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

Nouri Alnahawi[1,5,6,7], Jacob Alperin-Sheriff[2], Daniel Apon[3], Gareth T. Davies[4] and Alexander Wiesmaier[1,5,6]

[1] Hochschule Darmstadt, Germany
[2] Independent Researcher, USA
[3] Anduril Industries, USA
[4] NXP Semiconductors, Belgium
[5] National Research Center for Applied Cybersecurity ATHENE, Germany
[6] European University of Technology, European Union
[7] University of Regensburg, Germany

Abstract. We present the **N**o **IC** **E**ncryption (**NICE**)-**PAKE**, a (semi)-generic symmetric Password Authenticated Key Exchange (PAKE) framework providing a quantum-safe alternative for the *Ideal Cipher* (IC), utilizing simpler cryptographic components for the authentication step. We give a formal proof in the Bellare-Pointcheval-Rogaway (BPR) model, which we support by introducing the notions of **A**-Part Secrecy (A-SEC-CCA), Splittable Collision Freeness (A-CFR-CCA) and Public Key Uniformity (SPLIT-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. Finally, we show how to obtain a secure PAKE from our construction with concrete parameter choices for lattice KEMs.

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

1 Introduction

Password Authenticated Key Exchange (PAKE) protocols establish a session key between communication parties over insecure channels utilizing an asymmetric key agreement scheme combined with password-based authentication. In the absence of static public keys or certificates, PAKEs provide an alternative 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 [HvO22], the very first PAKE, Encrypted Key Exchange (EKE) [BM92], 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 be correctly used, by a receiver who has the same password. Thus, using the same password serves as authentication for both sides of the key agreement. Still, many PAKEs rely on quantum-vulnerable hardness assumptions such as (C)DH [HvO22], which has led to increased attention on generic PAKEs from PQC KEMs (cf. Sec. 1.4).

E-mail: nouri.alnahawi@h-da.de (Nouri Alnahawi), jacobmas@gmail.com (Jacob Alperin-Sheriff), dapon@anduril.com (Daniel Apon), garethhthomas.davies@nxp.com (Gareth T. Davies), alexander.wiesmaier@h-da.de (Alexander Wiesmaier)

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2026-05-15.

1.1 Motivation

Several generic KEM-based PQC PAKEs have been proposed in the literature [MRR20, BCP+23, PZ23, AHHR24, SGJ23, JRX25, ABJS24, ABJ25, HHKR25], with a noticeable focus on lattice-based schemes from Module Learning with Errors (MLWE). Despite some differences, most works rely on the *Ideal Cipher* (IC) model in formal security analysis. The IC models a random block cipher as being chosen uniformly from the set of all possible block ciphers [Bla06] and allows for defining a deterministic behavior on such ciphers. IC-modeled block ciphers have significant drawbacks in PAKEs that affect confidence in generic constructions (cf. Sec. 2). The public key permutations obtained from an IC need to be statistically close to uniform [HR25], i.e., all outputs yield possible valid public keys. However, ML-KEM (aka CRYSTALS-Kyber) does not satisfy this requirement, since its keys are only computationally indistinguishable from uniform [HR25]. Thus, encrypting an ML-KEM public key may yield an invalid IC output distinguishable from uniform, which leads to multiple password guesses in one PAKE session. Further, it is non-trivial to instantiate an IC as a block cipher over a group domain [ABJS24] and there are still no known adaptations for the IC model to deal with quantum adversaries (cf. Sec. 2).

1.2 Main Contribution

We present the **N**o **IC** **E**ncryption (**NICE**)-PAKE protocol (Fig. 1), a round-optimal PQC PAKE from lattice assumptions without an IC. We realize public key authentication in the form of a bit-wise XOR, instead of symmetric encryption of the public key. Inspired by [ABJS24], our construction incorporates LWE KEMs with splittable public keys of the form $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$, where uniform samples $\mathbf{A}$ are deterministically obtained from a fixed-length bit string seed. As the seed is 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 fixed length random bit strings without resorting to block ciphers. As a result, we represent a major step in understanding how secure PAKEs can be built from abstract properties of lattice-based KEMs.

Alice		**Bob**
Password π		*Password* π'
$sk_a, pk_a \leftarrow \mathsf{KGen}(1^\kappa)$		
$(\rho, \mathbf{b}) \leftarrow \mathsf{Split}(pk_a)$		
$z \leftarrow \rho \oplus \mathcal{H}(\pi)$		
	$\xrightarrow{z, \mathbf{b}}$	
		$\rho' \leftarrow z \oplus \mathcal{H}(\pi')$
		$pk'_a \leftarrow \mathsf{Split}^{-1}(\rho', \mathbf{b})$
		$(C_b, K') \leftarrow \mathsf{Encap}(pk'_a)$
		if !$\mathsf{validate}(pk'_a, C_b, K')$ **then** abort
	$\xleftarrow{C_b}$	
$K \leftarrow \mathsf{Decap}(sk_a, C_b)$		
if !$\mathsf{validate}(pk_a, C_b, K)$ **then** abort		

Figure 1: The NICE-PAKE protocol utilizing the triple (KGen, Encap and Decap) of a KEM with splittable keys using a random seed ρ for the uniform sampling of the $\mathbf{A}$-part of the public key, which can be obtained from the invertible function Split.

1.2.1 Intuitive Security

Without knowledge of the correct password π, it is not possible to retrieve the seed ρ required for re-composing the initiator's public key $pk_a = (\mathbf{A}, \mathbf{b}) : \mathbf{A} \leftarrow \rho$ to invoke encapsulation, since it internally requires both parts of the public key. Valid ciphertexts generated from a key a with a non-matching $\mathbf{A}$-part cannot be decapsulated correctly. Thus, only with a password known by both parties is it possible to successfully execute the KEM to obtain the same KEM output key. Although the protocol only provides implicit mutual authentication, explicit authentication can be realized through adding a key confirmation round as a final step.

1.2.2 Novel KEM Properties from (M)LWE Assumptions

We introduce new properties from LWE (RLWE and MLWE) assumptions (Sec. 4) and call them $\mathbf{A}$-Part Secrecy (A-SEC-CCA), Splittable Collision Freeness (A-CFR-CCA), and Public Key Uniformity (SPLIT-PKU) for KEMs with splittable public keys. Our proof for the newly introduced A-SEC-CCA property is based on additional LWE assumptions (cf. Sec. 3.3). These are the *Non-Uniform* and *Weak-Hint* LWE (NLWE and whLWE) assumptions, which allow for more flexible utilization of LWE schemes and variants thereof.

Security from Abstract KEM Properties. A-SEC-CCA and A-CFR-CCA (cf. Sec. 4) play a major role in our analysis by providing a comparable alternative to KEM properties used in recent PAKE proposals (cf. Sec. 1.4). However, ours are tailored for the indistinguishability and collision freeness w.r.t. the $\mathbf{A}$-part embedded in a ciphertext, as opposed to the known anonymity and collision resistance. Especially, we show security even for an adversary that actively controls $\mathbf{b}$ or generates it maliciously, as $\mathbf{A}$ can still induce enough randomness to provide ciphertexts that do not decapsulate under non-matching $\mathbf{A}$-parts. Note that honest password usage is protected (information-theoretically) through the one-time XOR. Thus, an adversary cannot prove knowledge, nor gain partial information, about an honest choice of the $\mathbf{A}$-part for non-matching passwords. This limits the number of online password guesses to one per session and thwarts offline dictionary attacks, except for a few cases, where $\mathbf{b}$ may enable trivial decapsualtion (cf. Sec. 1.3).

NLWE and whLWE. Informally, NLWE enables sampling lattice points from a non-uniform distribution, and is used to switch the $\mathbf{A}$-part in an LWE sample with negligible adversarial advantage (cf. Thm. 4). whLWE determines the adversarial advantage from crafting malicious LWE samples (i.e., public keys), as it defines the leakage obtained from having hints on the encapsulator's secret in a ciphertext in the form of linear combinations, embedded by the adversary in $\mathbf{b}$ (cf. Thm. 5). We show that A-SEC-CCA suffices for reducing PAKE security from anonymity provided that certain malicious public keys can be detected by the encapsulator. Similarly, A-CFR-CCA requires that certain malicious ciphertexts can be detected by the decapsulator. For concrete instantiations, we claim that this leads to a significant loss in bit security for schemes built from plain (unstructured lattices) LWE. Nevertheless, we present a way to regain a sufficient amount of the lost bits using a non-standard value for the Gaussian variance parameter (Sec. 6). 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. 6).

Comparison to Other Hint-MLWE-based Protocols. The lattice hMLWE assumption has received significant attention in recent years, and is thus reaching a more stable state with more confidence in their security. For example, hMLWE and Hint-RLWE were shown to reduce to standard MLWE and RLWE assumptions in the context of a Proof of

Knowledge protocol [KLSS23]. Dodis et al. [DJK+25] realized hybrid key exchange in the Signal protocol from *Katana*, a Kyber variant that they proved IND-CPA secure under the hardness assumptions of hMLWE. The authors provide concrete parameters for a secure instantiation, which are roughly in-line with our approximation for the Gaussian distribution (cf. Sec. 6). Although we do not verify the correctness of our approximation, we assume similar heuristics to [DJK+25, Sec. 6.4] could apply to a fine-tuned NICE-PAKE. Hence, we suggest that with a secret of higher entropy for the encapsulator's randomness, A-SEC-CCA is suitable to MLWE schemes with an acceptable loss in bit security resulting from reducing the dimension of the module lattice (cf. Sec. 6). Niot [Nio25] also showed how to improve the performance of the Split-KEM[1] *K-Waay*, the PQ variant of the X3DH (Triple DH) key exchange in the Signal protocol, using a new Split-KEM called *Sparrow-KEM*. Based on hMLWE, Noit showed how to provide deniability for *Sparrow-KEM* and presented concrete parameters for hMLWE based on similar heuristics in [Nio25, Sec. 5.2]. Duverger et al. [DFJ+25] presented the first Subversion-Resilient PQC key exchange protocol called *SR-AKE* based on a re-randomizable KEM from MLWE, where the encapsulator provides an additional "sanitizing" secret in the ciphertext to prevent malicious low-norm malicious public key attacks (cf. Sec. 1.3). The correctness of the decapsualtion and the leakage in ciphertext randomness are also shown according to hMLWE with the concrete parameters presented in [DFJ+25, App. C]. Other important works that involve this assumption, or build secure schemes from it include: Plover lattice signatures [EEN+24], Threshold Raccoon signatures [DKM+24], Lattice-Based IND-CCA Threshold KEM [LP25].

1.3 Challenges in the Formal Analysis of NICE-PAKE

Although our construction may not be immediately instantiable in some real-world applications, we believe that we lay a theoretical foundation (including multiple new ideas) that will lead to future work. In Sec. 6 we provide potential future extensions.

1.3.1 Dealing with malicious public keys and ciphertexts.

Abstract KEM properties can only be applied for public keys and ciphertexts generated using the underlying KEM routines. Since adversaries are not bound to these routines, we need to deal with malicious values forwarded by a malicious Alice (i.e., **b**-part) or a malicious Bob (i.e., C_b). Here, we differentiate between two classes of adversarial messages: *malicious-valid* and *malicious-invalid*. The former are malicious yet valid KEM messages, where a mismatch in the **A**-part still yields indistinguishable decapsulations and produces different KEM output keys; thus, A-SEC-CCA and A-CFR-CCA can be applied. The latter values are malformed, so that an adversary can bypass the mismatch in the **A**-part, and either trivially decapsulate (for a malicious Alice), or obtain the same KEM output key (for a malicious Bob). Although most KEMs incorporate the FO transform, which provides a validation method for the decapsulator, we also want to consider the potential of other KEMs in our construction. To overcome this challenge, we introduce a validity check in the protocol to ensure that honest parties abort the execution when such *malicious-invalid* values are detected. We discuss the concrete functionality and instantiation possibilities of a ciphertext validity check (i.e., oracle) in Sec. 6.1. We also discuss other (currently not yet formalized) methods to mitigate such attacks in Sec. 6, which we will consider in future work.

[1] Split-KEM was first introduced by Brendel et al. [BFG+20] and is not to be confused with our Splittable KEM as the two terms are not related to each other.

1.3.2 Idealized Objects (ROM vs. QROM Programming).

The Random Oracle Model (ROM) is considered a standard for modeling hash functions, which is also the case for PAKEs. Consequently, the presence of quantum adversaries requires modeling hash functions w.r.t. to quantum queries in superposition (QROM). We claim that our construction, although relying on one RO-modeled hash function, does not need to consider the QROM in the proof. This claim can be justified through observing that: A classical attacker can already (pre-)compute all password hashes and a quantum attacker cannot do more than that, which is also supported by our inability to extract password guesses, as they are not made using an online-accessible RO or quantum RO for that matter.

ROM on the PAKE Level. The space of all possible input values to $\mathcal{H}$ is an offline accessible and pre-defined password dictionary $\mathcal{D}$, the values in which are publicly known prior to PAKE session initiation. That is, the hash function $\mathcal{H}$ is used to deterministically map a password π to a specific fixed-length output that matches the length of a fixed-length bit string (i.e., the seed ρ). It is thus obvious that an adversary can launch an offline dictionary attack (even without a quantum computer), and prepare a list of all hash outputs for all possible $\pi \in \mathcal{D}$. Therefore, they do not need to query $\mathcal{H}$ in superposition, as they can already obtain all possible outputs $\mathcal{H}(\pi_i)$. Ultimately, they can choose only one output $\mathcal{H}(\pi)$ to make a *committing* online password guess, and cannot change it adaptively in further interactions. Therefore, our PAKE proof does not need to deal with any quantum adversaries.

For the curious reader, consider a malicious Alice, who initiates a session by sending $(z, \mathbf{b})$ to Bob. Alice, who may very well be a quantum adversary, can only send one message to Bob and then wait for a response C_b. Having sent the first message, Alice had already committed to a password guess that is embedded in z, which she had made based on $\mathcal{D}$ and the prepared list of outputs $\mathcal{H}(\pi_i)$. Upon receiving C_b, there is no further interaction with Bob, and thus Alice may or may not leverage a quantum computer for further offline actions, which however do not affect the session anymore. Now we consider a malicious Bob, who receives $(z, \mathbf{b})$ from Alice. Obviously, Bob needs to pick one π and hope he guessed correctly to sample a matching $\mathbf{A}$ and then encapsulate a key into C_b. However, after that, they can no longer adaptively change their guess.

ROM in Novel KEM Properties. We also utilize ROM in the proofs of the new properties introduced in Sec+4. Nevertheless, ROM is only used in modeling a function mapping arbitrary-length input to arbitrary-length output. That is, to expand the fixed-length public key seed ρ to a lattice basis (or matrix) $\mathbf{A}$. Hence, neither adaptive RO programming nor adaptive RO queries are involved. In other words, even if the adversary has access to a quantum computer, their access to the RO is still treated in the "classical" manner [YZ21]. In this work, Yamakawa and Zhandry provide a thorough analysis of ROM vs QROM, where one can observe that having quantum capabilities does not necessarily translate to quantum access. Quote [YZ21] at p.11 Sec. 2 Para. 3 "We stress that the classical ROM can be considered even when we consider security against quantum adversaries". This is evident in our case, e.g., in Lemma 2 of the proof of Thm. 4. Here, one can observe that the lattice base $\mathbf{A}_0$ is the output of a RO, which is typically instantiated as an XOF (e.g., SHAKE) in FrodoKEM or Kyber. The catch is that this XOF takes input form a password-derived value, and hence an adversary who may want to query the RO, has no additional advantage in leveraging superposition queries over the password dictionary, because they can ultimately only interact once with the challenger when they commit to $\mathbf{b}$. Upon receiving the ciphertext C, the adversary may or may not iterate over all RO outputs they could compute, or had already computed from the password dictionary. An identical line of argument can be found in the proof of UNI-PK by Arriaga et al. [ABJS24,

App. C], where the authors show that ML-KEM has splittable and pseudo-uniform public keys in the ROM (analogous to our SPLIT-PKU assumption in Sec. 4), without addressing the QROM , which we utilize in our proof of Thm. 3. That is, the RO modeled hash function is used as a programmable mapping function, and not as a quantum-accessible RO for superposition queries, or as an extractable compressed RO.

1.3.3 (Bounding) Password Extraction

Our proof is provided in the extended RoR (*Real-or-Random*) BPR (Bellare-Pointcheval-Rogaway) model [BPR00] (Sec. 5). Conducting the formal analysis, we concluded that it is infeasible to extract password guesses from an adversary's interaction with the protocol without an IC (cf. Sec. 1.4) or a similarly programmable construction (cf. POPF in[Sec. 3] [MRR20] and in [ABJ25, Sec. 4.1]). 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 in the ROM to extract password guesses in this case. Considering the flow of a protocol execution, one may observe the following.

1. Extraction from a RO-modeled function $\mathcal{H}$ is not possible, as the implementing oracle only takes the password as input, and not a public key. Thus, an adversary can query $\mathcal{H}$ on multiple passwords π_i and later decide which one to use to mask their public key $\mathbf{A}$-part seed ρ.
2. Extraction from an adversary's ciphertext C_b is not possible, as the underlying KEM provides strong anonymity (ANO-CCA and A-SEC-CCA) and hence protects the public key and the $\mathbf{A}$-part used in the encapsulation.

Since we discussed the first aspect in all detail in the previous paragraph on the ROM, we focus now on the second one and ignore any IC-related aspects, as they are sufficiently discussed throughout the paper.

Our construction looks very similar to the well known and well studied SPEKE [Jab97] and SPAKE2 [AP05] protocols. However, we cannot leverage computational hardness assumptions in a white-box manner to obtain new assumptions such as the *Password-Based Chosen-Basis Computational Diffie-Hellman* defined in [AP05], or Gap Computational Diffe-Hellman (Gap CDH) used in [ABB+20]. In DH-based PAKEs, there is no ciphertext in the second protocol message. Instead, there is usually Bob's DH key, and sometimes a confirmation tag (cf. OEKE [JRX25]). Informally, DH-based PAKEs allow simulators to compute matching session keys even in the presence of an active adversary, because the simulator receives the adversary's DH key, which they can combine with an honestly simulated one. Depending on the outcome, the simulator can (sometimes) later extract the password guess, if they had not already done so, which is thoroughly discussed in the relaxed variant of the UC PAKE functionality [ABB+20] (see the next paragraph for BPR vs. UC). We construct our PAKE using a black-box KEM, in which the KEM output key is already established on Bob's side through encapsulation, and Alice is not supposed to learn anything about how the ciphertext was computed. Essentially, this is what anonymity means in this context, and also why NIST KEMs use the re-encryption step in the FO. Still, re-encryption does not tell us which public key was used in the encapsulation. It only tells us, if the decapsualtion is correct s.t. the re-encrypted ciphertext matches the one we received (cf [HHKR25] for more details on FO KEMs in PAKEs). In fact, without anonymity, a malicious Alice would always be able to conduct a successful offline dictionary attack on any KEM-based PAKE, which closes the circle compared to DH, where one argues about the adversary's chances at breaking DH to determine the correct password.

That being the case, we conduct our proof using another strategy, where our BPR challenger cannot extract password guesses. Instead, we let our challenger choose randomized messages

from the same distribution of all possible guesses, which is illustrated in Fig. 7 and Fig. 8. By doing so, the BPR simulation basically runs the security games independent of honest passwords at the end, but the security loss is bounded over a smaller distribution than in cases where one can program according to extracted password guesses. One can imagine that the challenger makes their own password guess when they randomize messages in the presence of active adversaries, and consequently, the probability of both the adversary and the challenger guessing the same password, is almost the same as the adversary choosing the correct one. Although this leads to loose results in the overall security bound, we consider this trial an important step towards realizing a such PAKE without the IC.

1.3.4 BPR vs UC Security

In order to achieve UC security with composability guarantees, a simulation-based proof is required, where the simulator needs to be able to extract adversarial password guesses. That is, the extraction must be "online and straight-line" [ABB+20]. Since we do not extract guesses from adversarys' messages, we cannot provide a sound simulation in UC. A method to overcome this obstacle can be found in adding an extra round at the end of the protocol, where honest parties (and adversaries) have to query a RO-modeled function to generate a key confirmation using their passwords. While this can be achieved via lazy-extraction in the relaxed UC model [ABB+20, HTTY24] in the ROM, involving an interactively queryable ROM means that it can also be adaptively queried by a quantum adversary. This requires adaptive programming in QROM to realize a sound UC simulation [Hes20, MX23b, LL25]. Therefore, we deem such improvements a work in progress. Finally, we note that the majority of the PQC PAKE proposals in the literature are given in the BPR model [AHMW25].

1.4 Comparison to Recent Generic PAKE Constructions

Directly comparable Related work follows one design paradigm [AHMW25], namely that of (O)EKE [AHMW25], which enables generic usage of key agreement schemes in black-box manner. The high-level idea is to encrypt the public key and the ciphertext using the password in the EKE variant. Alternatively, in OEKE, the ciphertext is authenticated with a key confirmation tag only. We present an overview of the abstract security properties required in each PAKE construction in Tab. 1. These works can be further categorized into three groups [AHMW25]: 1. *EKE with IC* [BCP+23, PZ23, AHHR24, HHKR25]. 2. *EKE with half IC (HIC)* [SGJ23, ABJS24]. 3. *EKE without IC* [MRR20, JRX25, ABJ25].

Table 1: Required KEM Properties in Generic PQC PAKEs - Adapted from [AHMW25]

Protocol	Session Key	Public Key	Ciphertext	Model(s)
POPF-EKE [MRR20]	IND-CPA	PR	PR/ROB	ROM
(O)CAKE [BCP+23]	IND-CPA	Fuzziness	ANO-CPA	IC/ROM
HIC-EKE [SGJ23]	IND-CPA	UNI-PK	ANO-CCA	IC/ROM
Pan and Zeng (CAKE) [PZ23]	IND-CPA	Fuzziness	ANO-PCA	IC-ROM
Alnahawi et al. (OCAKE) [AHHR24]	IND-CPA	PKU	ANO-PCA	IC-ROM
Januzelli et al. [JRX25] (EKE)	IND-CCA*	PR	(S)PR-NM	ROM
Januzelli et al. [JRX25] (OEKE)	IND-CCA*	PR	PR-NM/CFR	ROM
CHIC [ABJS24]	OW-CPA	UNI-PK	ANO-PCA	IC/ROM
NoIC-PAKE [ABJ25]	OW-CPA	UNI-PK	ANO-PCA	ROM
Hövelmanns et al. [HHKR25] (CAKE)	IND-CPA	PKU	ANO-PCA	(Q)IC/QROM
Hövelmanns et al. [HHKR25] (OCAKE)	IND-CPA	PKU	ANO-PCA	(Q)IC/QROM
This work (NICE-PAKE)	IND-CCA	SPLIT-PKU	A-SEC/A-CFR	ROM

* Since [JRX25] requires (S)PR-NM, it is implied that the KEM also needs to provide IND-CCA.

1.4.1 IC-EKE (CAKE)

Beguinet et al. [BCP+23] presented one of the first generic PQC PAKEs ((O)CAKE) in UC. The Follow-up works by Pan and Zeng [PZ23] and Alnahawi et al. [AHHR24] offered a further detailed analysis for multi-user security in the BPR model. A common divisor among these works is that they all require KEMs satisfying public key uniformity (*Fuzziness* or PKU) and anonymity. Our proof requires roughly the same basic KEM properties in the proof steps concerning honest sessions.

QROM Extraction and QIC. Hövelmanns et al. [HHKR25] showed how to lift the classical OCAKE proof given in [AHHR24] to quantum settings by incorporating QROM for search and collision bounds, as well as online extraction of adversarial queries (including password guesses). The authors utilize novel QROM techniques based on the compressed oracle function (CO) and its extractable variant (eCO). However, the proof still relies on a quantum IC equivalent, for which is compensated via a simulator SIM that may be used should the QIC be realized in the future. While our construction does not require extraction in the QROM, a final key confirmation round, will indeed add a public oracle queryable by adversaries in superposition. As discussed previously, this step is currently a work in progress for NICE-PAKE. Nonetheless, since we eliminate the need for an IC, no direct comparison to SIM can be established in that sense.

Programming Anonymity Reductions. An interesting finding in [HHKR25] shows the impossibility of a black-box reduction from anonymity without a programmable IC. This implies the need for programming the challenge public keys in the proof using techniques similar to the ones provided by an IC. Otherwise, an adversary may be able to break anonymity through an offline dictionary attack upon receiving a ciphertext response to determine the most likely encapsulation public key. Although our construction relies on anonymity, we require the non-programmable variant only for passively observed sessions, where the public key is generated honestly. For an adversary controlling key generation, we reduce to A-SEC-CCA, where the adversary is challenged to guess the used $\mathbf{A}$-part. Here, an adversary does not need a decryption oracle, as they have the secret key to their chosen $\mathbf{b}$. The catch here is that they cannot determine an $\mathbf{A}$-part embedded in a reduction to A-SEC-CCA, even if they were able to successfully decrypt the challenge ciphertext (discussed previously). Therefore, our reduction from A-SEC-CCA only needs to embed a random $\mathbf{A}$ and an expanded one from a possible seed obtained by unmasking based on the password dictionary $\mathcal{D}$ in its challenge (cf. Sec. 4). Hence, their success probability will remain upper bounded by random guessing over $\mathcal{D}$; plus the advantage they have against the hardness of whLWE w.r.t. parameters chosen for a concrete instantiation.

1.4.2 Splitting a KEM Public Key (HIC-EKE)

Dos Santos et al. [SGJ23] presented a UC-secure PAKE under a relaxation of the IC called Half IC. Their EKE-KEM protocol utilizes a KEM and a modified 2-round Feistel construction (m2F). The m2F avoids using an IC over a group through defining an IC over a fixed-length bit string domain. Subsequently, Arriaga et al. [ABJS24] introduced the Compact HIC (CHIC) protocol, improving EKE-KEM [SGJ23] 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 [SGJ23]. The authors in [ABJS24] 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. Our construction extends

the concept of splittable KEMs to completely avoid the IC, and thus also extends the notion of UNI-PK to SPLIT-PKU via a straightforward reduction from (M)LWE.

1.4.3 POPF-EKE (NoIC-EKE)

McQuoid et al. [MRR20], Januzelli et al. [JRX25], and Arriaga et al. [ABJ25] rely on Programmable-Once Public Functions (POPF) to realize EKE without an IC. These PAKEs may be instantiated from any PKE or KEM scheme that fulfill certain properties as shown in Tab. 1. Basically, the POPF construction is a 2-round Feistel network (2f), which realizes a keyed randomized function [ABJ25]. The PAKE initiator's public key is encrypted with a password-derived key by hashing the public key into an algebraic group (H2G). The 2f can also be queried in the reverse direction enabling programming in UC. Whereas Arriaga et al. define the same KEM property requirements as in CHIC, McQuoid et al. and Januzelli et al. use different naming conventions for similar or identical properties and define further (stricter) requirements for (O)EKE (see Tab. 1).

1.4.4 Security Properties for (O)EKE and NICE-PAKE

Interestingly, and previously unaware of the following fact, our construction relies on similar properties as in [JRX25] to realize a secure KEM-based OEKE without a final key derivation function. NICE-PAKE requires a KEM that satisfies IND-CCA, SPLIT-PKU, ANO-CCA, A-SEC-CCA, and A-CFR-CCA. Considering that A-SEC-CCA and A-CFR-CCA serve as complementary properties for ANO-CCA and SCFR-CCA, NICE-PAKE is in-line with the definitions of [JRX25]. According to the authors, a secure (O)EKE requires KEMs with a pseudo-random non-malleable (PR-NM) and collision-resistant second message (i.e., the ciphertext). Here, PR-NM is similar to strong pseudo-randomness as defined in [MX23a], which can be achieved if the KEM simultaneously provides IND-CCA and ANO-CCA. In addition, collision-resistance corresponds to SCFR-CCA, which we provide through A-CFR-CCA. However, we leave a proof of equivalence for future work.

1.4.5 Comparing Instantiations

Vos et al. instantiated NoIC-PAKE in the OEKE variant using ML-KEM [VJW+25]. The resulting protocol, OQUAKE, first obfuscates the public key (as in [GRSV25]) before sending it through the 2F, where it is then XORed with a hash of the password and some fixed length randomness. This procedure is repeated in the reverse direction on the receiver side as well. Considering the performance overhead resulting from obfuscation and applying the 2F, NICE-PAKE should outperform in terms of efficiency. However, aspects related to modified scheme parameters and the validation steps in our construction indicate that the performance gains are not quite as good as we had hoped for.

1.5 Other Related Work

There already exist many PAKEs from PQC primitives in the literature, almost all of which are based on lattice LWE (e.g. [KV09, DAL+17, Tan21, Wan22, GSG+23, CKS23]) and isogenies (e.g. [ZHS14, TY19, AEK+22]). However, apart from the previously discussed OCAKE proof [HHKR25], the only PQC PAKE with a complete security analysis in QROM was given in 2024 by Lyu et al. [LLH24]. The authors proposed the first PAKE protocol from 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) transform to elevate the security of the chosen LPKE

from IND-CPA to IND-CCA. Nevertheless, a comparison with the aforementioned works is infeasible, since they utilize asymmetric PQC primitives directly and build upon different PAKE design paradigms [AHMW25], i.e., they do not utilize PQC KEMs generically.

2 Shortcomings of the IC Model in PAKE Proofs

The ideal cipher model was formalized in PAKE settings in the BPR model [BPR00]. It has been widely used in protocols instantiated with block ciphers (cf. Sec. 1.4). Thus, an IC serves in modeling block ciphers (e.g., AES) as idealized objects similar to hash functions in the ROM with some exceptions [Bla06]. Its advantage is in 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 1 (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 $\{0,1\}^{n}$.

Similarly 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 the 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.

Quantum Ideal Cipher. Some works in the literature [AR17, HY18, SS19, CMSZ19, Unr21, Ros21, ABPS24, Unr23, MMW25] address the notion of the *Quantum* IC model focusing on one-way functions and non-invertible permutations. However, it has not yet been shown how to obtain the lazy sampling technique required for programming in PAKE proofs. The work by Unruh [Unr23] models keyed invertible permutations similar to compressed function oracles (CFO) [Zha19] by introducing compressed permutation oracles (CPO). The work by Majenz et al. [MMW25] introduces permutation superposition oracles as a generalization of CFOs and shows the possibility of their usage in quantum queries on permutations and their inverses. Nevertheless, relying on a quantum equivalent of the IC model is not yet an option [HHKR25].

IC Issues in Real-World Instantiations. Unsafe IC instantiations often result in a mismatch between the domain of a standard block cipher and the domain of encrypted messages. Usually, the IC is instantiated using block ciphers over 128 bit domains (e.g., AES), which is obviously impossible for much larger domains such as in PQC KEM public keys (and ciphertexts). It was shown that an 8-round Feistel is indifferentiable from an IC [DS16]. However, an 8-round Feistel requires four H2G queries for each cipher operation, which is clearly less efficient.

IC Issues in PAKE Proofs. Narrowing down the issue at hand, and ignoring the QIC for a moment, using the 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 encryption or decryption. Furthermore, 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) 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 (malformed) keys may very well be used by adversaries to break the underlying scheme.

Practical Example. We devise a simple PAKE construction (Fig. 2), 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.

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

Figure 2: Simple NICE-PAKE

First, we consider a malicious *Bob* who receives a masked pk as z as in Fig. 2. *Bob* may guess a password π' and observe the result of unmasking z. With probability bounded by the size of the password dictionary, 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 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 $\mathbf{b}$-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 (malformed) 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 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 undetectable mismatch can be achieved.

3 Preliminaries

3.1 Notation

The security parameter is denoted by $\kappa \in \mathbb{N}$, and 1^κ is its unary representation. $a \leftarrow \mathsf{Alg}(b)$ denotes running (potentially randomized) algorithm Alg on input b (with fresh random coins). We denote value assignment 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, 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

$$\mathsf{Adv}^{\mathsf{PROP.BIT}}_{\mathcal{A},\Pi}(1^\kappa) = \big|\Pr\big[\mathsf{Exp}^{\mathsf{PROP.BIT}^0}_{\mathcal{A},\Pi}(1^\kappa) = 1\big] - \Pr\big[\mathsf{Exp}^{\mathsf{PROP.BIT}^1}_{\mathcal{A},\Pi}(1^\kappa) = 1\big]\big|$$

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

$$\mathsf{Adv}^{\mathsf{PROP.FIND}}_{\mathcal{A},\Pi}(1^\kappa) = \Pr\big[\mathsf{Exp}^{\mathsf{PROP.FIND}}_{\mathcal{A},\Pi}(1^\kappa) = 1\big]$$

3.2 The Random Oracle Model

A random oracle [BR93] (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 2 (Random Oracle)**.** *RO* is a function f that maps elements over the function space $\{0,1\}^* \to \{0,1\}^{poly(\lambda)}$, where:

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

3.3 Lattice Assumptions: NLWE and whLWE

To obtain the new KEM security properties mentioned in Sec. 1.2.2 and formally defined in Sec. 4.2, we make use of the Non-Uniform LWE and Weak-Hint LWE assumptions formalized 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 correspond to possibly maliciously generated LWE samples in our context. We refer the reader to App. B for more details on LWE and its variants, App. C.1 for Extended LWE (eLWE) and to App. C.2 and C.3 for more details on NMLWE and eMLWE respectively.

3.3.1 Non-uniform Learning with Errors (NLWE).

Boneh et al. [BLMR13] 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:

$$(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) \text{ and } (\mathbf{A}, \mathbf{u})$$

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. [BLMR13] 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 [BLMR13])**.** 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 ([BLMR13])**.** *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 [BLMR13, 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.

3.3.2 Weak-Hint Learning with Errors (whLWE)

Cheon et al. [CKK+21] (resp. Lee et al. [LKK+18]) defined Hint LWE (hLWE) and Weak-Hint LWE (whLWE) – with Liu et al. [LSTW25] 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 ([LKK+18], 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 [CKK+21, LKK+18].

Definition 5 (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 $\mathbf{a} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow D^k_{\sigma_q}$, 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. [CKK+21, LKK+18] are able to show the following security theorem, which we will leverage in what follows.

Theorem 2. *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.*

3.4 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 be viewed as a special type of public key encryption (PKE) algorithm [CDM24], where the PKE is used to encrypt and decrypt a payload message, which is then used to derive said shared key. 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.

3.4.1 KEM Syntax

In line with Cremers et al. [CDM24], we adopt a modified KEM API definition that allows us to explicitly use a public key as input to the decapuslation routine. Thus, we can easily state whether the same public key was used in both encapsulation and decapsulation.

Definition 6 (Key Encapsulation Mechanism (KEM))**.** KEM is a triple $(\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$:

- $\mathsf{KGen}(1^\kappa) \to (pk, sk)$: On input security parameter κ **return** key pair (pk, sk) (probabilistic algorithm), s.t. $pk \in \mathcal{PK}$, the public key space for KEM.
- $\mathsf{Encap}(pk) \to (C, K)$: On input pk **return** ciphertext $C \in \mathcal{C}$ and key $K \in \mathcal{K}$ (probabilistic algorithm).
- $\mathsf{Decap}(sk, pk, c) \to K$: 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.

3.4.2 Correctness of KEM

Except for a small probability, the decapsulation algorithm $\mathsf{KEM.Decap}$ recovers the same shared key K produced by the encapsulation algorithm $\mathsf{KEM.Encap}$ for a public key generated by the key generation algorithm $\mathsf{KEM.KGen}$.

Definition 7 (KEM Correctness)**.** For every key pair $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ and each respective 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 $\delta = 0$.

3.4.3 FO-KEM with Implicit Rejection

The Fujisaki–Okamoto (FO) transform introduced in [FO99, FO13] is a common construction used to elevate the security of an IND-CPA PKE to an IND-CCA KEM [CDM24]. As opposed to explicitly rejecting KEMs, most NIST PQC KEMs use variants of FO with implicit rejection ($\mathsf{FO}_m^{\not\perp}$), 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 $\mathsf{FO}_m^{\not\perp}$ KEM as depicted in Fig. 3. This type of KEM can be viewed as a generalization for most LWE (and variants) KEMs: it is used in CRYSTALS-Kyber, FrodoKEM, NewHope and Korea PQC competition winner SMAUG-T[2].

$\mathsf{KEM.KGen}(1^\kappa)$	$\mathsf{KEM.Encap}(pk)$	$\mathsf{KEM.Decap}(sk', pk', C)$
$(pk, sk) \leftarrow \mathsf{PKE.KGen}(1^\kappa)$	$m \leftarrow\!\$\ \mathcal{M}$	parse sk' as $(s, sk, pk, G_1(pk))$
$s \leftarrow\!\$\ \{0,1\}^{\lvert s\rvert}$	$(r, k) \leftarrow G_2(G_1(pk), m)$	$m' \leftarrow \mathsf{PKE.Dec}(sk, C)$
$sk' \leftarrow (s, sk, pk, G_1(pk))$	$C \leftarrow \mathsf{PKE.Enc}(pk, m; r)$	$(r', k') \leftarrow G_2(G_1(pk), m')$
return (pk, sk')	$K \leftarrow F(k, C)$	$C' \leftarrow \mathsf{PKE.Enc}(pk, m'; r')$
	return (C, K)	**if** $C' = C$ **then**
		return $K' \leftarrow F(k', C)$
		else return $K' \leftarrow F(s, C)$

Figure 3: $\mathsf{FO}_m^{\not\perp}$ KEM with pk as explicit input to Decap using a PKE with message space $\mathcal{M}$ and hash functions G_1, G_2 and F. The output of G_2 depends on pk and m and can be viewed as a concatenation of r and k as in e.g., Kyber [BDK+18] using SHA-512 for G_2 - Adapted from [BCD+16, CDM24, GMP22].

3.4.4 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 [CDM24, GMP22, JRX25]. Furthermore, we adopt the notion of public key uniformity for a KEM, which was introduced in [BCP+23] and [PZ23] as the *Fuzziness* property, and later formalized in [AHHR24]. We deem this property vital in the context of PAKE dictionary attacks, as it prevents adversaries from distinguishing honestly generated public 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. 4).

Definition 8 (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. 4.

Definition 9 (IND-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 encapsulated 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. 4.

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

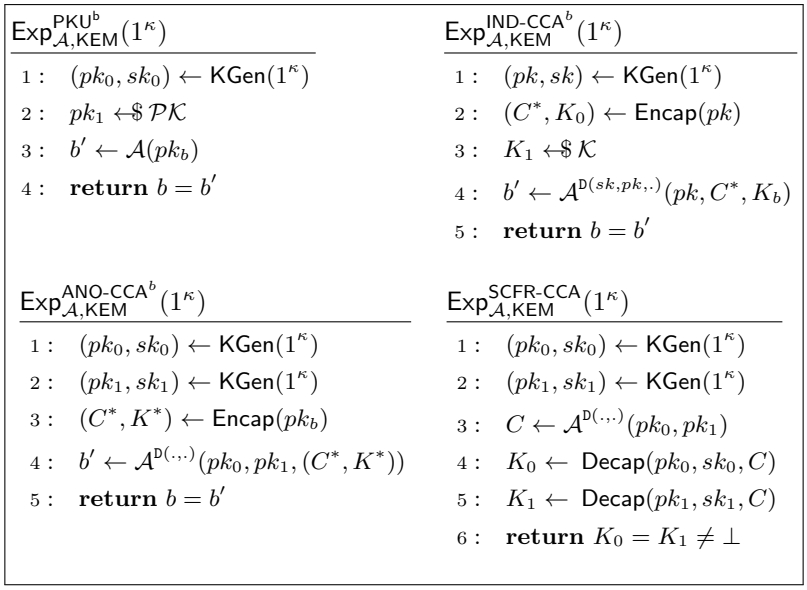


Figure 4: KEM Experiments $\mathsf{Exp}^{\mathsf{PKU}}_{\mathcal{A},\mathsf{KEM}}$, $\mathsf{Exp}^{\mathsf{IND\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$, $\mathsf{Exp}^{\mathsf{ANO\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$ and $\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}$. The experiments IND-CCA, ANO-CCA and SCFR-CCA are adopted from [GMP22]. The decaspulation 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). The Decap routine in $\mathsf{Exp}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{KEM}}$ explicitly takes an input parameter pk_i as in [CDM24]. CPA versions of the aforementioned properties can be obtained by omitting the decryption oracles from the experiments.

Definition 10 (ANO-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. 4.

Definition 11 (SCFR-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 decapsulates 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. 4.

We note that KEMs with implicit rejection satisfying the collision freeness property (SCFR-CCA) cannot satisfy the strong robustness property [GMP22, CDM24] (SROB-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 guaranties 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.

3.5 Lattice KEMs with Splittable Public Keys

KEMs based on the lattice LWE problem and its variants have public keys of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$, where a matrix $\mathbf{A}$ defines the base of the lattice expanded 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. 6, which can be obtained from an LWE-based PKE with splittable public keys (Fig. 5).

$\mathsf{PKE.Enc}(pk, m; r)$

$(\mathbf{A}, \mathbf{b}) \leftarrow pk$ // 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 $\mathbf{b} = \mathbf{As} + \mathbf{e}$ (simplified)

return $C := (\mathbf{u}, \mathbf{v})$

Figure 5: Simplified exposition of encryption using an internal PKE in a lattice LWE KEM with splittable public keys of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$. Inputs to PKE.Enc are passed down from KEM as in Fig. 3.

Definition 12 (The Split $(\mathbf{A}, \mathbf{b}) \leftarrow \mathsf{Split}(pk)$)**.** Let KEM be a key encapsulation mechanism with public keys of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$ defined over public key space $\mathcal{PK}$, where pk is a simple concatenation of two bit strings ρ and $\mathbf{b}$ (in any given order). Then: $\mathsf{Split}(pk)$ is the function that trivially decomposes pk into the two said components $\mathbf{b}$ and ρ, and Split^{-1} is its reverse operation. In the following, we simplify the notation to $(\mathbf{A}, \mathbf{b}) \leftarrow (pk)$ for ease of exposition.

Recall that in such KEMs, a PKE is used for encryption and decryption (Fig. 3), 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. 5). When decapsulating a ciphertext C to obtain K, the decapsulator also learns the message m 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 [CDM24]. Further, the encryption yields a ciphertext of the form $(\mathbf{u}, \mathbf{v})$ (Fig. 5), 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 encapsulator; otherwise, they cannot decrypt the message m. Adapting the definition in [ABJS24] we specify the subset of KEMs that are in scope in the rest of the document. It follows[3]:

Definition 13 (LWE KEM with Splittable Public Keys (LS-KEM))**.** Let KEM be a key encapsulation mechanism (Def. 6) 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 ($\mathrm{FO}^{\not\perp}_m$) transform as defined in Fig. 3,
- it utilizes an underlying PKE following the syntax defined in Fig. 5 (and therefore has public keys of the form $pk := (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$), and
- $\mathbf{A}$ is computed (expanded) from $\rho \in \{0,1\}^l$ via a deterministic function G.

4 Novel Security Properties for Splittable KEMs

We introduce the security properties for KEMs with splittable public keys (LS-KEM) of the form $pk : (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$, where $\mathbf{A}$ is deterministically computed (expanded) from a fixed-length random bit string seed ρ, and is thus chosen from the $\mathbf{A}$-part space $\mathbb{A}$, the set of uniform distributed lattice matrices of the same size as the matrices probabilistically generated using KEM.KGen.

[3]See [ABJS24] for a more formal definition, here we simplify for ease of exposition.

4.1 Key Uniformity of KEM with Splittable Public Keys

Based on the general KEM public key uniformity (cf. Sec. 3.4 Def. 8), we derive a similar notion for our previously defined LS-KEM, the security experiment of which is shown in Fig. 6. We claim with this extension of the original notion, and the UNI-PK notion of Arriaga et al. [ABJS24], 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 UNI-PK for MLWE (e.g., Kyber) follows from [ABJS24], and can be similarly shown for LWE (e.g., FrodoKEM) via a trivial reduction from the decisional LWE assumptions.

Definition 14 (Key Uniformity of KEM with Splittable Public Keys)**.** For 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 an 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.

$\mathrm{Exp}^{\mathsf{UNI\text{-}PK}^b}_{\mathsf{KEM},\mathsf{Split}}(\mathcal{A})$	$\mathsf{Exp}^{\mathsf{SPLIT\text{-}PKU}^b}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
1 : $(pk_0, -) \leftarrow \mathsf{KGen}(1^\kappa)$	1 : $(pk_0, sk_0) \leftarrow \mathsf{KGen}(1^\kappa)$
2 : $(r_0, M_0) \leftarrow \mathsf{Split}(pk_0)$	2 : $(\mathbf{A_0}, \mathbf{b}_0) \leftarrow pk_0$
3 : $(r_1, M_1) \leftarrow N_\kappa \times R_\kappa$	3 : $\mathbf{A}_1 \leftarrow\!\$\, \mathbb{A}$
4 : $b' \leftarrow \mathcal{A}(r_b, M_b)$	4 : $b' \leftarrow \mathcal{A}(\mathbf{A}_b, \mathbf{b}_0)$
5 : **return** $b == b'$	5 : **return** $b == b'$

Figure 6: Key Uniformity for LS-KEM. UNI-PK (left) is adopted from [ABJS24] in original notation where Split is an invertible map $\mathcal{PK}_\kappa \to N_\kappa \times R_\kappa$ s.t $r \in N_\kappa$ is a fixed-length seed and $M \in R_\kappa$ is a lattice sample, which corresponds to our experiment (right).

Theorem 3. *Let* LS-KEM *be the key encapsulation mechanism defined in Def. 13, where* $\mathbf{A}$ *is deterministically computed from the* $\mathbf{A}$*-part space* $\mathbb{A}$ *via a uniformly sampled fixed-length bit string seed* ρ *using the function* G *modeled as a RO, then the advantage of an adversary against* SPLIT-PKU *can be expressed as follows:*

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

Proof of Theorem 3. Recall that $\mathbf{A}$ is deterministically expanded 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 $\mathsf{Exp}^{\mathsf{UNI\text{-}PK}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$ 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 [ABJS24], 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. □

4.2 A-Part Secrecy for KEM with Splittable Public Keys

Based on the definitions in Sec. 3.3, 3.4, and 3.5, we rely on the fact that the **u**-part in C resembles a chosen sample that belongs to the same hardness assumption of the chosen KEM (i.e., LS-KEM), so that it is not possible to determine which **A** was used to create **u**. This assumption plays an essential role in our security analysis in Sec. 5. Formally, we claim that using a different **A**-part for encryption results in ciphertexts that cannot be decrypted correctly and cannot be used to reveal a non-related **A**. We formalize this claim as KEM **A**-Part Secrecy (A-SEC-CCA) depicted in Fig. 7, and prove its correctness.

Definition 15 (Splittable KEM **A**-Part Secrecy)**.** For LS-KEM with public key space $\mathcal{PK}$ and **A**-part space $\mathbb{A}$, we define the advantage of an adversary $\mathcal{A}$ in distinguishing a random **A**-part of a known public key used to probabilistically generate a ciphertext as $\mathsf{Adv}^{\text{A-SEC-CCA}}_{\mathcal{A},\text{KEM}}(1^\kappa)$ where $\mathsf{Exp}^{\text{A-SEC-CCA}}_{\mathcal{A},\text{KEM}}(1^\kappa)$ is the game depicted in Fig. 7.

$\mathsf{Exp}^{\text{A-SEC-CCA}^b}_{\mathcal{A},\text{KEM}}(1^\kappa)$	$\mathsf{Exp}^{\text{A-SEC-CCA}^{(b,\mathcal{D})}}_{\mathcal{A},\text{KEM}}(1^\kappa)$
$(pk, sk) \leftarrow \mathcal{A}$	$(pk, sk) \leftarrow \mathcal{A}$
$\mathbf{A}_0 \leftarrow\$ \mathbb{A}$	$\mathbf{A}_0 \leftarrow\$ \mathcal{D}$
$\mathbf{A}_1 \leftarrow\$ \mathbb{A}$	$\mathbf{A}_1 \leftarrow\$ \mathbb{A}$
$(C, K) \leftarrow \mathsf{Encap}'_{\mathbf{A}_b}(pk)$	$(C, K) \leftarrow \mathsf{Encap}'_{\mathbf{A}_b}(pk)$
$b' \leftarrow \mathcal{A}(C, pk, sk, \mathbf{A}_0, \mathbf{A}_1)$	$b' \leftarrow \mathcal{A}(C, pk, sk, \mathbf{A}_0, \mathbf{A}_1)$
if $!\mathsf{validate}(pk[\mathbf{A}_b], C, K)$	**if** $!\mathsf{validate}(pk[\mathbf{A}_b], C, K)$
$\quad b' \leftarrow\$ \{0,1\}$	$\quad b' \leftarrow\$ \{0,1\}$
return $b == b'$	**return** $b == b'$

Figure 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 $\mathsf{Encap}'_{\mathbf{A}}(pk)$ denotes invoking Encap explicitly with a matrix $\mathbf{A}_b$, i.e. ignoring the one embedded in pk (that was initially chosen by $\mathcal{A}$). $pk[\mathbf{A}_b]$ denotes performing the validation w.r.t. $pk = (\mathbf{A}_b, \mathbf{b})$. The ciphertext validity check is defined in Sec. 6.1 Def. 18.

Remark on property idea. Unlike the standard anonymity notion, we want to explicitly capture the case where an adversary controls the key generation. As we will discuss in detail in Sec. 6.1, our PAKE protocol (and its analysis) requires both parties to detect ciphertexts generated using malicious key material: this step is also necessary in this game to disallow trivial wins through re-encryption. We therefore incorporate the ciphertext validity check (cf. Def. 18) and abort if C can be trivially decapsulated (without the secret key sk), where the validity check is called using the **A**-part used in the experiment and the **b**-part provided by the adversary $\mathcal{A}$.

Remark on usage in PAKE formal analysis. Since an adversary is 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. Here, we follow the BPR definition of $\mathcal{D}$ assuming a uniform distribution of passwords within the dictionary. Thus, any seed obtained from unmasking z follows the same distribution, since the masked seed is an unstructured random bit string.

Theorem 4. *Let* LS-KEM *be the key encapsulation mechanism defined in Def. 13 where* **A** *is deterministically computed from the* **A***-part space* $\mathbb{A}$ *via a uniformly sampled fixed-length*

bit string seed ρ using the function G modeled as a RO, then the advantage of an adversary against A-SEC-CCA *can be expressed as follows:*

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

Remark on concrete A-SEC-CCA-security. In the following, we show that the **A**-Part Secrecy (A-SEC-CCA) property applies to LS-KEM as previously defined in Sec. 3.5 (Def. 13) that is instantiated with any FrodoKEM-style plain LWE PKE. We will show that the concrete security loss using FrodoKEM (cf. Sec. 6), 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 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 of Theorem 4. We show A-SEC-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. Note that if the ciphertext validity check fails, then $\mathsf{Adv}^{\text{A-SEC-CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa) = 0$ and therefore the following proof only considers the case where the check passes.

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 computed from distribution $\mathcal{D}$, which is a well-spread depending on the entropy of the password π with support in the **A**-space of the KEM, and $\mathbf{A}_1$ is chosen uniformly from the **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 the ciphertext C.

Game $\mathbf{G}_1$. In this hybrid, we swap $\mathbf{A}_0$ to be chosen uniformly from the **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 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 computed $\mathbf{A}_0$ as the output of a RO, which was given as input a randomly chosen 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

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

in a straightforward way. Following [BLMR13, 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 $v = \mathbf{br} + \mathbf{e}''$. 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 remains the same.

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 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. 6.2).

Completing the proof of Theorem 4. 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 4. □

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. C.2 and App. C.3), and if there exists a reduction from* MLWE *to* whMLWE*, which closely preserves the adversary's advantage against (*MLWE*), then it follows a similar proof for Theorem 4 for any* LS-KEM *instantiated from MLWE hardness assumptions (e.g., CRYSTALS-Kyber), the concrete parameters of which are discussed in Section 6.*

4.3 Collision Freeness for KEM with Splittable Public Keys

Analogously 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 16 (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.

Remark on property idea. Following Cremers et al. [CDM24], 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. Thus, the adversary's goal in the A-CFR-CCA game (Fig. 8) is to enforce a collision on ciphertexts so that they decapsulate to matching (i.e., identical) keys although encapsulated with different **A**-parts.

$\mathsf{Exp}^{\mathsf{A\text{-}CFR\text{-}CCA}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$	$\mathsf{Exp}^{\mathsf{A\text{-}CFR\text{-}CCA}^{\mathcal{D}}}_{\mathcal{A},\mathsf{KEM}}(1^\kappa)$
1 : $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$	1 : $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$
2 : $(\mathbf{A}, \mathbf{b}) \leftarrow pk$	2 : $(\mathbf{A}, \mathbf{b}) \leftarrow pk$
3 : $\mathbf{A}^* \leftarrow\$\, \mathbb{A}$	3 : $\mathbf{A}^* \leftarrow\$\, \mathcal{D}$
4 : $C \leftarrow \mathcal{A}^{\mathsf{D}(\cdot,\cdot)}(\mathbf{b}, \mathbf{A}, \mathbf{A}^*)$	4 : $C \leftarrow \mathcal{A}^{\mathsf{D}(\cdot,\cdot)}(\mathbf{b}, \mathbf{A}, \mathbf{A}^*)$
5 : $K \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}, sk, C)$	5 : $K \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}, sk, C)$
6 : $K^* \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}^*, sk, C)$	6 : $K^* \leftarrow \mathsf{Decap}'(\mathbf{b}, \mathbf{A}^*, sk, C)$
7 : **return** $K = K^* \neq \perp$	7 : **return** $K = K^* \neq \perp$

Figure 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 $\mathbf{A}$ not necessarily matching the one embedded in $\mathbf{b}$.

Theorem 5. *Let* LS-KEM *be the key encapsulation mechanism defined in Def. 13, where* $\mathbf{A}$ *is deterministically computed from the* $\mathbf{A}$*-part space* $\mathbb{A}$ *via a uniformly sampled fixed-length bit string seed* ρ *using the function* G *modeled as a RO, then the advantage of an adversary against* A-CFR-CCA *can be expressed as follows:*

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

Proof of Theorem 5. 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. $\mathbf{A}$ is obtained from splitting $\mathbf{b}$, and is thus uniformly chosen from the $\mathbf{A}$-space of the KEM. Then, $\mathbf{A}^*$ is computed from distribution $\mathcal{D}$, which is a well-spread distribution depending on the entropy of the password π with support in the $\mathbf{A}$-space of the KEM. While $pk = (\mathbf{A}, \mathbf{b})$, only the $\mathbf{b}$ term is fixed. The adversary can thus freely choose whether to encapsulate using $\mathbf{A}$ or $\mathbf{A}^*$, and they use one of those with $\mathbf{b}$ to construct C. The challenger receives C from the adversary and invokes KEM.Decap twice using both $\mathbf{A}$-parts and obtains K and K^* respectively. The encapsulation routine of the KEM (as shown in Fig. 3) 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 chosen 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 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 14 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 5. 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. [GMP22] and Cremers et al. [CDM24], we get Theorem 5. □

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 5 for any $\mathsf{LS\text{-}KEM}$ instantiated from MLWE hardness assumptions.*

5 Security Analysis of NICE-PAKE

The security analysis follows the BPR model for authenticated key exchange (AKE) [BPR00]. It models an adversary in full control of the network, but without knowledge of the password or 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 defined in [BPR00]. While many readers will already be familiar with this model, we defer the full model description to Appendix A.

Definition 17 (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 key space $\mathcal{K}$ and seed length l as defined in Def. 13, 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 $\Pi(\mathcal{U}_i, \mathcal{U}_j)$. 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. 3.4.

Theorem 6. *Let Π be the NICE-PAKE protocol defined in Fig, 1, instantiated over password dictionary $\mathcal{D}$ with a key encapsulation mechanism* LS-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 $\mathcal{A}$:*

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

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 $|\rho| = l$ is the $\mathbf{A}$-part seed size of* LS-KEM.

Proof of Theorem 6. 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 $C_{b,\text{sim}}$ instead of C_b, where $(C_{b,\text{sim}}, \overline{K}'_{\text{sim}}) \leftarrow \mathsf{KEM.Encap}(pk_{\text{sim}})$ for a fresh key pair $(sk_{\text{sim}}, pk_{\text{sim}}) \leftarrow \mathsf{KEM.KGen}()$, for an independently sampled $\mathbf{A}_{\text{sim}}$. When the honest Alice-session which has sent $(z, \mathbf{b})$ (and which is unique by Game $\mathbf{G}_2$) receives this ciphertext $C_{b,\text{sim}}$, it uses the original key $\overline{K}'$ that Bob had encapsulated instead of decapsulating $C_{b,\text{sim}}$; 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 sub-steps.

Game $\mathbf{G}_4$ (Simulate KEM Key). First, we replace C_b by $C_{b,\text{sim}}$ 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 ANO-CCA. First, note that there can be at most q_e honest Bob-sessions with an honest Alice-session, and we can apply the above modification step-by-step. Second, ANO-CCA ensures that adversaries cannot gain information by forwarding ciphertexts to other connected sessions. Via a hybrid argument, we lose a factor q_e 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 ANO-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_e 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 ANO-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 IND-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 ($\mathbf{G}_6$ - $\mathbf{G}_{10}$):

So far, 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. Due to our inability to extract $\mathcal{A}$'s guess, this event is also not detectable. However, we intuitively argue that an adversary who guessed the password correctly would be able to distinguish between messages created by honest instances and ones randomized by the simulation. Here, 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 $C_{b,\text{sim}}$ instead of ciphertext C_b, where $(C_{b,\text{sim}}, \overline{K}'_{\text{sim}}) \leftarrow \mathsf{KEM.Encap}_{\mathbf{A}_{\text{sim}}}(pk)$ for an independently sampled $\mathbf{A}_{\text{sim}}$. The honest Bob-session also uses the key $\overline{K}'_{\text{sim}}$ 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 $\mathbf{G}_6$ and injecting the challenge $\mathbf{A}$-parts $(\mathbf{A}_0, \mathbf{A}_1)$ for the received $\mathbf{b}$ from the A-SEC-CCA game into the Alice-session, using $\mathbf{A}_0$ as $\mathbf{A}_a$ resp. using $\mathbf{A}_1$ as $\mathbf{A}_{\text{sim}}$ 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 $C_{b,\text{sim}}$, which in turn does not reveal the randomized key $\overline{K}'_{\text{sim}}$, 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

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

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 $C_{b,\text{sim}}$, 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 show that this game change 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 A-SEC-CCA. This is justified through observing that $\mathcal{A}$'s advantage in $\mathbf{G}_6$ is bounded by the reduction to A-SEC-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 A-SEC-CCA experiment. It follows

$$\Pr[\mathbf{G}_6] = \Pr[\mathbf{G}_7]$$

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 SPLIT-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 SPLIT-PKU simulating game $\mathbf{G}_8$ and injecting the challenge $\mathbf{A}$-part $\mathbf{A}_1$ for the received $\mathbf{b}$-part from the SPLIT-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

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

for some adversary $\mathcal{B}_3$, where we lose a factor 2 for switching from a left-or-right game SPLIT-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 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 $\mathbf{G}_{11}$). It then follows via a reduction $\mathcal{B}_4$ to A-CFR-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 A-CFR-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 A-CFR-CCA outputs 1 if and only if $\mathcal{A}$ wins the simulated game. Hence, if the pair (C^*, K^*) in the A-CFR-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 show that this step does not affect the previous advantage. 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 decapsulates 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

$$\Pr[\mathbf{G}_9] = \Pr[\mathbf{G}_{10}]$$

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 6. □

6 Remarks on Concrete Instantiations

We note that App. C.2 and App.C.3 contain basics on some of the arguments and parameter choices we present in the following.

6.1 Remarks on Ciphertext Validation and Sanitization

To exemplify ciphertext validation and sanitizazion, we first recall that an LWE KEM ciphertext consists of the two components $(\mathbf{u}, \mathbf{v})$ as previously illustrated in Fig. 5, and that the decapsulation internally invokes a decryption routine to obtain the encrypted message $m = \mathsf{decode}(\mathbf{v} - \mathbf{s}^T\mathbf{u})$, e.g., in Kyber. We ignore the decoding step for simplicity and write for a modulus q and a decryption threshold $q/4$:

$$\begin{aligned} m &= \mathbf{b}^T\mathbf{r} + e_2 + \lceil q/2 \rfloor \cdot m \; - \; \mathbf{s}^T(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1) \\ &= \mathbf{r}^T(\mathbf{A}^T\mathbf{s} + \mathbf{e}) + e_2 + \lceil q/2 \rfloor \cdot m \; - \; \mathbf{s}^T(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1) \\ &= \underline{\mathbf{r}^T\mathbf{A}^T\mathbf{s}} + \; \mathbf{e}^T\mathbf{r} + e_2 + \lceil q/2 \rfloor \cdot m \; - \; \underline{\mathbf{r}^T\mathbf{A}^T\mathbf{s}} - \; \mathbf{e}_1^T\mathbf{s} \\ &= \lceil q/2 \rfloor \cdot m \; + \underbrace{\mathbf{e}^T\mathbf{r} + e_2 - \mathbf{e}_1^T\mathbf{s}}_{\text{small values} \leq q/4} \end{aligned}$$

Since the term $\mathbf{r}^T\mathbf{A}^T\mathbf{s}$ cancels out in the above equation, all that remains are the message m and a combination of error terms. Let this combination be ω. In order for LWE decryption and its variants to work, ω must be bounded by some β s.t. $\omega \in [2\alpha\beta^2 + \beta]$, where α is the number of samples (i.e, rows) in $\mathbf{A}$; and for any ω it must hold that $[2\alpha\beta^2 + \beta] < q/4$. Hence, each bit in $m \in \{0,1\}^{|m|}$ decodes depending on the remaining errors. That is, if they are closer to 0 or to $q/2$ [Lyu24]. Consequently, if $\mathbf{v}$ in the ciphertext was already below this threshold, one could trivially decode the message without knowledge of the secret key. On the other hand, if $\mathbf{u}$ was equal to 0, the secret key is again no longer needed, and $\mathbf{v}$ will always decode to the same message m.

Attack on NICE-PAKE. Now, consider a malicious Alice whose goal is to either guess Bob's password, or trick them into accepting the same final session key. Since Alice has complete control over the $\mathbf{b}$-part, she can craft it any way she wants. While the A-SEC-CCA property allows us to bound the advantage of Alice in distinguishing the $\mathbf{A}$-part for a given $\mathbf{b}$-part, it also means that Alice has implicit access to a decryption oracle that accepts queries on the challenge ciphertext when the $\mathbf{A}$-parts match. That is, since she already knows the secret to her own public key. Similarly, a malicious Bob aims to trick Alice by forwarding a malicious ciphertext that bypasses A-CFR-CCA and decapsulate into the same key under different encapsulation public keys. If malicious Alice forwards a small norm $\mathbf{b}$-part (e.g., generated without a secret), any honest ciphertext can be trivially and correctly decapsulated by Alice, regardless of the $\mathbf{A}$-part used by Bob in the encapsulation. Analogously, if a malicious Bob forwards a ciphertext that discards the component containing the $\mathbf{A}$-part and the encapsulation randomness, the decapsulation will proceed independently of Alice's secret key. Hence, we observe that KEM security properties only cover malicious public keys or ciphertexts that were generated using some valid secret key in the key generation, or some valid secret randomness in the encapsulation.

Ciphertext validation. As it would be futile to try to capture all the edge cases on the $\mathbf{b}$-part level (as discussed in the following sections), we observe that we only need to detect invalid values for which our new KEM properties cannot be applied. Hence, we sketch a general method to validate the KEM ciphertext inspired by the recent and ongoing

work on defining new KEM validation methods other than the standard FO re-encryption step [HK25, HHMS25].

Definition 18 (KEM Ciphertext validation oracle KCVO)**.** For a given LS-KEM with public key space $\mathcal{PK}$, ciphertext space $\mathcal{C}$, and output key space $\mathcal{K}$, we define the ciphertext validation oracle illustrated in Fig. 9 as $\mathsf{KCVO}(pk, C, K)$, where KCVO takes as input a public key pk, a ciphertext C and an output key K, and returns a bit.

$\mathsf{KCVO}(pk, C, K)$	$\mathsf{TrivDecap}(pk, C)$
1 : $K' = \mathsf{TrivDecap}(pk, C)$	1 : $\mathbf{v} \leftarrow C$
2 : **if** $K = K'$	2 : $m' \leftarrow \mathsf{PKE.decode}(\mathbf{v})$
3 : **return** 0	3 : $k' \leftarrow G_2(G_1(pk), m')$
4 : **else return** 1	4 : $K' \leftarrow F(k', C)$
	5 : **return** K'

Figure 9: LS-KEM ciphertext validation oracle KCVO and its internal function TrivDecap utilizing the hash functions F, G_2 and G_1 as defined in Fig. 3.

The KCVO internally uses a function TrivDecap (Fig. 9), which attempts to trivially decapsulate the challenge ciphertext without a secret key. The oracle checks if the key returned by TrivDecap matches its input and outputs 0 signaling an invalid ciphertext, otherwise 1. It is relatively obvious that TrivDecap can for example be instantiated using the internal decode (or compress as in Kyber [BDK+18]) function of a ForodKEM-like PKE combined with the transform used to derive the output key (e.g., the T transform in $\mathsf{FO}^{\not\perp}_m$). A proof is in this case is very straight forward, and follows directly from the fact that the correctness of the decryption only holds, if the size of errors (i.e., noise) can be handled by the decoding or decompression algorithms (cf. for example [BDK+18, Sec. 3, Proof of Thm. 1]). Thus, a proof by contradiction can be shown for an adversary who is asked to determine the output bit of the KCVO, which can be bounded by random guessing and a reduction to an LWE RLWE, or MLWE adversary. Nevertheless, we defer defining concrete bounds for future work, as each concrete KEM has different parameters for correctness depending on the exact definition of the decoding and the underlying lattice problem. Other options may also be found in performing a range check on the value $\mathbf{v}$ [HK25, HHMS25], asserting for example if $\mathbf{v} \in \mathbb{Z}_q$ (in RLWE and MLWE this corresponds to $\mathbf{v} \in R_q$), which we may consider for future work as well.

Ciphertext Sanitization. As discussed in the attack above, the main issue with an adversary crafting the $\mathbf{b}$-part at will lies within detecting small-norm public keys. However, an approach based on merely identifying such keys may not be entirely sufficient, as subtle mathematical attacks could suddenly emerge, which we are not aware of (discussed in the following section). We observe, and similar to the re-randomized KEM introduced in [DFJ+25], that the decryption threshold can tolerate adding more noise in the encapsulation routine. In other words, one can force a small norm $\mathbf{b}$ to become good again by augmenting the secret $\mathbf{s}$ with another secret $\mathbf{x}$ from the same distribution s.t. trivial decryption is no longer possible for a malicious Alice. That is, if there is no secret at all, in the extreme case $\mathbf{b} = \mathbf{A}$, the new value $\mathbf{x}$ provides sufficient noise to prevent correct decryption. Similarly, an already good $\mathbf{b}$, or one that was not computed with sufficient noise, would not be affected by the added value $\mathbf{x}$, if a decapsualtor can obtain the sanitization randomness in honest executions. However, in contrast to SR-AKE, a normal PAKE cannot deploy a third instance (a reverse firewall). Hence, we aim at finding a method to securely send a sanitizing secret to Alice that is chosen by Bob at random.

Probably, this method will imply incorporating the password in some way, and needs to be handled carefully. Intuitively, and informally speaking, we assume that a construction that ensures that only the correct password leads to a meaningful and usable sanitization randomness can be safely combined with A-CFR-CCA. That is, since the same password guess applies for both the seed ρ and this new secret $\mathbf{x}$ at the same time, which means that an adversary will carry out an offline password attack on pairs of the form $(\mathbf{A}, \mathbf{x})$. The challenge here is then to ensure that each guessed pair corresponds to a decapsualtion that is indistinguishable from the real one with overwhelming probability. A natural conjecture of this idea is to try to employ Split-KEMs, as they inherently allow both sides to contribute to the key establishment. That being said, we will attempt these ideas in future work.

6.2 Remarks on the Concrete Value of $\varepsilon_{\text{whLWE}}$ in Theorem 4

There are two incomparable reductions for Extended LWE applicable to our use-case: one from O'Neill et al. in [OPW11] that yields a $(1/q)$ multiplicative advantage loss per application, and one from Brakerski et al. [BLP+13] that yields a negligible advantage loss, up to a mild change in parameters. In the first case [OPW11], 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 [BLP+13], 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 [BD20], 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. [CKK+21, LKK+18] and Liu et al. [LSTW25], we can calculate a small loss in concrete bit-security for a FrodoKEM-instantiation as follows: Following Theorem 2, 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 [APS15] to find optimal choices.

6.3 A Complete Break 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 $\mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}$ 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 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 $i \in [k]$)

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

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 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 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 $v = \mathbf{b}\mathbf{r} + \mathbf{e}''$, 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 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 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 $(4, 4^2, 4^3, 4^4) = (4, 16, 64, 256) \in R_{3329}^4$. 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.

6.4 (Speculatively) Tweaking Kyber 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 [ACPS09]. 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 [GKPV10, BD20] 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 [BBPS19, LW20]. 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.

7 Conclusion

In this work, we addressed the issues accompanying quantum-resistant PAKE designs and constructions from lattice-based 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. 2) 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 KEMs beyond those concerned with the uniformity and anonymity of keys, and the final key security. Other interesting approaches can surely be found in defining the concrete security of the introduced validation oracle, sanitizing the ciphertext, or employing Split-KEMs.

Acknowledgment

We thank Marc Fischlin, Hart Montgomery, Scott Simon, Jonathan Katz, Jintai Ding, Shuichi Katsumata, Julia Hesse, Johannes Ottenhues, Marcel Tiplet 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 former 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, to which time Daniel Apon contributed to this work while at the MITRE corporation. All Rights Reserved. Approved for Public Release. Distribution Unlimited. Case Number 24-3386.

References

[AA16] Jacob Alperin-Sheriff and Daniel Apon. Dimension-preserving reductions from LWE to LWR. Cryptology ePrint Archive, Report 2016/589, 2016. URL:

https://eprint.iacr.org/2016/589.

[ABB+20] Michel Abdalla, Manuel Barbosa, Tatiana Bradley, Stanislaw Jarecki, Jonathan Katz, and Jiayu Xu. Universally composable relaxed password authenticated key exchange. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 278–307, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-56784-2_10.

[ABJ25] Afonso Arriaga, Manuel Barbosa, and Stanislaw Jarecki. NoIC: PAKE from KEM without ideal ciphers. Cryptology ePrint Archive, Report 2025/231, 2025. URL: https://eprint.iacr.org/2025/231.

[ABJS24] Afonso Arriaga, Manuel Barbosa, Stanislaw Jarecki, and Marjan Skrobot. C'est Très CHIC: A compact password-authenticated key exchange from lattice-based KEM. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part V*, volume 15488 of *LNCS*, pages 3–33, Kolkata, India, December 9–13, 2024. Springer, Singapore, Singapore. doi:10.1007/978-981-96-0935-2_1.

[ABPS24] Gorjan Alagic, Chen Bai, Alexander Poremba, and Kaiyan Shi. On the two-sided permutation inversion problem. *CiC*, 1(1):27, 2024. doi:10.62056/a0qj89n4e.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618, Santa Barbara, CA, USA, August 16–20, 2009. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-03356-8_35.

[AEK+22] Michel Abdalla, Thorsten Eisenhofer, Eike Kiltz, Sabrina Kunzweiler, and Doreen Riepel. Password-authenticated key exchange from group actions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 699–728, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-15979-4_24.

[AFL16] Daniel Apon, Xiong Fan, and Feng-Hao Liu. Deniable attribute based encryption for branching programs from LWE. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 299–329, Beijing, China, October 31 – November 3, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53644-5_12.

[AHHR24] Nouri Alnahawi, Kathrin Hövelmanns, Andreas Hülsing, and Silvia Ritsch. Towards post-quantum secure PAKE - A tight security proof for OCAKE in the BPR model. In Markulf Kohlweiss, Roberto Di Pietro, and Alastair R. Beresford, editors, *CANS 2024, Part II*, volume 14906 of *LNCS*, pages 191–212, Cambridge, UK, September 24–27, 2024. Springer, Singapore, Singapore. doi:10.1007/978-981-97-8016-7_9.

[AHMW25] Nouri Alnahawi, David Haas, Erik Mauß, and Alexander Wiesmaier. SoK: PQC PAKEs - design, security and performance. Cryptology ePrint Archive, Report 2025/119, 2025. URL: https://eprint.iacr.org/2025/119.

[AP05] Michel Abdalla and David Pointcheval. Simple password-based encrypted key exchange protocols. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 191–208, San Francisco, CA, USA, February 14–18, 2005.

Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-30574-3_14.

[AP12] Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 334–352, Darmstadt, Germany, May 21–23, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-30057-8_20.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Math. Cryptol.*, 9(3):169–203, 2015. URL: http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml.

[AR17] Gorjan Alagic and Alexander Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 65–93, Paris, France, April 30 – May 4, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-56617-7_3.

[BBPS19] Madalina Bolboceanu, Zvika Brakerski, Renen Perlman, and Devika Sharma. Order-LWE and the hardness of ring-LWE with entropic secrets. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 91–120, Kobe, Japan, December 8–12, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-34621-8_4.

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1006–1018, Vienna, Austria, October 24–28, 2016. ACM Press. doi:10.1145/2976749.2978425.

[BCP+23] Hugo Beguinet, Céline Chevalier, David Pointcheval, Thomas Ricosset, and Mélissa Rossi. GeT a CAKE: Generic transformations from key encaspulation mechanisms to password authenticated key exchanges. In Mehdi Tibouchi and Xiaofeng Wang, editors, *ACNS 2023, Part II*, volume 13906 of *LNCS*, pages 516–538, Kyoto, Japan, June 19–22, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-33491-7_19.

[BD20] Zvika Brakerski and Nico Döttling. Hardness of LWE on general entropic distributions. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 551–575, Zagreb, Croatia, May 10–14, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-45724-2_19.

[BDK+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367, London, United Kingdom, April 24–26, 2018. IEEE Computer Society Press. doi:10.1109/EuroSP.2018.00032.

[BFG+20] Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson, and Douglas Stebila. Towards post-quantum security for Signal's X3DH handshake. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 404–430, Halifax, NS, Canada

(Virtual Event), October 21-23, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-81652-0_16.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325, Cambridge, MA, USA, January 8–10, 2012. ACM. doi:10.1145/2090236.2090262.

[Bla06] John Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. In Matthew J. B. Robshaw, editor, *FSE 2006*, volume 4047 of *LNCS*, pages 328–340, Graz, Austria, March 15–17, 2006. Springer Berlin Heidelberg, Germany. doi:10.1007/11799313_21.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428, Santa Barbara, CA, USA, August 18–22, 2013. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-40041-4_23.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press. doi:10.1145/2488608.2488680.

[BM92] Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy*, pages 72–84. IEEE Computer Society Press, May 1992. doi:10.1109/RISP.1992.213269.

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 139–155, Bruges, Belgium, May 14–18, 2000. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-45539-6_11.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. doi:10.1145/168588.168596.

[CC17] Ran Canetti and Yilei Chen. Constraint-hiding constrained PRFs for NC^1 from LWE. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 446–476, Paris, France, April 30 – May 4, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-56620-7_16.

[CDM24] Cas Cremers, Alexander Dax, and Niklas Medinger. Keeping up with the KEMs: Stronger security notions for KEMs and automated analysis of KEM-based protocols. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 1046–1060, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press. doi:10.1145/3658644.3670283.

[CKK+21] Jung Hee Cheon, Dongwoo Kim, Duhyeong Kim, Joohee Lee, Junbum Shin, and Yongsoo Song. Lattice-based secure biometric authentication for hamming

distance. In Joonsang Baek and Sushmita Ruj, editors, *ACISP 21*, volume 13083 of *LNCS*, pages 653–672, Virtual Event, December 1–3, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-90567-5_33.

[CKS23] Dharminder Chaudhary, Uddeshaya Kumar, and Kashif Saleem. A construction of three party post quantum secure authenticated key exchange using ring learning with errors and ECC cryptography. *IEEE Access*, 11:136947–136957, 2023. doi:10.1109/ACCESS.2023.3325886.

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Report 2019/428, 2019. URL: https://eprint.iacr.org/2019/428.

[DAL+17] Jintai Ding, Saed Alsayigh, Jean Lancrenon, Saraswathy RV, and Michael Snook. Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In Helena Handschuh, editor, *CT-RSA 2017*, volume 10159 of *LNCS*, pages 183–204, San Francisco, CA, USA, February 14–17, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-52153-4_11.

[DFJ+25] Kévin Duverger, Pierre-Alain Fouque, Charlie Jacomme, Guilhem Niot, and Cristina Onete. Subversion-resilient key-exchange in the post-quantum world. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025*, pages 1200–1214, Taipei, Taiwan, October 13–17, 2025. ACM Press. doi:10.1145/3719027.3765165.

[DJK+25] Yevgeniy Dodis, Daniel Jost, Shuichi Katsumata, Thomas Prest, and Rolfe Schmidt. Triple ratchet: A bandwidth efficient hybrid-secure signal protocol. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VIII*, volume 15608 of *LNCS*, pages 302–331, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91101-9_11.

[DKM+24] Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 219–248, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-58723-8_8.

[DS16] Yuanxi Dai and John P. Steinberger. Indifferentiability of 8-round Feistel networks. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 95–120, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53018-4_4.

[EEN+24] Muhammed F. Esgin, Thomas Espitau, Guilhem Niot, Thomas Prest, Amin Sakzad, and Ron Steinfeld. Plover: Masking-friendly hash-and-sign lattice signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VII*, volume 14657 of *LNCS*, pages 316–345, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-58754-2_12.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554, Santa Barbara, CA, USA, August 15–19,

1999. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-48405-1_34.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. doi:10.1007/s00145-011-9114-1.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 230–240, Tsinghua University, Beijing, China, January 5–7, 2010. Tsinghua University Press.

[GMP22] Paul Grubbs, Varun Maram, and Kenneth G. Paterson. Anonymous, robust post-quantum public key encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 402–432, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-07082-2_15.

[GRSV25] Felix Günther, Michael Rosenberg, Douglas Stebila, and Shannon Veitch. Hybrid obfuscated key exchange and KEMs. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 575–609, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-032-01881-6_18.

[GSG+23] Songhui Guo, Yunfan Song, Song Guo, Yeming Yang, and Shuaichao Song. Three-party password authentication and key exchange protocol based on MLWE. *Symmetry*, 15(9):1750, 2023. URL: https://doi.org/10.3390/sym15091750, doi:10.3390/SYM15091750.

[Hes20] Julia Hesse. Separating symmetric and asymmetric password-authenticated key exchange. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20*, volume 12238 of *LNCS*, pages 579–599, Amalfi, Italy, September 14–16, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-57990-6_29.

[HHKR25] Kathrin Hövelmanns, Andreas Hülsing, Mikhail Kudinov, and Silvia Ritsch. CAKE requires programming - on the provable post-quantum security of (O)CAKE. Cryptology ePrint Archive, Report 2025/458, 2025. URL: https://eprint.iacr.org/2025/458.

[HHMS25] Kathrin Hövelmanns, Andreas Hülsing, Christian Majenz, and Fabrizio Sisinni. (Un)breakable curses - re-encryption in the Fujisaki-Okamoto transform. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 245–274, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91124-8_9.

[HK25] Kathrin Hövelmanns and Mikhail A. Kudinov. Treating dishonest ciphertexts in post-quantum KEMs - explicit vs. implicit rejection in the FO transform. In Ruben Niederhagen and Markku-Juhani O. Saarinen, editors, *Post-Quantum Cryptography - 16th International Workshop, PQCrypto 2025, Part II*, pages 325–350, Taipei, Taiwan, April 08–10, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-86602-9_12.

[HR25] Julia Hesse and Michael Rosenberg. PAKE combiners and efficient post-quantum instantiations. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 395–420, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91124-8_14.

[HTTY24] Shuya Hanai, Keisuke Tanaka, Masayuki Tezuka, and Yusuke Yoshida. Universally composable relaxed asymmetric password-authenticated key exchange. In Clemente Galdi and Duong Hieu Phan, editors, *SCN 24, Part II*, volume 14974 of *LNCS*, pages 272–293, Amalfi, Italy, September 11–13, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-71073-5_13.

[HvO22] Feng Hao and Paul C. van Oorschot. SoK: Password-authenticated key exchange - theory, practice, standardization and real-world lessons. In Yuji Suga, Kouichi Sakurai, Xuhua Ding, and Kazue Sako, editors, *ASIACCS 22*, pages 697–711, Nagasaki, Japan, May 30 – June 3, 2022. ACM Press. doi:10.1145/3488932.3523256.

[HY18] Akinori Hosoyamada and Kan Yasuda. Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 275–304, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-030-03326-2_10.

[Jab97] David P. Jablon. Extended password key exchange protocols immune to dictionary attacks. In *6th Workshop on Enabling Technologies (WET-ICE '97), Infrastructure for Collaborative Enterprises, 18-20 June 1997, MIT, Cambridge, MA, USA, Proceedings*, pages 248–255. IEEE Computer Society, 1997. doi:10.1109/ENABL.1997.630822.

[JRX25] Jake Januzelli, Lawrence Roy, and Jiayu Xu. Under what conditions is encrypted key exchange actually secure? In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 451–481, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91124-8_16.

[KLSS23] Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 549–580, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-38554-4_18.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange from lattices. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 636–652, Tokyo, Japan, December 6–10, 2009. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-10366-7_37.

[LKK+18] Joohee Lee, Dongwoo Kim, Duhyeong Kim, Yongsoo Song, Junbum Shin, and Jung Hee Cheon. Instant privacy-preserving biometric authentication for hamming distance. Cryptology ePrint Archive, Report 2018/1214, 2018. URL: https://eprint.iacr.org/2018/1214.

[LL25] You Lyu and Shengli Liu. Hybrid password authentication key exchange in the UC framework. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 421–450, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91124-8_15.

[LLH24] You Lyu, Shengli Liu, and Shuai Han. Universal composable password authenticated key exchange for the post-quantum world. In Marc Joye and

Gregor Leander, editors, *EUROCRYPT 2024, Part VII*, volume 14657 of *LNCS*, pages 120–150, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-58754-2_5.

[LP25] Oleksandra Lapiha and Thomas Prest. A lattice-based IND-CCA threshold KEM from the BCHK+ transform. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part III*, volume 16247 of *LNCS*, pages 461–494, Melbourne, VIC, Australia, December 8–12, 2025. Springer, Singapore, Singapore. doi:10.1007/978-981-95-5099-9_15.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23, French Riviera, May 30 – June 3, 2010. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-13190-5_1.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015. doi:10.1007/s10623-014-9938-4.

[LSTW25] Zeyu Liu, Katerina Sotiraki, Eran Tromer, and Yunhao Wang. Snake-eye resistant PKE from LWE for oblivious message retrieval and robust encryption. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 126–156, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91131-6_5.

[LW20] Feng-Hao Liu and Zhedong Wang. Rounding in the rings. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 296–326, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-56880-1_11.

[Lyu24] Vadim Lyubashevsky. Basic lattice cryptography: The concepts behind Kyber (ML-KEM) and Dilithium (ML-DSA). Cryptology ePrint Archive, Report 2024/1287, 2024. URL: https://eprint.iacr.org/2024/1287.

[MMW25] Christian Majenz, Giulio Malavolta, and Michael Walter. Permutation superposition oracles for quantum query lower bounds. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 1508–1519, Prague, Czechia, June 23–27, 2025. ACM Press. doi:10.1145/3717823.3718266.

[MRR20] Ian McQuoid, Mike Rosulek, and Lawrence Roy. Minimal symmetric PAKE and 1-out-of-N OT from programmable-once public functions. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 425–442, Virtual Event, USA, November 9–13, 2020. ACM Press. doi:10.1145/3372297.3417870.

[MX23a] Varun Maram and Keita Xagawa. Post-quantum anonymity of Kyber. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 3–35, Atlanta, GA, USA, May 7–10, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-31368-4_1.

[MX23b] Ian McQuoid and Jiayu Xu. An efficient strong asymmetric PAKE compiler instantiable from group actions. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VIII*, volume 14445 of *LNCS*, pages 176–207, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8742-9_6.

[Nio25] Guilhem Niot. Practical deniable post-quantum X3DH: A lightweight split-KEM for K-waay. In *ASIACCS 25*, pages 298–312, Hanoi, Vietnam, August 25–29, 2025. ACM Press. doi:10.1145/3708821.3736192.

[OPW11] Adam O'Neill, Chris Peikert, and Brent Waters. Bi-deniable public-key encryption. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 525–542, Santa Barbara, CA, USA, August 14–18, 2011. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-22792-9_30.

[PZ23] Jiaxin Pan and Runzhi Zeng. A generic construction of tightly secure password-based authenticated key exchange. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VIII*, volume 14445 of *LNCS*, pages 143–175, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8742-9_5.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press. doi:10.1145/1060590.1060603.

[Reg06] Oded Regev. Lattice-based cryptography (invited talk). In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 131–141, Santa Barbara, CA, USA, August 20–24, 2006. Springer Berlin Heidelberg, Germany. doi:10.1007/11818175_8.

[Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 2010. doi:10.1109/ccc.2010.26.

[RHCB22] Prasanna Ravi, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Lattice-based key-sharing schemes: A survey. *ACM Comput. Surv.*, 54(1):9:1–9:39, 2022. doi:10.1145/3422178.

[Ros21] Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments. *CoRR*, abs/2103.08975, 2021. URL: https://arxiv.org/abs/2103.08975, arXiv:2103.08975.

[SGJ23] Bruno Freitas Dos Santos, Yanqi Gu, and Stanislaw Jarecki. Randomized half-ideal cipher on groups with applications to UC (a)PAKE. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 128–156, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30589-4_5.

[SS19] Shingo Sato and Junji Shikata. SO-CCA secure PKE in the quantum random oracle model or the quantum ideal cipher model. In Martin Albrecht, editor, *17th IMA International Conference on Cryptography and Coding*, volume 11929 of *LNCS*, pages 317–341, Oxford, UK, December 16–18, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-35199-1_16.

[Tan21] 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. Publisher: Hindawi. doi:10.1155/2021/6952869.

[TY19] Shintaro Terada and Kazuki Yoneyama. Password-based authenticated key exchange from standard isogeny assumptions. In Ron Steinfeld and Tsz Hon Yuen, editors, *ProvSec 2019*, volume 11821 of *LNCS*, pages 41–56, Cairns,

QLD, Australia, October 1–4, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-31919-9_3.

[Unr21] Dominique Unruh. Compressed permutation oracles (and the collision-resistance of Sponge/SHA3). Cryptology ePrint Archive, Report 2021/062, 2021. URL: https://eprint.iacr.org/2021/062.

[Unr23] Dominique Unruh. Towards compressed permutation oracles. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part IV*, volume 14441 of *LNCS*, pages 369–400, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8730-6_12.

[VJW+25] Jelle Vos, Stanislaw Jarecki, Christopher A. Wood, Cathie Yun, Steve Myers, and Yannick Sierra. A hybrid asymmetric password-authenticated key exchange in the random oracle model. Cryptology ePrint Archive, Report 2025/1343, 2025. URL: https://eprint.iacr.org/2025/1343.

[Wan22] 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. doi:10.1007/978-3-031-30623-5_3.

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 568–597, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-77886-6_20.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-26951-7_9.

[ZHS14] Hongfeng Zhu, 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(4):672–689, 2014. URL: http://bit.kuas.edu.tw/%7Ejihmsp/2014/vol5/JIH-MSP-2014-04-009.pdf.

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[5]. 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 hash functions $\mathcal{H}$.

- `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.

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 19 (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

[5]This simplified description is sufficient for symmetric PAKEs.

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.

B 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 [RHCB22]. 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 [Reg06]. 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 [Reg10] 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 [Reg05] 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 [Reg10], yet was proven secure through a reduction to the SVP problem in [Reg05].

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 [Reg10]. 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 [LPR10] 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. [BGV12] and further studied by Langlois and Stehlé [LS15]. 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 $\mathbf{A} \in R_q^{m \times d}$.

C Further Lattice Assumptions

C.1 Extended Learning with Errors (eLWE).

The extended Learning with Errors assumption (eLWE) was first defined by O'Neill et al. in [OPW11] in the context of proving security of deniable encryption from lattices (cf. Apon et al. [AFL16]). eLWE has additionally been used by Alperin-Sheriff and Peikert [AP12] to show security theorems for circular and key-dependent message security of lattice identity-based encryption; and to show the classical hardness of LWE itself by Brakerski et al. [BLP+13]. The 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 20 (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.

Prior works define a multi-hint version of eLWE, but we will only need the single-hint version as stated above. The hardness of eLWE follows from LWE, summarized as follows.

Theorem 7. *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 21 (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. [OPW11]), 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.

C.2 Non-uniform Module Learning with Errors (NMLWE)

For completeness, we define a module-lattice version of NLWE, adapted from Canetti and Chen [CC17, Lemma 2.13]

Definition 22 (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 $(\mathbf{D}, \mathbf{K}\mathbf{D} + \mathbf{E}) \in (R^{m \times m} \times R^{1 \times m})$ 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 [CC17, 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 $\gamma_\sigma = D_{R^m,\sigma}^{1 \times m}, \chi_\sigma = D_{R,\sigma}$.*

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 $\mathsf{P/poly}$ from standard lattice assumptions. We view this as negative evidence for obtaining such proof in the near term.

C.3 Extended Module Learning with Errors (eMLWE)

For completeness, following Alperin-Sheriff and Apon [AA16], we define a module-lattice version of eLWE as follows:

Definition 23 (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 [AA16, 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 [AP12]:* $\mathsf{eMLWE}_{d,w,q,\alpha,k}$ *that reduces the advantage by at most* q^k*, multiplicatively.*
2. *Following [BLP+13]:* $\mathsf{eMLWE}_{d+k,w,q,(\alpha^2+r^2)^{1/2},k}$*, 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.