an honest Bob-session is marked *unfresh* after receiving $(z, \mathbf{b})$ which has not been created by a connected honest Alice-session. This game simulates the case where $\mathcal{A}$ obtains the password through *corrupting* an honest Bob instance in a protocol session. We define an *unbounded adversary*, whose interaction in this step does not affect the previous advantage. Thus, this game change is conceptual, and does not affect the security bound of the protocol. Recall that no test query can be placed by $\mathcal{A}$ on *unfresh* sessions as per the security model. Although an honest Bob would have otherwise unknowingly generated a ciphertext C_b based on $(z, \mathbf{b})$ and their honest password, obtaining this password via corruption will not aid $\mathcal{A}$ in decapsulating a ciphertext $C_{b,\text{sim}}$ with a non-matching $(sk, (\mathbf{A}_{sim}, \mathbf{b}))$ pair. That is since the ciphertext is simulated using an independently sampled $\mathbf{A}_{\text{sim}}$ as per $\mathbf{G}_6$, and we had hence already accounted for the adversarial advantage stemming from $\mathsf{A\text{-}SEC\text{-}CCA}$. This is justified through observing that $\mathcal{A}$'s advantage in $\mathbf{G}_6$ is bounded by the reduction to $\mathsf{A\text{-}SEC\text{-}CCA}$ and a multiplicative factor of guessing the attacked session over the password dictionary size. Thus, restricting $\mathcal{A}$'s interaction to a specific and known session (marked unfresh), eliminates the factors in the previous advantage term. Consequently, the adversary interacts with a single instance of the $\mathsf{A\text{-}SEC\text{-}CCA}$ experiment. It follows that $\Pr[\mathbf{G}_6] = \Pr[\mathbf{G}_7]$.

Game $\mathbf{G}_8$ *(Randomize z):* Change Game $\mathbf{G}_6$ by letting an honest Alice-session initiating the protocol use a randomly generated z_{sim} instead of an honestly masked seed ρ. Since we already accounted for collisions on z in game $\mathbf{G}_2$, we claim that this modification is indistinguishable according to $\mathsf{SPLIT\text{-}PKU}$ that an adversary $\mathcal{A}$ cannot distinguish between $\mathbf{A}$-parts resulting from splitting pk_a and ones embedded within the $\mathbf{b}$-part of the same public key. It then follows via a simple reduction $\mathcal{B}_3$ to $\mathsf{SPLIT\text{-}PKU}$ simulating game $\mathbf{G}_8$ and injecting the challenge $\mathbf{A}$-part $\mathbf{A}_1$ for the received $\mathbf{b}$-part from the $\mathsf{SPLIT\text{-}PKU}$ security game into the Bob-session. Here as well, we lose a factor for ignoring trivial password guessing over the password dictionary size. It follows

$$\Pr[\mathbf{G}_7] \le \Pr[\mathbf{G}_8] + 2q_s^2 \cdot \mathsf{Adv}_{\mathcal{B}_3,\mathsf{KEM}}^{\mathsf{SPLIT\text{-}PKU}}(1^\kappa) \cdot |\mathcal{D}|^{-1}$$

for some adversary $\mathcal{B}_3$, where we lose a factor 2 for switching from a left-or-right game $\mathsf{SPLIT\text{-}PKU}$ to the comparison between games $\mathbf{G}_6$ and $\mathbf{G}_8$.

Game $\mathbf{G}_9$ *(Randomize pk_a):* Change Game $\mathbf{G}_8$ by letting an honest Alice-session initiating the protocol use a randomly generated key pair $(sk_{\text{sim}}, pk_{\text{sim}})$ instead of an honestly generated one, where $(\mathbf{b}_{\text{sim}}, \rho_{\text{sim}}) \leftarrow pk_{\text{sim}}$ and an independently generated z_{sim} from $\mathbf{G}_8$. Upon receiving a response from a Bob-session, the Alice-session uses the previously randomized key pair to invoke $\mathsf{KEM.Decap}$. Since $\mathcal{A}$ encapsulated their own key $K_{\mathcal{A}}$ into $C_{\mathcal{A}}$, it will not match the decapsulated key $\overline{K}$ for an Alice-Session. This change is indistinguishable for $\mathcal{A}$ according to the collision freeness of KEM (except for the case of a `Reveal` query addressed in a following step). It then follows via a reduction $\mathcal{B}_4$ to $\mathsf{A\text{-}CFR\text{-}CCA}$ simulating game $\mathbf{G}_9$ and injecting the challenge $\mathbf{A}$-parts $(\mathbf{A}_0, \mathbf{A}_1)$ for the received public keys from the $\mathsf{A\text{-}CFR\text{-}CCA}$ game into the session using $\mathbf{A}_0$ as $\mathbf{A}_a$ resp. using $\mathbf{A}_1$ as $\mathbf{A}_{\text{sim}}$. Note that we don't lose a factor for trivial password guessing, as we already accounted for it in game $\mathbf{G}_8$. However, we still lose a factor q_s for guessing the correct session. The reduction against $\mathsf{A\text{-}CFR\text{-}CCA}$ outputs 1 if and only if $\mathcal{A}$ wins the simulated game. Hence, if the pair (C^*, K^*) in the $\mathsf{A\text{-}CFR\text{-}CCA}$ game is created under the public key with $\mathbf{A}_0$, then the simulation corresponds to game $\mathbf{G}_8$. If (C^*, K^*) is created under the public key with $\mathbf{A}_1$, then the simulation corresponds to this game. We derive

$$\Pr[\mathbf{G}_8] \leq \Pr[\mathbf{G}_9] + 2q_s^2 \cdot \mathsf{Adv}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{B}_4,\mathsf{KEM}}(1^\kappa)$$

Game $\mathbf{G}_{10}$ *(Corrupt Alice):* Declare the adversary to lose, if an honest Alice-session is marked *unfresh* after sending $(z, \mathbf{b})$ to a Bob-Session. This game simulates the case where an adversary $\mathcal{A}$ obtains the password through *corrupting* an honest Alice instances in a protocol session. As in $\mathbf{G}_7$, we define an *unbounded adversary*, whose interaction in this step does not affect the previous advantage. Thus, this game change is conceptual, and does not affect the security bound of the protocol. Similarly, we first recall that no test query can be placed by $\mathcal{A}$ on *unfresh* sessions as per the security model. Second, although an honest Alice would have otherwise sent a z that indeed reveals the correct seed for sampling the honest $\mathbf{A}$-part in her *pk*, and thus allowing $\mathcal{A}$ to generate a valid ciphertext that normally decapsualtes the same key for both users, obtaining the correct password via a corruption will not aid $\mathcal{A}$ in obtaining an identical key K^* with a non-matching (pk_a). That is since Alice's *pk* and her masked seed z are simulated using independently sampled values as per $\mathbf{G}_8$ and $\mathbf{G}_9$, we had already ruled out trivial password guesses on z, $\mathcal{A}$ commits to a single (non-adaptive) response $C_{\mathcal{A}}$, and we have already accounted for the adversarial advantage stemming from both SPLIT-PKU and A-CFR-CCA respectively. This is justified through observing that $\mathcal{A}$'s advantage in $\mathbf{G}_8$ and $\mathbf{G}_9$ is bounded by the respective reductions and a multiplicative factor for guessing the attacked session (and the password dictionary size in $\mathbf{G}_8$). Thus restricting $\mathcal{A}$'s interaction to a specific and known session (marked unfresh) eliminates the factors in the previous advantage terms. Consequently, the adversary interacts with a single instance of SPLIT-PKU and A-CFR-CCA. It follows that $\Pr[\mathbf{G}_9] = \Pr[\mathbf{G}_{10}]$.

Randomizing Session Keys ($\mathbf{G}_{11}$):
We showed that the remaining sessions should either be unfresh and cannot be tested, or if the sessions are fresh, then they must have already been replaced by some independent data on the network. It now remains to randomize the final sessions keys such that they yield values indistinguishable from real keys.

Game $\mathbf{G}_{11}$ *(Randomize K):* Finally, we replace the final session key with a key chosen independently at random from the key space $\mathcal{K}$. That is, on all connected instances and for all `Execute` and `Send` queries, rendering the keys independent of all previous messages and the password. We claim that this change is indistinguishable from games $\mathbf{G}_1$ through $\mathbf{G}_{10}$ for $\mathcal{A}$. In other words, a `Test` query may be placed by $\mathcal{A}$ at any point of the simulation (except on *unfresh* instances), and the tested key will always be indistinguishable based on the key security of KEM. Hence, the adversary's advantage is negligible and bounded by IND-CCA security of the KEM and the number of placed queries. It follows

$$\Pr[\mathbf{G}_{10}] \leq \Pr[\mathbf{G}_{11}] + 2q_s^2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{S},\mathsf{KEM}}(1^\kappa)$$

Where $\mathcal{S}$ is some adversary playing the indistinguishability game of KEM. Note that `Reveal` queries on Alice do not need to be explicitly handled by the simulation. Since the used KEM is implicitly rejecting, decapsulating on honest values will always yield a valid key $K^* \in \mathcal{K}$ that is with negligible probability bound to the key space size indistinguishable from a real key (covered by IND-CCA). Having randomized all values involved in encapsulating and decapsulating a session key over the network, and since a `Test` query is rendered unavailable for an *unfresh* instance, $\mathcal{A}$'s view on honest keys as well as their their own key is dependent only on their password guess being trivially correct (handled previously); and is indistinguishable from the randomization within the simulation.

Putting It All Together: Finally, all keys returned via a `Test` query are random values independent from the protocol simulation. Thus, an adversary cannot distinguish real keys from random ones. Their advantage is therefore upper-bounded by $\frac{1}{2}$ and the numbers of protocol executions, and the overall advantage by $\frac{1}{2}$ plus any losses collected throughout the games, giving Theorem 7. □

7 Remarks on Instantiations using Frodo and Kyber

Remarks on the concrete value of $\varepsilon_{\text{whLWE}}$ in Theorem 5:
We first mention why alternative reductions in the literature do not work here. There are two incomparable reductions for Extended LWE applicable to our use-case: one from O'Neill et al. in [45] that yields a $(1/q)$ multiplicative advantage loss per application, and one from Brakerski et al. [22] that yields a negligible advantage loss, up to a mild change in parameters. In the first case [45], we cannot afford to iterate a reduction with $(1/q)$ multiplicative loss over a column of v in FrodoKEM (that is an 8×8 matrix) for a total multiplicative loss of at least $(1/8q^8)$, as this would give concrete loss of bits of security at least $3 \cdot 16 \cdot 8 = 384$. In the second case [22], we do not see (currently) how to prove the analogue of their Claim 4.6: that is, constructing an unimodular $\mathbf{U}$ (with small largest singular value) so that removing $\mathbf{U}$'s left column yields that all the remaining columns are orthogonal to an arbitrary element of $\mathbb{Z}_q^k$. If such $\mathbf{U}$ can be demonstrated, this would provide an alternative, concretely-effective proof of security. We also do not see how to easily use the "noise lossiness" techniques of Brakerski and Döttling [20], as we require decisional hardness, but their proof only guarantees search hardness in the case of a (non-prime) power-of-2 integer modulus q as in FrodoKEM. However, using the techniques of Cheon et al. [25,39] and Liu et al. [41], we can calculate a small loss in concrete bit-security for a FrodoKEM-instantiation as follows: Following Theorem 3, the noise and error distributions of FrodoKEM are identical, so we have $\sigma_1' = \sigma_2'$. Then, $\sigma_1 = \sigma_1'/\sqrt{2} = \sigma_2'/\sqrt{2}$. Here, $k = 8$, so we have concrete security from Plain LWE, but with a $\sqrt{8} \cdot \sqrt{2} = 4$ multiplicative loss in variance. This yields that an instantiation of A-SEC-CCA security in our protocol from FrodoKEM-640 is as hard as if the variance were reduced from 2.8 to 0.7; or from FrodoKEM-1344 if the variance were reduced from 1.4 to 0.35. In the first case (Frodo-640), this is close to (but actually, slightly higher entropy) having a uniform $\{0, 1\}$-valued marginal distribution of the secret key and error terms in the view of the adversary after leakage. In the

second case, it's slightly lower entropy than uniform-binary secrets and errors, but the additional dimension of FrodoKEM-1344 may make up for this. In general, it would be better to re-parameterize a FrodoKEM-style Plain LWE KEM (call it "GimliKEM") with either slightly higher noise variance or slightly higher dimension (or both). One can consider such trade-offs in practice by using Albrecht et al.'s LWE Estimator tool [3] to find optimal choices.

A complete break (provable insecurity) of a Kyber-instantiation: We describe three different (very efficient) attacks by a Malicious Alice against a Kyber-instantiation of the protocol, which result in either a major leakage on or **complete recovery** of Bob's ciphertext randomness $\mathbf{r}$ in any given Kyber session. In what follows, recall the concrete Kyber set-up. Let $R = \mathbb{Z}[X]/(X^{256}+1)$ and $q = 3329$. Honest public keys are $(\mathbf{A}, \mathbf{b}) \in R_q^{k\times k} \times R_q^k$ for $k \in \{2, 3, 4\}$, where $\mathbf{A}$ is unrolled via G from a random seed $\rho_{\mathbf{A}}$ and so can be treated as having the uniform distribution over the coordinates of the Chinese Remainder Theorem (CRT) embedding, respectively the coefficient embedding, of each polynomial $\mathbf{A}_{i,j\in[k]} \in R_q$, and where $\mathbf{b} := \mathbf{As} + \mathbf{e}$ for $\mathbf{s}, \mathbf{e}$ drawn coefficient-wise from a binomial distribution η with parameter 2 (or parameter 3 for the secret $\mathbf{s}$ for Kyber-512) and thus having support in $\{-2, -1, 0, +1, +2\}$. Since $n = 256, q = 3329$, we have $n|(q-1)$ and so the ring R_q factors into $n/2$ distinct quadratic factors, meaning there is a CRT coordinate system defined that is isomorphic to $(\mathbb{Z}_q[X]/(X^2+1))^{n/2}$. Recall that X^2+1 is the 4th cyclotomic polynomial and that elements of the quadratic ring $\mathbb{Z}_q[X]/(X^2+1)$ are represented by two integers modulo q with slot-wise addition and simple, grade-school convolution-multiplication. One can map efficiently from the coefficient representation of polynomials in R_q to their CRT coordinate system and back using the Number Theoretic Transform (NTT) and inverse-NTT.

A first attack: When honest $\mathbf{A}$ *contains "correlated" zero divisors.* Recall that the protocol's security proof contemplates hints to leak on the secret $\mathbf{r}$ of the challenge ciphertext's MLWE secret in the form of v. To demonstrate that such hints "only reveal unstructured entropy," we generically need that $\mathbf{A}$ is invertible, so that we can consider the expression $\mathbf{r} + \mathbf{A}^{-1}\mathbf{e}'$ when attempting to swap u to uniform in order to show anonymity. While FrodoKEM's $\mathbf{A}$-part is invertible except with negligible probability, this is not the case for Kyber. In particular, if there exists an CRT-coordinate index $z \in [n/2]$ and fixed column index $j \in [k]$ so that for all R_q polynomials $\mathbf{A}_{i,j}$ (i.e. for all row indices $i \in [k]$)

$$\text{we have that, } \mathsf{NTT}(\mathbf{A}_{i,j})[z] = 0 \in \mathbb{Z}_q[X]/(X^2+1)$$
$$\text{that is, } \mathsf{CRT}(\mathbf{A}_{i,j})[z] = (0,0) \in (\mathbb{Z}_q)^2,$$

then $\mathbf{A}$ is not invertible in $R_q^{k\times k}$. In this event, the leakage produced by the hint v is "unexpectedly" algebraically-structured and will convey information about $\mathbf{r}$ and $\mathbf{e}''$ that is constrained to non-maximal ideals of the ring R_q. This event occurs with probability at least $\frac{k\cdot n/2}{q^{2\cdot k}}$, which is noticeable since k is constant. For e.g., Kyber-1024 this event occurs with probability at least $\frac{4\cdot 256/2}{3329^{2\cdot 4}} \approx 2^{-84.6} \gg 2^{-256}$.

"Patching" the first attack: Check $\mathbf{A}$*-invertibility.* A simple patch for this issue is for Honest Bob to attempt to invert any Kyber $\mathbf{A}$ before using it to construct his ciphertext in the protocol. While relatively efficient (if costing an undesirable practical slowdown), this especially has the downside of forcing the protocol to be "aware" of the particular lattice KEM being used.

A second attack: When Alice chooses $\mathbf{b}$ *(resp. each* $\mathbf{b}_i$*) to be a unit in the ring* R_q*.* A larger concern is if malicious Alice arbitrarily chooses $\mathbf{b}$ to be a "degenerate" element of R_q, such as a unit – for example, $1 \in R_q$. In this case, $\mathbf{b}$ vanishes completely in the expression $v = \mathbf{br} + \mathbf{e}''$, resulting in the hint $v = \sum_{i\in[k]} \mathbf{r}_i + \mathbf{e}''_i$, which dangerously exposes much of the entropy in $\mathbf{r}$ and $\mathbf{e}''$.

"Patching" the second attack: Check if $\mathbf{b}$ *is a unit (per slot).* Before Honest Bob constructs a ciphertext, he can test if elements of $\mathbf{b}$ are units in R_q with a straightforward calculation. However, concern should continue growing: How many bad cases could there be? When have you caught them all?

A third attack: When Alice chooses $\mathbf{b}$ *to be a "gadget vector" with moderately-sized radix.* Finally, consider when Malicious Alice chooses $\mathbf{b}$ to be a non-unit scalar. This is simplest to see in the Ring-LWE (i.e., rank 1) "case," when $\mathbf{b}, \mathbf{r}$, and $\mathbf{e}''$ are all simply polynomials in R_q. Since the support of each coefficient of $\mathbf{b}$ and $\mathbf{e}''$ are very small – e.g. in $\{-2, -1, 0, +1, +2\}$ – and the modulus $q = 3329$ is relatively large by comparison, consider if $\mathbf{b}$ is chosen as (say) $64 \in R_q$. Then, the hint $v = 64\mathbf{r} + \mathbf{e}''$, and the coefficients of $\mathbf{r}$ and $\mathbf{e}''$ can be read, directly, from the bit representation of the coefficients of v. In the case of Kyber-1024, Alice can choose $\mathbf{b}$ as a "gadget vector" such as $(4, 4^2, 4^3, 4^4) = (4, 16, 64, 256) \in R^4_{3329}$. Then, even with the vector-wise addition over rank 4, each of the coordinates of each of the $\mathbf{r}_i$ can still be read off directly from the bit representation of v, independent of $\mathbf{e}''$ – this is a complete break.

(speculatively) Tweaking Kyber / MLWE to Achieve A-SEC-CCA: Intuitively, the failure of A-SEC-CCA security for Kyber is due to the existence of many choices of $\mathbf{b} \in R_q^k$ such that computing the term $v = \mathbf{br} + \mathbf{e}''$, for $\mathbf{r}$ and $\mathbf{e}''$ supported on a small range like $[-2, +2]$, is *very far* from inducing wrap-around in the arithmetic modulo q. When reduction modulo q does not occur, then releasing an R_q-element as a hint can completely determine the secret values $(\mathbf{r}, \mathbf{e}'')$ of $\mathbf{br}+\mathbf{e}''$. Alternatively, if (sufficient) wrap-around modulo q occurs when computing $\mathbf{br} + \mathbf{e}''$, such an R_q-element cannot – "on its own" – convey the full entropy that went into the sampling of $(\mathbf{r} = (\mathbf{r}_1, ..., \mathbf{r}_k); \mathbf{e}'' = (\mathbf{e}''_1, ..., \mathbf{e}''_k)) \in R_q^k \times R_q^k$. Of course, in the algebraically-structured case, one not only needs to ensure reduction modulo q, but also that there is no special algebraic structure (e.g., confinement of the arithmetic to non-maximal ideals) that can be adversarially abused. For the sake of the science, we *speculate* that changing the distribution of Kyber (respectively: Module-LWE with rank $k \geq 2$ or preferably even higher rank) to use much larger supports and much higher entropy distributions for $\mathbf{r}$ and $\mathbf{e}''$ (as well as for $\mathbf{e}'$ in the u-part of the ciphertext, etc.) might lead to a A-SEC-CCA-secure implementation. Concretely, to ensure that the arithmetic in each coordinate wraps around modulo q for every non-adaptive choice of (not easily rejection-samplable) $\mathbf{b}$ that would otherwise "separate the coordinates" as in our gadget-based attack, it seems one needs to sample secret and error coordinates with weight at least up to $\approx \pm\sqrt{q}$, or around $\{-60, ..., +60\}$ for Kyber's modulus $q = 3329$. Note that this is a requirement of *larger secrets* than the asymptotically-suggested $\approx \sqrt{n}$ magnitude given by traditional security theorems in the literature [9]. The reason for this is that the typical Hermite Normal Form style of proof (that one can use the (M)LWE error distribution for the secret) inherently leverages that (M)LWE samples are built from uniform $\mathbf{A}$, which is not our case here. We emphasize that we have no security proof for this idea. In fact – we lack a theory for how to prove its security. While Plain LWE is known to be fairly robust against leakage [30,20] due to being able to use a leftover hash lemma argument that "plays with the dimension," proving a practically useful form of entropic security for algebraically-structured lattice cryptography is notoriously difficult [16,40]. We re-highlight this gap in the theory as an interesting open problem area, especially when considering practical instantiations similar to Kyber, which we leave for future research.

8 Conclusion

In this work, we addressed the issues accompanying quantum-resistant PAKE designs and constructions from lattice-based PAC KEMs and ideal ciphers. We presented our PAKE construction **NICE-PAKE**, which eliminates the usage of an IC in the public key authentication step through masking (XOR-ing) a purely uniform part of the splittable key in LWE KEMs. Our security proof relies on standard KEM properties and assumptions, as well as newly introduced ones (A-SEC-CCA, A-CFR-CCA, SPLIT-PKU). However, our construction indeed suffers from the lack of existence of standard LWE and MLWE KEMs, with which NICE-PAKE can be directly instantiated. To overcome this issue, we presented a discussion on concrete modifications for LWE Frodo-style and possible tweaks for MLWE Kyber-style KEMs.

The development and proof of **NICE-PAKE** has shed new light on the possibility of replacing troublesome idealized objects (i.e., the IC) in PAKE designs. Most importantly, we are now confident that the IC can be completely eliminated. Our work has indicated the precise properties needed for lattice-based KEMs in PAKEs, opening the door for dedicated constructions. Eliminating the IC highlighted the fact that neither uniform nor non-uniform public keys are per se favorable for building PAKEs. Both come with a respective (specific) security issue, that is idealized away by IC. From our perspective, FrodoKEM remarkably showed its strength being built from unstructured lattices and standard plain LWE assumptions.

A number of open questions remain. The first is to investigate the consequences of adjusting the variance value in FrodoKEM's error distribution, both for the core security and the performance of the scheme. The same goes also for Kyber/ML-KEM, where we suggest that using larger supports and higher entropy in the encapsulator's secrets could lead to secure splittable key applications. A similar question arises regarding the concrete overall security and performance, should such adjustments be taken. Regarding the usage of KEMs for protocol design, and disregarding size and performance differences, we found that there are major differences regarding the effective remaining security of different LWE-based KEMs in protocol contexts; i.e., the loss of bits of security under the different KEM properties (such as A-SEC-CCA) in the security proof substantially varies between the schemes. We deem deeper research on the security effects of swapping KEMs in higher level cryptographic constructions (such as PAKEs) an important future work. Observing the ongoing interest in purely generic KEM-based PAKE designs, it is also worth investigating whether our first attempt **Simple NICE PAKE** (cf. Sec. 4) could actually be achieved with a formal analysis based only on inherent KEM properties.

In order to do so, we suggest tackling properties directly aimed at the security of KEM public keys, such that some structured keys are statically far from uniform, yet still computationally close to uniform. However, and to the best of our knowledge, the only key property addressed so far in the literature concerns mainly the uniformity of keys. An interesting approach is to consider utilizing Threshold Secret Sharing schemes built from the same primitives as the KEMs, so that destroying the structure of the keys through "encryption" can be eliminated.

Acknowledgments

We thank Marc Fischlin, Hart Montgomery, Scott Simon, and Timm Lauser for useful discussions. This research work has been partly funded by the German Federal Ministry of Education and Research and the Hessian State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. Daniel Apon's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author. This work was supported in part by the MITRE Independent Research and Development Program. Approved for Public Release. Distribution Unlimited. Case Number 24-3386.

References

1. Abdalla, Michel and Eisenhofer, Thorsten and Kiltz, Eike and Kunzweiler, Sabrina and Riepel, Doreen: Password-Authenticated Key Exchange from Group Actions. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022. Lecture Notes in Computer Science, Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-15979-4_24
2. Alagic, G., Russell, A.: Quantum-secure symmetric-key cryptography based on hidden shifts. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 65–93. Springer (2017)

3. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology. Volume 9, Issue 3, Pages 169–203, ISSN (Online) 1862-2984, ISSN (Print) 1862-2976 DOI: 10.1515/jmc-2015-0016, October 2015
4. Alnahawi, N., Hövelmanns, K., Hülsing, A., Ritsch, S.: Towards post-quantum secure pake - a tight security proof for ocake in the bpr model. In: Cryptology and Network Security. pp. 191–212. Springer Nature Singapore (2024)
5. Alnahawi, N., Müller, J., Oupický, J., Wiesmaier, A.: A comprehensive survey on post-quantum TLS. IACR Communications in Cryptology **1**(2) (2024). https://doi.org/10.62056/ahee0iuc
6. Alperin-Sheriff, J., Apon, D.: Dimension-preserving reductions from LWE to LWR. Cryptology ePrint Archive, Paper 2016/589 (2016), https://eprint.iacr.org/2016/589
7. Alperin-Sheriff, J., Peikert, C.: Circular and kdm security for identity-based encryption. In: International Workshop on Public Key Cryptography. pp. 334–352. Springer (2012)
8. Apon, D., Fan, X., Liu, F.H.: Deniable attribute based encryption for branching programs from lwe. In: Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part II 14. pp. 299–329. Springer (2016)
9. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009. pp. 595–618. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
10. Arriaga, A., Barbosa, M., Jarecki, S., Skrobot, M.: C'est très chic: A compact password-authenticated key exchange from lattice-based kem. Cryptology ePrint Archive, Paper 2024/308 (2024), https://eprint.iacr.org/2024/308
11. Beguinet, H., Chevalier, C., Pointcheval, D., Ricosset, T., Rossi, M.: Get a cake: Generic transformations from key encaspulation mechanisms to password authenticated key exchanges. In: Applied Cryptography and Network Security: 21st International Conference, ACNS 2023, Kyoto, Japan, June 19–22, 2023, Proceedings, Part II. p. 516–538. Springer-Verlag, Berlin, Heidelberg (2023). https://doi.org/10.1007/978-3-031-33491-7_19
12. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 62–73 (1993)
13. Bellare, Mihir and Pointcheval, David and Rogaway, Phillip: Authenticated Key Exchange Secure against Dictionary Attacks. In: Advances in Cryptology – EUROCRYPT 2000, vol. 1807. Springer Berlin Heidelberg, Berlin, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_11, series Title: Lecture Notes in Computer Science
14. Bellovin, S.M. and Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy (May 1992). https://doi.org/10.1109/RISP.1992.213269
15. Black, J.: The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In: Fast Software Encryption: 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers 13. pp. 328–340. Springer (2006)

16. Bolboceanu, M., Brakerski, Z., Perlman, R., Sharma, D.: Order-lwe and the hardness of ring-lwe with entropic secrets. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 91–120. Springer International Publishing, Cham (2019)
17. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key homomorphic prfs and their applications. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013. pp. 410–428. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
18. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from lwe. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 1006–1018 (2016)
19. Bos, Joppe and Ducas, Leo and Kiltz, Eike and Lepoint, T and Lyubashevsky, Vadim and Schanck, John M. and Schwabe, Peter and Seiler, Gregor and Stehle, Damien: CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, London (Apr 2018). https://doi.org/10.1109/EuroSP.2018.00032
20. Brakerski, Z., Döttling, N.: Hardness of LWE on General Entropic Distributions. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology – EUROCRYPT 2020. pp. 551–575. Springer International Publishing, Cham (2020)
21. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014)
22. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. p. 575–584. STOC '13, Association for Computing Machinery, New York, NY, USA (2013), https://doi.org/10.1145/2488608.2488680
23. Canetti, R., Chen, Y.: Constraint-Hiding Constrained PRFs for NC^1 from LWE. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017. pp. 446–476. Springer International Publishing, Cham (2017)
24. Chaudhary, Dharminder and Kumar, Uddeshaya and Saleem, Kashif: A Construction of Three Party Post Quantum Secure Authenticated Key Exchange Using Ring Learning with Errors and ECC Cryptography. IEEE Access (2023)
25. Cheon, J.H., Kim, D., Kim, D., Lee, J., Shin, J., Song, Y.: Lattice-based secure biometric authentication for hamming distance. In: Information Security and Privacy: 26th Australasian Conference, ACISP 2021, Virtual Event, December 1–3, 2021, Proceedings. p. 653–672. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-90567-5_33
26. Cremers, C., Dax, A., Medinger, N.: Keeping up with the KEMs: Stronger security notions for KEMs and automated analysis of KEM-based protocols. Cryptology ePrint Archive, Paper 2023/1933 (2023), https://eprint.iacr.org/2023/1933
27. Ding, Jintai and Alsayigh, Saed and Lancrenon, Jean and Rv, Saraswathy and Snook, Michael: Provably Secure Password Authenticated Key Exchange Based on RLWE for the Post-Quantum World. In: Topics in Cryptology – CT-RSA 2017, vol. 10159. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-52153-4_11, series Title: Lecture Notes in Computer Science
28. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Annual international cryptology conference. pp. 537–554. Springer (1999)
29. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology **26**, 80–101 (2013)

30. Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: International Conference on Supercomputing (2010), https://api.semanticscholar.org/CorpusID:6166048
31. Grubbs, P., Maram, V., Paterson, K.G.: Anonymous, robust post-quantum public key encryption. In: Advances in Cryptology – EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 – June 3, 2022, Proceedings, Part III. p. 402–432. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-07082-2_15
32. Guo, S., Song, Y., Guo, S., Yang, Y., Song, S.: Three-party password authentication and key exchange protocol based on mlwe. Symmetry **15**, 1750 (09 2023). https://doi.org/10.3390/sym15091750
33. Hao, F., van Oorschot, P.C.: Sok: Password-authenticated key exchange – theory, practice, standardization and real-world lessons. In: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. p. 697–711. ASIA CCS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3488932.3523256
34. Hesse, J., Rosenberg, M.: PAKE combiners and efficient post-quantum instantiations. Cryptology ePrint Archive, Paper 2024/1621 (2024), https://eprint.iacr.org/2024/1621
35. Hongfeng Zhu and Xin Hao and Yang Sun: Elliptic Curve Isogenies-Based Three-party Password Authenticated Key Agreement Scheme towards Quantum-Resistant. J. Inf. Hiding Multim. Signal Process. **5** (2014), https://api.semanticscholar.org/CorpusID:16988408
36. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-meyer and merkle-damgård constructions. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part I 24. pp. 275–304. Springer (2018)
37. Katz, Jonathan and Vaikuntanathan, Vinod: Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In: Advances in Cryptology – ASIACRYPT 2009, vol. 5912. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_37, series Title: Lecture Notes in Computer Science
38. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography **75**(3), 565–599 (2015). https://doi.org/10.1007/s10623-014-9938-4
39. Lee, J., Kim, D., Kim, D., Song, Y., Shin, J., Cheon, J.H.: Instant privacy-preserving biometric authentication for hamming distance. Cryptology ePrint Archive, Paper 2018/1214 (2018), https://eprint.iacr.org/2018/1214
40. Liu, F.H., Wang, Z.: Rounding in the rings. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 296–326. Springer International Publishing, Cham (2020)
41. Liu, Z., Sotiraki, K., Tromer, E., Wang, Y.: Snake-eye resistance from LWE for oblivious message retrieval and robust encryption. Cryptology ePrint Archive, Paper 2024/510 (2024), https://eprint.iacr.org/2024/510
42. Lyu, Y., Liu, S., Han, S.: Universal composable password authenticated key exchange for the post-quantum world. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 120–150. Springer (2024)

43. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29. pp. 1–23. Springer (2010)
44. Majenz, C., Malavolta, G., Walter, M.: Permutation superposition oracles for quantum query lower bounds. Cryptology ePrint Archive, Paper 2024/1140 (2024)
45. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings 31. pp. 525–542. Springer (2011)
46. Pan, J., Zeng, R.: A generic construction of tightly secure password-based authenticated key exchange. In: Advances in Cryptology – ASIACRYPT 2023. pp. 143–175. Springer Nature Singapore, Singapore (2023)
47. Ravi, P., Howe, J., Chattopadhyay, A., Bhasin, S.: Lattice-based key-sharing schemes: A survey. ACM Comput. Surv. **54**(1) (2021). `https://doi.org/10.1145/3422178`
48. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. p. 84–93. STOC '05, Association for Computing Machinery, New York, NY, USA (2005). `https://doi.org/10.1145/1060590.1060603`
49. Regev, O.: Lattice-based cryptography. In: Annual International Cryptology Conference. pp. 131–141. Springer (2006)
50. Regev, O.: The learning with errors problem. Invited survey in CCC **7**(30), 11 (2010)
51. Santos, B.F.D., Gu, Y., Jarecki, S.: Randomized half-ideal cipher on groups with applications to uc (a) pake. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 128–156. Springer (2023)
52. Sato, S., Shikata, J.: So-cca secure pke in the quantum random oracle model or the quantum ideal cipher model. In: Cryptography and Coding: 17th IMA International Conference, IMACC 2019, Oxford, UK, December 16–18, 2019, Proceedings 17. pp. 317–341. Springer (2019)
53. Shannon, C.E.: Communication theory of secrecy systems. The Bell system technical journal **28**(4), 656–715 (1949)
54. Tang, Yongli and Li, Ying and Zhao, Zongqu and Zhang, Jing and Ren, Lina and Li, Yuanhong: Improved Verifier-Based Three-Party Password-Authenticated Key Exchange Protocol from Ideal Lattices. Security and Communication Networks **2021** (Nov 2021). `https://doi.org/10.1155/2021/6952869`, publisher: Hindawi
55. Taraskin, Oleg and Soukharev, Vladimir and Jao, David and LeGrow, Jason T.: Towards Isogeny-Based Password-Authenticated Key Establishment. Journal of Mathematical Cryptology **15**(1) (2020). `https://doi.org/10.1515/jmc-2020-0071`
56. Terada, S., Yoneyama, K.: Password-based authenticated key exchange from standard isogeny assumptions. In: Provable Security. pp. 41–56. Springer International Publishing, Cham (2019)
57. Unruh, D.: Towards compressed permutation oracles. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 369–400. Springer (2023)
58. U.S. National Institute of Standards and Technology (NIST): Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf` (2016)

59. U.S. National Institute of Standards and Technology (NIST): FIPS 203: Module-lattice-based key-encapsulation mechanism standard. `https://csrc.nist.gov/pubs/fips/203/final` (2024)
60. Wang, Jinhua and Chen, Ting and Liu, Yanyan and Zhou, Yu and Dong, XinFeng: Efficient Two-Party Authentication Key Agreement Protocol Using Reconciliation Mechanism from Lattice. In: International Conference on Security and Privacy in New Computing Environments. Springer (2022)
61. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39. pp. 239–268. Springer (2019)

A The Bellare-Pointcheval-Rogaway (BPR) Model

An adversary $\mathcal{A}$ is given a number of capabilities, or better said actions, called queries. $\mathcal{A}$ interacts with the protocol through these queries with so-called oracles that implement the protocol flow on behalf of honest parties called users (our friends *Alice* and *Bob*). $\mathcal{A}$ may assume any role they wish, i.e., either an active man-in-the-middle (malicious *Mallory*), or a passive connector forwarding messages between honest parties (eavesdropping *Eve*).

Sessions, Users, and Initialization: The game consists of an unlimited number of sessions between protocol instances. These sessions can be initiated by $\mathcal{A}$ in parallel or sequentially whenever they want. Each of these sessions chooses, upon initialization, two protocol instances $\mathcal{U}_i, \mathcal{V}_j$. Instances are users in a set $\mathcal{U}$ and can be viewed as initiators and receivers. An initiator $\mathcal{U}_i$ is assigned their own unique password $\pi_{\mathcal{U}_i} \in \mathcal{D}$ that is also known to a receiver $\mathcal{V}_j$.

Termination, Accepting, and Partnering: An instance may terminate either in an accepting state (key agreement is successful), or with no output in the case of rejection (key agreement is unsuccessful). A terminated instance may refuse to participate further in a protocol session. Two instances are partnered if they both terminate in accepting state with the same session key. An instance can however be terminated or accepting without being partnered. That is if an instance is ready to use a session key, but does not necessarily have any accepting partner.

Adversarial Model: $\mathcal{A}$'s goal is to distinguish between real and random session keys. At any point, $\mathcal{A}$ choose one session to test, which then outputs a bit b (random challenge). $\mathcal{A}$ outputs another bit b', and wins if they guessed correctly, i.e., if $b' = b$. We say that the adversary wins if on issuing a `Test` query for a user $\mathcal{U}_i$ that has terminated in accepting state (i.e. has a session key K) and no `Reveal` or `Corrupt` query has been issued to this user or any user partnered with them, $\mathcal{A}$ correctly guesses the bit selected in the `Test` query. Hence, the security here, same as in KEX security, lies within the ability of the protocol to guarantee the indistinguishability of honest session keys from random ones against any efficient adversary. $\mathcal{A}$ is given access to the following queries, and if necessary an oracle modeled as a RO for the hash function $\mathcal{H}$.

- `Execute`$(\mathcal{U}_i, \mathcal{V}_j)$: Execute an honest instance of the protocol that terminates in accepting state. $\mathcal{A}$ is given a full transcript of the execution (models eavesdropping).
- `Send`$(\mathcal{U}_i, m)$: Send a message to an honest user that causes them to proceed depending on their state (models impersonating attack)
- `Reveal`$(\mathcal{U}_i)$: Get the final session key for an accepting user, or the rejection symbol $\perp$ for a non-accepting user (models key leaking).
- `Corrupt`$(\mathcal{U}_i)$: Get the password $\pi_{\mathcal{U}_i}$ of a user in the weak-corruption model. In the strong-corruption model $\mathcal{A}$ may also view the internal state of the user (total break of an honest user).
- `Test`$(\mathcal{U}_i)$: Output a real or a random session key based on a bit flip made by the challenger. If a user is unfresh or not in accepting state, the test returns the rejection symbol $\perp$. Otherwise, if $b = 1$ it returns the real key K, else if $b = 0$ it returns a random key K'.

Without loss of generality, we assume that connected users in a session have the same password π from dictionary $\mathcal{D}$, and that the oracle chooses that same password upon initiation of a protocol instance Π_{ij} between two users $(\mathcal{U}_i, \mathcal{V}_j)$.

Freshness: An instance or user $\mathcal{U}_i$ is called **fresh**, if neither a `Reveal` query, nor a `Corrupt` query was placed by $\mathcal{A}$ upon it, or upon any other partnered user. Otherwise an instance is then called **unfresh**. This notion disallows $\mathcal{A}$ from winning the AKE security game trivially by testing sessions they previously revealed or corrupted their instances, and any instances partnered with them. The restriction made w.r.t. the `Corrupt` query allows for modeling (perfect) forward secrecy (PFS) and adaptive corruption.

B Proof Strategy

Passive Security. $\mathcal{A}$'s goal is to compromise the final session key K. Without actively interfering with protocol executions, they only have one option. That is to record a session transcript attempt to guess K through attacking the KEM. The oracle `Execute` implements an honest protocol session between two connected users. Through the provided interface, $\mathcal{A}$ may view the transmitted pk in its splitted form $(\mathbf{b}, z)$, and the correspondingly transmitted ciphertext C_b. Considering the available `Test` interface, $\mathcal{A}$ may also query either the real key or a random key resulting from one execution. Therefore, the security aimed at here is similar to the experiment of IND-CCA for a chosen KEM, however, without access to a decryption oracle. Hence, we claim that a KEM with only IND-CPA should also suffice. Intuitively, one would like the query `Test`$(\mathcal{U}_i, \mathcal{V}_j)$ to always output a random key through replacing the real session key K with a random one K^* chosen uniformly from the same key space $\mathcal{K}$. We will show that the $\mathcal{A}$'s advantage in distinguishing between the two keys is (1) determined through outputting a bit b (2) bound to the number of honestly executed sessions q_e they passively observe (3) bound to random password guessing on dictionary size plus a negligible advantage based on the IND-CPA security of the chosen KEM denoted by $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$.

Active Security. We would also like to show that $\mathcal{A}$ cannot distinguish between real or random keys in sessions they actively interact with. Here, they may, analogously to the passive case, also test a final session key for a partnered instance or for a terminated one in an accepting state. However, $\mathcal{A}$ has the possibility of choosing z, b (i.e. (pk)) and/or C at will through forwarding them via `Send` queries to initiated instances. Hence, we would like to especially capture the cases where they may attempt to trick an honest user to use a malicious pk, and then try to guess the correct password by relating C to the pk it was encapsulated with. The other possibility an adversary has is through choosing C and forwarding it to an honest user. Lastly, we recall that $\mathcal{A}$ may also place a `Reveal` and/or a `Corrupt` query on an instance. We primarily differentiate between the two cases here informally.

Adversary Impersonates Alice. Should $\mathcal{A}$ choose to impersonate *Alice*, they may initialize the protocol and start a receiving instance $\mathcal{V}_i$ via a $\texttt{Send(}\mathbf{b}, z\texttt{)}$ query with a pk of their choosing. In this case, they may guess (randomly choose) any π^* and use it with the seed ρ of their chosen pk as input for the authentication function f producing z, a quasi encryption of ρ under π^*. Intuitively, an honest Bob decrypts z, however, with the real password π, and gets ρ' to sample $\mathbf{A'}$, which is with probability depending on the dictionary size $|\mathcal{D}|$ not the same as $\mathcal{A}$'s choice. Bob then encapsulates pk using his reconstructed $\mathbf{A'}$ and gets (K, C_b) and sends C_b back. Since active instances only accept one pk per session, $\mathcal{A}$ is restricted to one password guess on that instance. Thus, we would first like to show that the advantage of $\mathcal{A}$ choosing the correct password on random guessing is negligible (trivial). Second, we will show that neither a malicious key pair (sk, pk) nor a received ciphertext C or the decapsulated key K could raise $\mathcal{A}$'s advantage in determining an honest pk leading them to the correct π.

Anonymity: Limiting $\mathcal{A}$'s advantage on relating C to any pk (observing decapsulations on ciphertexts without prior manipulation of a key pair) is somewhat simple and can be based on the anonymity property of the chosen KEM. In other words, should $\mathcal{A}$ be able to distinguish between public keys used for encapsulating a shared key into a chosen ciphertext with non-negligible advantage, it would break the anonymity of the chosen KEM. Therefore, $\mathcal{A}$'s advantage is bounded by a probability not better than random guessing.

$\mathbf{A}$-*Part Secrecy:* We would also like to account for the case where $\mathcal{A}$ chooses the public key pair at will, while also utilizing an offline dictionary attack on the public key space that is connected to the password dictionary. Here, we need to show that the chosen KEM does not allow randomly or maliciously chosen secret keys sk to correctly decapsulate on non-matching A-parts for known public keys pk. We rely on the assumption, that $\mathcal{A}$ is not capable of finding a second key pair that will decrypt a valid ciphertext and consequently decapsulate into a valid shared key correctly for a non-matching A-part resulting from reconstruction under the correct password, regardless of their choice for the key pair (sk, pk). Thus, regardless of the chosen sk, a decryption will always fail if a non-matching A-part was used by $\mathcal{A}$ and Bob. Here, $\mathcal{A}$ has the following options: 1) Iteratively change their chosen b part after receiving C in order to go through all possible A-part values resulting from reconstruction over the password dictionary, 2) iteratively changing sk used for decryption, and 3) combining both iterations to test all possible A-parts with all possible secret keys. Option 1 will yield successful with negligible probability bound by random password guessing, as the b-part contains the sk originally chosen by $\mathcal{A}$. Option 2 will yield unsuccessful, as the decryption will fail for all non-matching A-parts. Option 3 seems less promising, as the number of possible combinations will increase exponentially for all possible sk values iterated over password dictionary sizes. Nonetheless, such arguments break the purpose of a generic construction as they will force treating the KEM in white-box manner to formulate a reduction based on the encryption and decryption functions of the underlying PKE. To overcome this obstacle, we rely on the notion of **A**-Part Secrecy ($\mathsf{A\text{-}SEC\text{-}CCA}$) for a KEM with splittable public keys. Using this property of KEM, we can argue about the case where $\mathcal{A}$ freely chooses a key pair without opening up the KEM to apply a security reduction in the proof based on the KEM PKE, and thus maintain the generic construction. The reduction to this property will follow from the fact that using the seed in the authentication step binds an adversary to a subset of all possible key pairs limited by the dictionary size for reconstructing all possible $\mathbf{A}_i \in \mathbf{A}_\pi$. Meaning, that the adversary will commit to a password guess upon sending the value z, which will limit their choice for creating key pairs by the possible values of $\mathbf{A}$ that can be used in a pk, which in turn leads to the impossibility of choosing secret keys that are advantageous for the adversary in decapsulation and re-encryption.

Adversary Impersonates Bob. Should $\mathcal{A}$ impersonate Bob, they receive $(\mathbf{b}, z)$ from Alice. Not knowing π, $\mathcal{A}$ executes an offline dictionary attack to reconstruct all possible ρ_i values. $\mathcal{A}$ then sends some C to Alice, who in turn decapsualtes C with her honest sk. Here, a ciphertext can be any of the following: *Correct, incorrect, valid, or invalid.* Correctness means that C is a ciphertext that can be generated by the probabilistic encapsulation algorithm. Validity means that C will decrypt using the secret key corresponding to the public key used for encapsulation, without resulting in non-matching keys between Alice and Bob. Since the chosen KEM provides implicit rejection, Alice will get $K \neq K'$ and $\mathcal{A}$ would have failed in creating the same session key. $\mathcal{A}$ may try all possible $\mathbf{A}_i$ values. However, $\mathcal{A}$ will fail to create an identical pair (C, K) under any number of different keys, since the KEM encapsulation is probabilistic, and the KEM provides strong collision freeness. In other words, $\mathcal{A}$ will fail at creating a C that decapsulates into the same K as an honest Bob. Since we adopt a modified version of collision freeness, this attack is bound to the splittable collision freeness of the chosen KEM denoted by $\mathsf{Adv}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$. Should $\mathcal{A}$ choose to place either a Reveal or Corrupt query on Alice after sending C, they can obviously learn the final key decapsulated by Alice or her state, including her key pair. However, learning the final key has no additional value for $\mathcal{A}$, as the keys are expected to not match on non-matching A-parts. Further, corrupting an instance renders $\mathcal{A}$ unable to test the session connected to this instance, and therefore, they cannot win the security game by guessing the test bit b.

Adaptive Corruption and Perfect Forward Secrecy. $\mathcal{A}$ may leverage the Corrupt query to view the password of an honest user (weak corruption), or the complete internal state of an instance including the password (strong corruption). However, we recall that $\mathcal{A}$ can only win if they guess the test bit in a fresh accepting instance. Corrupting an instance will render it unfresh, and thus disqualify it as a possibly winnable one. It remains to discuss, if corruption could aid $\mathcal{A}$ in compromising fresh, yet non-partnered sessions. In the previously described scenarios, $\mathcal{A}$ could place such a query and easily see the honest KEM key pair of Alice. But even if they know how C was created, they will not be able to compromise other sessions since the KEM key pairs are always ephemeral (i.e., are newly generated for each fresh instance in a unique session) and the encapsulation routine is always probabilistic (i.e., each encapsulation yields statistically independent session keys and ciphertexts with overwhelming probability). They will also fail to forward ciphertexts to other connected sessions as per the KEM anonymity, and they will fail to create identical session keys for different sessions, based on the collision freeness of the KEM. Hence, any later on compromised protocol instance with an ephemeral KEM key pair will not affect previous ones.

C Plain, Ring and Module LWE

A lattice is a discrete subgroup under addition of $(\mathbb{R}^n, +)$ and can be described as a set of points in n-dimensional space with a periodic structure. Formally, given n-linearly independent vectors $v_1, ..., v_n \in \mathbb{R}^n$ the generated lattice is the set of vectors $L(v_1, ..., v_n) := \{\sum_{i=1}^{n} \alpha_i v_i | \alpha_i \in \mathbb{Z}\}$ constructed by all possible integer linear combinations. The basis of a lattice L can also be denoted by a base matrix $\mathbf{A}$ representing its basis vectors. Lattices are of great interest to cryptographers since there are several classical computational problems in lattices from which cryptosystems can be built [47]. The most prominent are the Shortest Vector Problem (SVP), Closest Vector Problem (CVP), the Smallest Integer Solution (SIS), and the Learning with Errors (LWE) problem. Nevertheless, public key cryptosystems from lattice problems are mostly based on worst-case reductions, making their choice of parameters stricter than average-case reductions [49]. Ultimately, even though lattices are defined in the Euclidean vector space $\mathbb{R}^n$, from a computational viewpoint, these problems are defined on integral lattices, whose representation is a matrix of integers. Hence, an irreducible polynomial f can replace a matrix base, and thus, a lattice can be defined as a special subset where all vectors form an ideal in a certain ring $\mathbb{Z}[x]/\langle f \rangle$. This led to extensive work on optimizing their key sizes in particular, and as a result several variations of lattice-based schemes dominated the NIST PQC process.

LWE. The decisional LWE problem [50] is basically to distinguish between (or find in the search variant) random linear equations from uniform equations after applying a small amount of noise. The Regev LWE-based public cryptosystem [48] is parametrized by a security parameter n, two integers (m, q), and a probability distribution $\mathcal{X}$ over $\mathbb{Z}_q$. It defines the private key sk as a small vector $\mathbf{s} \in \mathbb{Z}_q^n$. The public key pk consists of LWE samples $(\mathbf{a}_i, b_i)_{i=1}^m$ from the LWE distribution such that $\mathbf{b} \equiv \mathbf{as} + \mathbf{e} \bmod q$ with the secret $\mathbf{s}$, modulus q and the small noise (error parameter) $\mathbf{e} \in \mathbb{Z}_q$. Without an error $\mathbf{e}$ finding a secret $\mathbf{s}$ would be easy using Gaussian elimination, hence the error, which according to Regev resembles the problem of decoding random linear codes [50], yet was proven secure through a reduction to the SVP problem in [48].

RLWE. In the ring variant (RLWE), the ring $R_q = \mathbb{Z}[x]/\langle x^n + 1\rangle$ is defined as the ring of integer polynomials modulo $f(x)$ with n being a power of 2 and q a prime modulus so that $q = 1 \bmod 2n$. Consequently, the elements of R_q are residues modulo both $f(x)$ and q. They can thus be viewed as integer polynomials of degree less than n with coefficients in a set of canonical representatives in $\mathbb{Z}_q$. An RLWE sample is chosen as $(\mathbf{a}, \mathbf{b})$ with $\mathbf{b} \equiv \mathbf{as} + \mathbf{e} \bmod q$ where $\mathbf{a} \in R_q$ is chosen uniformly, $\mathbf{s} \in R$ is a fixed secret chosen together with $\mathbf{e} \in R$ from an error distribution [50]. Hence, an RLWE sample $(\mathbf{a}, \mathbf{b}) \in R_q \times R_q$ can replace n standard LWE samples $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, which reduces the size of the public key. The Lyubashevsky-Peikert-Regev RLWE cryptosystem [43] from ideal lattices encrypts a message $z \in \{0,1\}^n$ by using its bits as $\{0,1\}$ coefficients of a polynomial and choosing random elements $\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2 \in R$ and outputting the encryption of z as a pair $(\mathbf{u}, \mathbf{v}) \in R_q^2$ where $\mathbf{u} = \mathbf{ar} + \mathbf{e}_1 \bmod q$ and $\mathbf{v} = \mathbf{br} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot z \bmod q$. The decryption computes $\mathbf{v} - (\mathbf{u} \cdot \mathbf{s}) = (\mathbf{r} \cdot \mathbf{e} - \mathbf{s} \cdot \mathbf{e}_1 + \mathbf{e}_2) + \lfloor q/2 \rfloor \cdot m \bmod q$. The coefficients of the small terms $(\mathbf{r} \cdot \mathbf{e} - \mathbf{s} \cdot \mathbf{e}_1 + \mathbf{e}_2)$ are in the range $-q/4, q/4$. Thus, a bit in z is then decrypted as 0 if $\mathbf{b} - \langle \mathbf{a}, \mathbf{s} \rangle$ is closer to 0 than to $\lfloor q/2 \rfloor \bmod q$, otherwise it is 1.

MLWE. The module (MLWE) variant was first defined by Brakerski et al. [21] and further studied by Langlois and Stehlé [38]. In essence, MLWE also replaces the integers in $\mathbb{Z}$ by a ring of algebraic integers R of a number field K. The new component here is $M \subseteq K^d$, a module of R. Therefore, the parameter n is introduced as the degree of the number field, and the integer d denotes the module rank. With M being a rank d module and K of degree n, the resulting module lattice has the dimension $N = nd$. Hence, the MLWE problem generalizes both LWE and RLWE, as the RLWE variant is obtained if the module rank is $d = 1$. As a consequence, an MLWE sample is defined as $(\mathbf{a}, \mathbf{b})$ with $\mathbf{b} \equiv \mathbf{as} + \mathbf{e} \bmod R$ where $\mathbf{a} \in R_q^d$ is chosen uniformly, $\mathbf{s} \in R_q^d$ is a fixed secret, and $\mathbf{e}$ is sampled from an error distribution. The matrix representation for the lattice base is used when the number of samples m is fixed, which is denoted by $\mathbf{A} \in R_q^{m \times d}$.

D Non-uniform Module Learning with Errors (NMLWE)

For completeness, we define a module-lattice version of NLWE, adapted from Canetti and Chen [23, Lemma 2.13]

Definition 20 (Non-uniform Module Learning with Errors). *Let $\lambda \in \mathbb{N}$ be the security parameter. Let $m, m, q, p \in \mathbb{N}, \sigma$ s.t. $0 < \sigma < q$. Let $R = \mathbb{Z}[X]/(x^n + 1)$ for n a power of 2, γ_σ be a distribution over $R^{m \times m}$ parameterized by σ, and χ_σ be distribution over R parameterized by σ, with $||\gamma_\sigma||, ||\chi_\sigma|| \leq \sigma\sqrt{m}$.*

The NMLWE *problem asks to distinguish between samples $(\mathbf{D}, \mathbf{KD} + \mathbf{E}) \in (R^{m \times m} \times R^{1 \times m})$ from $(\gamma \times U(R_q^{1 \times m}))$, where $\mathbf{D} \leftarrow \gamma_\sigma$ is possibly non-uniform, $\mathbf{K} \leftarrow U(R_q^{1 \times m}), \mathbf{E} \leftarrow \chi_\sigma^{1 \times m}$.*

We cite Canetti and Chen's [23, Lemma 2.13] security claim for NMLWE.

Theorem 8. *For $R = \mathbb{Z}[X]/(X^n + 1)$ where n is a power of 2, set parameters $m \geq 2n \log(q), \sigma = \omega(\sqrt{n \log(q)})$, and set discrete Gaussian distributions $\gamma_\sigma = D^{1 \times m}_{R^m, \sigma}, \chi_\sigma = D_{R,\sigma}$.*

Then the hardness of NMLWE follows from the hardness of Plain LWE.

On the weakness of NMLWE security proofs. We remark that – to date – NMLWE has only been shown secure in contexts where one can choose a *high rank*; that is, $m \geq 2n \log(q)$. When the module-rank is appreciably lower than linear in the security parameter (especially in the case of Kyber, which uses rank 2, 3, or 4), known security reduction are vacuous.

To wit, we speculate that improving such security proofs to allow for arbitrarily small, constant rank would imply general-purpose program obfuscation $i\mathcal{O}$ for P/poly from standard lattice assumptions. We view this as negative evidence for obtaining such proof in the near term.

E Extended Module Learning with Errors (eMLWE)

For completeness, following Alperin-Sheriff and Apon [6], we define a module-lattice version of eLWE as follows:

Definition 21 (Extended Module Learning with Errors (eMLWE)). *For security parameter $1^\kappa \in \mathbb{N}$, let $n = n(\lambda)$ be an integer dimension, let $f(x) = x^d + 1$ where $d = d(1^\kappa)$ is a power of 2, let $q = q(1^\kappa) > 2$ be an integer, let $R = \mathbb{Z}[x]/(f(x))$ and $R_q = R/qR$, and let $\chi = \chi(1^\kappa)$ be a distribution over R. Then, the* $\mathsf{eMLWE}_{n,d,f,q,\chi}$ *problem is to distinguish between the following two distributions:*

In the first distribution, one samples $(\mathbf{a}_i, b_i)$ uniformly from R_q^{n+1}. The adversary chooses $z_i \in R_q^n$ and then, for u_i uniformly from R_q^n, is presented with:

$$(\mathbf{a}_i, b_i)_i, (z_i)_i, \mathrm{Tr}(\langle z_i, u_i \rangle)_i).$$

In the second distribution, one first draws $\mathbf{s} \leftarrow R_q^n$ *uniformly, and samples* $(\mathbf{a}_i, b_i) \in R_q^{n+1}$ *by sampling* $\mathbf{a}_i \leftarrow R_q^n$ *uniformly,* $e_i \leftarrow \chi$*, and setting* $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i$*. The adversary chooses* $z_i \in R_q^n$ *and then is presented with:*

$$(\mathbf{a}_i, b_i)_i, (z_i)_i, \mathrm{Tr}(\langle z_i, s_i \rangle)_i),$$

where the use of the trace function $\mathrm{Tr}(\cdot)$ *explicitly permits the adversary to request any* $\mathbb{Q}$*-linear function of the secret vector, as viewed in the coefficient embedding, for its hints.*

We cite Alperin-Sheriff and Apon's [6, Lemma 3.3] security claim for eMLWE.

Theorem 9. *There is a reduction from* $\mathsf{LWE}_{d,w,q,\alpha}$ *to either of:*

1. *Following [7]:* $\mathsf{eMLWE}_{d,w,q,\alpha,k}$ *that reduces the advantage by at most* q^k*, multiplicatively.*
2. *Following [22]:* $\mathsf{eMLWE}_{d+k,w,q,(\alpha^2+r^2)^{1/2},k}$*, where* $r \geq \omega(\sqrt{\log(w)})$*, which reduces the advantage by at most a negligible amount, additively – given the change in dimension and error rate.*

On the weakness of eMLWE security proofs. A key gap in the security offered by eMLWE is that it considers leakage of (potentially, traces of) inner products $\langle z_i, u_i \rangle$ over the various coordinates i of the vectors of polynomials in the problem statement. Fundamentally, there is no compression of dimension as in (Plain) eLWE, so an entire ring element leaks (compared to a single integer modulo q for Plain eLWE), which – to date – prevents many applications in entropic security arguments.