

Lattice-based NIKE with optimal tightness

Roman Langrehr¹

Si An Oliver Tran²

¹ University of Waterloo, Canada

roman.langrehr@uwaterloo.ca

² ETH Zurich, Switzerland

oliver.tran@alumni.ethz.ch

Abstract

In this work we present a new variant of the non-interactive key exchange (NIKE) scheme based on the learning with errors (LWE) assumption and prove its security with a security reduction that incurs a security loss that is only linear in the number of users. This improves upon all prior reductions for lattice-based NIKE schemes, which had a security loss that is quadratic in the number of users. Our tight reduction can handle the setting with super-polynomial modulus-to-noise ratio and negligible correctness error as well as the more challenging setting with polynomial modulus-to-noise ratio and inverse polynomial correctness error.

We also give a matching lower bound on the tightness for a natural class of lattice-based NIKE schemes (that captures all existing variants of lattice-based NIKE), showing that our security loss is optimal (up to constant factors). This generalizes a lower bound by Hesse, Hofheinz and Kohl (Crypto 2018) and is the first lower bound for the tightness of lattice-based NIKE schemes. Several previous lower bounds for the tightness of NIKE exist, but none of them can be applied to lattice-based schemes.

Keywords: Non-interactive key exchange, Learning with errors, Tightness

1 Introduction

Non-interactive key exchange. Non-interactive key exchange (NIKE) is a cryptographic primitive that allows any pair of users to obtain a common shared key using a public-key infrastructure (PKI) and no communication (besides what is needed to set up the PKI). More concretely, each user generates a public/secret key pair and can obtain a shared key with every other user using their own secret key and the other user’s public key. A famous example of such a scheme is the Diffie-Hellman key exchange [DH76], which is the basis of many protocols.

However, with the threat that quantum computers pose on many cryptographic assumptions, such as the Diffie-Hellman key exchange, it is important to investigate possible post-quantum alternatives. For NIKE, post-quantum candidates are the isogeny-based schemes CSIDH [Cas+18] and CORAL [Bas+26], as well as lattice-based constructions.

The lattice-based NIKE construction appeared in the literature first in works by Ding, Xie, and Lin [DXL12] and Peikert [Pei14], but the idea of this construction was folklore before these works appeared, and was subsequently studied in [Guo+20; Lan23; Gaj+24].

Applications. In lattice-based cryptography, it is more common to derive common shared keys by using a key encapsulation mechanism (KEM), such as the NIST-standardized ML-KEM aka Kyber [Sch+22], than to use a NIKE. This is because lattice-based KEMs are currently significantly more efficient than lattice-based NIKES and much better understood. However, there are several applications of NIKE where using a KEM instead is difficult or impossible.

Firstly, changing protocols that are based on NIKES (such as the Diffie-Hellman key exchange) is a challenging task that can require significant changes to the protocol and can make additional interaction

*Part of the work was done while the author was affiliated with ETH Zurich.

or round-trips necessary (see e.g. [SSW20] for TLS, [Hül+21] for WireGuard, and [Ang+22] for the Noise protocol framework). Secondly, some protocols or schemes rely on the non-interactive nature of NIKE in way that cannot be replaced by KEMs or other interactive key exchange variants. One such example is deniable authentication [BMP04; Dod+09; Dam+20; Alw+23; GJK24], which exploits that in a NIKE scheme both parties can obtain the shared key (at least, assuming that both parties generated their NIKE key pairs honestly. Without this assumption, deniable authentication seems very challenging to achieve [FL25]). Another application of NIKE, where a KEM cannot be used instead, is decentralizing functional encryption [Cho+20; GY24; NPS25].

Security model. The security model we consider in this work is adaptive honest key registration (HKR) security, introduced in [CKS08; Fre+13]. In this model the adversary can make adaptive queries to register many users, corrupt these users (which means learning their secret key) and reveal the shared key between a pair of users. Additionally, the adversary can make a single challenge query where it obtains a real or random shared key between two users of its choice. The adversary wins if it guesses correctly which of these two variants was played without making any query that trivially reveals this challenge shared key.

Tightness. The most widespread approach to analyze the security of a scheme is to give a reduction to an assumption, that should be much simpler and whose hardness is already well understood. However, these reductions often come at a loss, meaning that there is a gap between the hardness guarantee of the scheme compared to the assumption. This is called the security loss and can be, for example, quantified as follows (as done in [Bad+16, Definiton 9]). The security loss $\ell(r-\Lambda)$ of a reduction $r-\Lambda$, that runs the adversary $\mathcal{A}$ r times, is a function of the security parameter that satisfies

$$\frac{r \cdot T(\mathcal{A}) + T(r-\Lambda)}{\text{Adv}_{r-\Lambda, \mathcal{A}, \text{NICA}}(\lambda)} \leq \ell(r-\Lambda) \cdot \frac{T(\mathcal{A})}{\text{Adv}_{\mathcal{A}, \text{NIKE}}(\lambda)}$$

where $T(\mathcal{A})$ is the runtime of the adversary and $T(r-\Lambda)$ is the runtime overhead of the reduction, $\text{Adv}_{r-\Lambda, \mathcal{A}, \text{NICA}}(\lambda)$ is the advantage of the reduction (when run with the adversary) against the underlying assumption (called NICA here) and $\text{Adv}_{\mathcal{A}, \text{NIKE}}(\lambda)$ is the advantage of the adversary in the security game of the scheme (here, the NIKE).

To simplify our results, we ignore the term $T(r-\Lambda)$ in the above definition for our results. This is meaningful, when the runtime overhead of the reduction is much smaller than the time needed to run $\mathcal{A}$ r times. This is typically the case and a very common simplification in the literature, even though it is not always made explicit.

1.1 Related works on tightness of NIKE

All previous lattice-based NIKE schemes were only proven secure with a security loss of $\mathcal{O}(N^2)$, where N is the number of users [DXL12; Pei14; Lan23; Gaj+24]. In fact, most NIKE constructions suffer from this quadratic security loss. This can be explained with a lower bound by Bader et al. [Bad+16], who show that a loss of $\Omega(N^2)$ is unavoidable for a large class of NIKE schemes. Their result captures most NIKE schemes in the literature, including the traditional Diffie-Hellman key exchange.

The first work that bypasses their impossibility result is by Hesse, Hofheinz and Kohl [HHK18], who give a (group-based) NIKE construction that escapes the requirements of their impossibility result and achieves a $\mathcal{O}(N)$ security loss. They also give a new matching lower bound, that shows that a $\Omega(N)$ security loss is unavoidable for a larger class of NIKes, including theirs.

The only NIKE scheme known that can achieve a tighter reduction was presented in [Hes+21] (and is also group-based). However, this comes at the price of having public and secret keys that grow with the number of users (which, for concrete efficiency, probably offsets any advantage a tight reduction might have). If one is willing to allow a key size of $\mathcal{O}(N)$, they get a security loss of only $\mathcal{O}(\log(N))$. They also give a lower bound showing that these large keys are in some sense necessary for such a tight reduction. This lower bound assumes an inner-product structure of the NIKE. Formally it is incomparable to the requirements by [Bad+16; HHK18], but it also captures a large number of schemes (including, the Diffie-Hellman NIKE, the NIKE in [HHK18], and their own scheme). Finally, they combine this with lattice techniques to reduce the key size at the cost of having a less adaptive security game. For the

purpose of this paper, we do not consider this as a lattice-based scheme because its security can only be proven based on group *and* lattice-based assumption (and thus is not a post-quantum scheme).

Rao [Rao14] gave a lower bound that is similar to [Bad+16] for multi-party NIKE, where a group of users (and not just pairs of users) can derive a shared key. Their lower bound is exponential in the maximum size of the group that can derive a common shared key. This lower bound was bypassed only recently by [KWZ22] with a highly involved construction based on indistinguishability obfuscation.

Interestingly, none of the previous lower bounds can handle lattice-based NIKE schemes (see Section 2). One reason is that they all require the NIKE to be perfectly correct, but none of the lattice-based NIKES is perfectly correct (and this seems very hard to achieve [Guo+20]). This leaves the problem of what is the optimal security loss for lattice-based NIKES wide open.

1.2 Our result

In this work, we present a lattice-based NIKE scheme that has a security loss of $\mathcal{O}(N)$ as well as a matching lower bound of $\Omega(N)$ that captures a wider class of NIKE schemes than [HHK18], including all known variants of lattice-based NIKE schemes.

Technically, we follow the template of [HHK18] to construct a NIKE from a hash-proof system (HPS). The technical difficulty here is that lattice-based HPSs only achieve weaker properties than their group-based counterparts that [HHK18] used. This creates some technical challenges, in particular in the setting with polynomial modulus-to-noise ratio where we have a non-negligible correctness error (which is in some sense unavoidable [Guo+20]). In this setting, we combine this approach with the techniques introduced in [Lan23] to prove adaptive HKR security in this setting.

We then generalize the impossibility result by [HHK18] to capture lattice-based schemes and obtain our matching lower bound. The difficulty here is that the noise terms that are needed in lattice-based schemes can cause errors in the resulting shared keys. Even when these errors happen only with negligible correctness error, they would cause the metareduction argument to fail. We solve these problems by introducing additional checks on pairs of users, that ensure that all pre-rounding shared keys are far away from a rounding boundary and thus noise terms cannot cause these errors.

Similar to previous works, our lower bound captures only reductions that use the adversary in a black-box way (but can rewind the adversary) and reduce the security to a non-interactive assumption.

1.3 Extensions and open problems

Stronger security models. For many scenarios, a more realistic notion is the variant where the adversary can make many challenge queries (and either all of them are answered with a real key or all of them are answered with a random key). This model was previously only considered in [Hes+21]. We sketch how our positive result (and, in fact, also the one in [HHK18]) can be extended to this setting without increasing the security loss.

Another strengthening of the security model is to consider dishonest key registrations (DKR). In this model, the adversary can register maliciously generated public keys and reveal shared keys of honest users with these malicious public keys. A generic approach to achieve this security notion is to equip public keys with a non-interactive zero-knowledge proof of knowledge (NIZKPoK) of the secret key [HHK18]. However, the generic result by [HHK18] requires the NIKE to be perfectly correct and thus cannot be applied directly to lattice-based schemes. The work [Lan23] showed that this approach can still be used in combination with the noise smudging technique (which requires super-polynomial modulus-to-noise ratio). When the NIZKPoK used for this approach is tightly secure, this does not decrease the tightness of the scheme. Concurrently, [Gaj+24] showed a different approach that uses NIZKPoKs and the (quantum) random oracle model. Their scheme has a very large security loss that is quadratic in the number of random oracle queries of the adversary.¹

The technique in [Lan23] is compatible with our NIKE construction with super-polynomial modulus-to-noise ratio and can be used to upgrade it to DKR security. This upgrade preserves the tightness assuming the NIZKPoK is tightly secure.

¹The loss they incur is the number of random oracle queries times the query depth of the adversary. This can be upper bounded by the square of the number of random oracle queries, but for adversaries that make their QROM queries in parallel the security loss is actually better.

An interesting open problem is how DKR security can be achieved in the setting with polynomial modulus-to-noise ratio. The work [Lan23] shows that adding NIZKPoKs is not sufficient in this setting.

Counting corruptions. Moreover, we quantify in this work the security loss by the number of users the adversary registers. A recent work [Bel+24] argued that it is beneficial to count the number of corrupted users instead, which is in many real-world scenarios much lower than the number of total users. They also present a generic framework that shows that in many settings it is possible to get a security loss that only grows with the number of corrupted users. However, their framework does not capture the security notions needed for non-interactive key exchange. We leave it as an open problem whether the security loss for our construction (or NIKEs, in general) can be improved to grow only (linearly) in the number of corrupted users.

Such a result would *not* contradict our lower bound showing that a security loss of $\Omega(N)$ is unavoidable, because the number of corruptions that the metareduction makes is approximately the same as the total number of users.

2 Technical overview

Lattice-based NIKE. We first recall the folklore lattice-based NIKE, that has been studied (in a few variations) in prior work [DXL12; Pei14; Lan23; Gaj+24].

In this NIKE we have two different types of key pairs, which we call left and right keys. A shared key can only be computed between a left and right key pair (using either the left or the right secret key and the right or left public key, respectively). This “asymmetric NIKE” can be easily converted into a regular NIKE by generating a left and right key pair for every user.

The scheme uses a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times n}$ as global setup. Left public keys are of the form $\mathbf{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ and right public keys are of the form $\mathbf{pk}_R := \mathbf{A} \mathbf{s}_R + \mathbf{e}_R$ where $\mathbf{s}_L, \mathbf{e}_L, \mathbf{s}_R, \mathbf{e}_R \in \mathbb{Z}_q^n$ are short vectors and $\mathbf{s}_L/\mathbf{s}_R$ are used as secret keys. This allows both parties to obtain approximately the same shared key (which we will call the pre-rounding shared key in this work), by taking the inner product of their secret key and the other user’s public key. For security reasons, we also add small noise terms $e'_L, e'_R \in \mathbb{Z}_q$ to these pre-rounding shared keys.²

$$\mathbf{s}_L^\top \cdot \mathbf{pk}_R + e'_L = \mathbf{s}_L^\top \mathbf{A} \mathbf{s}_R + \mathbf{s}_L^\top \mathbf{e}_R + e'_L \approx \mathbf{s}_L^\top \mathbf{A} \mathbf{s}_R + \mathbf{e}_L^\top \mathbf{s}_R + e'_R = \mathbf{pk}_L \cdot \mathbf{s}_R + e'_R.$$

By rounding these values, both parties can get the same shared key with high probability. If the modulus-to-noise ratio is super-polynomial, we get a negligible correctness error. For polynomial modulus-to-noise ratio the correctness error is inverse polynomial. Guo et al. [Guo+20] show that rounding is essentially optimal in this setting, and it seems very challenging to construct lattice-based NIKE schemes with better correctness. For simplicity, we focus in this overview on the setting with negligible correctness error.

We can argue security of the NIKE as follows. Assume for now that we have just one left and one right user, the adversary gets access to their public keys (and, of course, the public matrix $\mathbf{A}$), and tries to distinguish the shared key computed by the left user from a random key. We can then use the learning with errors (LWE) assumption to switch from the real security game to hybrid game, where the left user’s public key is uniformly random in $\mathbb{Z}_q^n$ (instead of having the form $\mathbf{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$). This is straightforward, since the LWE assumption tells us $(\mathbf{A}, \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top) \approx_c (\mathbf{A}, \mathbf{pk}_L)$ for $\mathbf{pk}_L \xleftarrow{\$} \mathbb{Z}_q^n$. In a second step, we switch to a hybrid where the pre-rounding shared key is uniformly random. As a side-effect, in this hybrid also the right user’s public key will be switched to uniformly random. To argue this transition, we use the LWE assumption to argue

$$\left(\mathbf{A}, \mathbf{pk}_L, \left(\mathbf{A} \right) \mathbf{s}_R + \left(\mathbf{e}_L \right) \right) \approx_c \left(\mathbf{A}, \mathbf{pk}_L, \left(\mathbf{pk}_R \right) \right)$$

for $\mathbf{pk}_R \xleftarrow{\$} \mathbb{Z}_q^n, k \xleftarrow{\$} \mathbb{Z}_q$.

If the rounding function is designed to map uniformly random inputs to uniformly random outputs, the rounded shared is also uniformly random in the final hybrid, which finishes the proof.

²The noise terms e'_L, e'_R can be omitted in certain parameter settings. We keep them here for generality and simplicity.

To prove adaptive HKR security, where the adversary can create many (left and right) users and adaptively corrupt them (i.e. obtain their secret key)³, we have to be careful to apply the argument outlined above only to the public keys of the left and right challenge users (for whom the adversary retrieves a real-or-random shared key), because all other users can be corrupted by the adversary, and it is impossible to return a valid-looking secret key for users whose public key has been switched to uniformly random. This requires guessing both the left and the right challenge user ahead of time, leading to a security loss of $\mathcal{O}(N^2)$.

To achieve (adaptive) DKR security, the noise terms e'_L and e'_R have to be large enough to smudge the other noise terms ($\mathbf{s}_L^\top \mathbf{e}_R$ and $\mathbf{e}_L^\top \mathbf{s}_R$) and public keys have to be equipped with a NIZKPoK proving knowledge of the corresponding secret keys [Lan23]. This approach works in the same way for the tighter variant of this work, presented below.

Improving tightness. The key to get a tighter reduction is to use a “wide” matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with $m \approx n \log q$. With such a wide matrix, we can argue for a short vector $\mathbf{u}$ (e.g. $\mathbf{u} \in \{0, 1\}^m$) that $(\mathbf{A}, \mathbf{A}\mathbf{u}) \approx_s (\mathbf{A}, \mathbf{x})$ for $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^n$ with the left-over hash Lemma. This allows us to replace the computational argument we used above for the right user with a statistical argument. The advantage of the statistical argument is that it does not require guessing a user ahead of time, reducing the security loss to $\mathcal{O}(N)$. Additionally, this argument does not require the right user to have an error term in its public key, which in turn allows the left user to pick the secret $\mathbf{s}_L$ uniformly from $\mathbb{Z}_q^n$ (instead of using a short vector).

In more detail, we use a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ with $m \approx n \log q$ as global setup. Left public keys are of the form $\mathbf{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ and right public keys are of the form $\mathbf{pk}_R := \mathbf{A}\mathbf{u}_R$ where $\mathbf{e}_L, \mathbf{u}_R \in \mathbb{Z}_q^n$ are short vectors and $\mathbf{s}_L \xleftarrow{\$} \mathbb{Z}_q^n$ is uniformly random. The vectors $\mathbf{s}_L/\mathbf{u}_R$ are used as secret keys. As before, each user computes the shared key as the inner product of its own secret key and the public key of the other user, resulting in

$$\mathbf{s}_L^\top \cdot \mathbf{pk}_R = \mathbf{s}_L^\top \mathbf{A}\mathbf{u}_R \approx \mathbf{s}_L^\top \mathbf{A}\mathbf{u}_R + \mathbf{e}_L^\top \mathbf{u}_R + e'_R = \mathbf{pk}_L \cdot \mathbf{u}_R.$$

To prove security, we guess the left challenge user and switch, as before, to a hybrid argument where this user’s public key is uniformly random instead of having the format $\mathbf{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$. After this single change, we can already argue information-theoretically that the challenge pre-rounding shared key is uniformly random in the adversary’s view. Let $\mathbf{u}_R$ be the right challenge user’s secret key. Apart from the challenge shared key, the adversary learns about this secret key only $\mathbf{A}\mathbf{u}_R$, which leaves a significant amount of entropy in $\mathbf{u}_R$ (for suitable parameters). The challenge (pre-rounding) shared key has the form $\mathbf{pk}_L \mathbf{u}_R$, where $\mathbf{pk}_L$ is uniformly random and independent of $\mathbf{A}$. With the generalized left-over hash Lemma, we can argue $(\mathbf{A}, \mathbf{pk}_L, \mathbf{A}\mathbf{u}_R, \mathbf{pk}_L \mathbf{u}_R) \approx_s (\mathbf{A}, \mathbf{pk}_L, \mathbf{A}\mathbf{u}_R, k)$ for $k \xleftarrow{\$} \mathbb{Z}_q$.

Hash proof systems. From a bird’s eye view, our NIKE has a similar structure to the NIKE in [HHK18]. This is no coincidence. They give a generic transformation from a hash-proof system to a NIKE with $\mathcal{O}(N)$ security loss (and a group-based instantiation of this). The core of our NIKE construction can be seen as instantiating their transformation with a lattice-based hash proof system.

However, we cannot use their generic transformation as a black-box, since lattice-based hash proof systems have some limitations that make them unsuitable for their generic result and that require some lattice-specific tricks to handle. Concretely, the lattice-based hash proof system only provides approximate correctness. Thus, we present our results in this work without the abstraction of hash proof systems and elaborate more on the connection to hash proof systems in Appendix C for interested readers.

Polynomial modulus-to-noise ratio. Langrehr [Lan23] is so far the only work that studied the lattice-based NIKE with polynomial modulus-to-noise ratio. In this setting, the NIKE has a non-negligible correctness error and this is problematic for the proof strategy that we have highlighted above. Namely, the proof strategy that we have outlined here requires the reduction to choose for the left challenge user a public key that has no corresponding secret key. However, in the security game the adversary can ask for shared keys with right users (except for the right challenge user) that are computed using these secret keys. Moreover, it can also specify that the challenge shared key should be computed using the left user’s

³For the purpose of this overview, we ignore that the adversary can also reveal shared keys.

scheme	$\frac{\text{modulus}}{\text{noise}}$	security model	security loss	assumptions/models
folklore	super-poly.	adaptive HKR	$O(N^2 q_C)$	LWE
[Lan23, Section 3]	poly.	N-user adaptive HKR	$O(N^2 q_C)$	LWE
[Lan23, Appendix A.2]	super-poly.	adaptive DKR	$O(N^2 q_C)$	LWE + NIZK
SWOOSH [Gaj+24]	super-poly.	adaptive DKR	$O(N^2 q_C q_H^2)$	LWE + QROM + NIZK
Ours (Sec. B.3)	poly.	N-user adaptive HKR	$O(N)$	LWE
Ours (Sec. 4.1, B.2)	super-poly.	adaptive HKR	$O(N)$	LWE
Ours	super-poly.	adaptive DKR	$O(N)$	LWE + NIZK

Table 1: Overview of all variants of the lattice-based NIKE studied in this work and prior work. The second column shows whether the scheme is based on LWE with a polynomial or super-polynomial modulus-to-noise ratio. In the security loss column N refers to the number of registered users, q_C to the number of corrupted users, and q_H to the number of QROM queries. The scheme in the final row is obtained by combining our construction with super-polynomial modulus-to-noise ratio (Sections B.2 and 4.1) with the generic transformation of [Lan23, Appendix A.2].

secret key. If the correctness error is negligible, we can simply use the right user’s secret key in these cases instead.

In [Lan23] these problems are solved by adding in some cases “correction terms” to the shared keys (before rounding them), which negate the effect of using a different secret key than specified by the adversary has. These “correction terms” need in their work some leakage about the LWE secret and error, and they leverage the leakage resilience properties of the LWE assumption to justify this leakage.

We show in Section B.3 that this approach can be adapted to our NIKE as well. Moreover, we show that with the modified structure of the LWE-NIKE, we can get rid of the leakage about the LWE secret. Instead, we just need noisy linear leakage about the LWE error, which can be argued completely statistically. This has the advantage that we can prove security directly based on the standard LWE assumption without any leakage. This allows the usage of more efficient parameters when instantiating our scheme, because leakage about the LWE secret comes at the price of requiring a larger dimension n for a fixed concrete security level.

Our approach needs, like [Lan23], an a priori fixed bound on the number of users, to bound the leakage about the LWE error that we still need. But, as in [Lan23], our parameters scale well (in particular, sublinear) in the bound on the number of users. Essentially, they scale linear with the expected number of users that will have a correctness error with one fixed user.

Table 1 gives an overview of all variants of the lattice-based NIKE studied in this work and prior work.

2.1 Optimality of the tightness.

Finally, we show that our security loss of $\mathcal{O}(N)$ is optimal for lattice-based NIKE (at least, for a natural structure of lattice-based NIKE schemes that captures all schemes proposed so far). The literature already has several works that show lower bounds on the tightness of NIKE [Bad+16; HHK18; Hes+21]. However, none of these results apply to lattice-based NIKE. One reason is that all of these schemes only apply to *perfectly correct* NIKE, which none of the existing LWE candidates is (and it seems very challenging to construct such schemes [Guo+20]). Another reason is that all of them have some requirements on the NIKE that are not fulfilled by the known lattice-based NIKE constructions.

Metareductions. All the existing lower bounds (as well as the new one we propose) make use of the metareduction technique, introduced by Coron [Cor02]. For this technique, we assume that the reduction makes only black-box use of the adversary, (except for rewinding the adversary) and reduces the security notion, here adaptive HKR security of the NIKE, to any non-interactive assumption.

Such a reduction is then used to build an adversary $\mathcal{B}$ that uses the reduction to attack the underlying assumption. Therefore, we typically first formulate an inefficient adversary $\mathcal{A}$ for the cryptographic scheme (here, the NIKE), that has a high probability in winning the security game (for example, by brute-force

searching the secret key). Typically, this adversary has probability 1 in winning the game, but in our case it has a smaller success probability. Depending on the parameters for the LWE-NIKE, it can be for example overwhelming or a constant less than 1. The adversary $\mathcal{B}$ then internally runs the reduction and tries to efficiently simulate the behavior of $\mathcal{A}$. Assuming that $\mathcal{A}$ succeeds, that $\mathcal{B}$ successfully simulates $\mathcal{A}$, and that the reduction is successful, the adversary $\mathcal{B}$ breaks the underlying hardness assumption. By analyzing how successful $\mathcal{A}$ and $\mathcal{B}$ are and assuming that the underlying assumption is hard, this gives us a bound on the success probability and therefore tightness of the reduction.

Next, we briefly revisit each of these metareductions and explain why they cannot capture lattice-based schemes. For simplicity, we assume for this overview that the reduction does not rewind the adversary. However, all the results hold even for rewinding reductions.

Quadratic lower bound. The first metareduction for NIKE was published by Bader et al. in [Bad+16]. In their work, the inefficient adversary $\mathcal{A}$ first registers N users and performs a check on the public key for each of them. If all of these checks pass, it selects a random pair of challenge users and asks for the secret keys for all the other users. Each of these secret keys is then checked if it matches the public key.⁴ Finally, if all of these checks pass, $\mathcal{A}$ brute-force searches a secret key that matches the public key of the first challenge user, otherwise it outputs $\perp$. Of course, honestly generated key pairs should satisfy the checks performed on the public keys and the public/secret key pairs with high probability, to make sure that $\mathcal{A}$ has high success probability.⁵ Moreover, these checks need to be efficiently computable, because they are also executed by $\mathcal{B}$.

The efficient adversary $\mathcal{B}$ now runs the reduction internally and, like $\mathcal{A}$ it will also register N users initially and check the public keys that the reduction outputs. After that, it runs $N(N - 1)$ parallel copies of the reduction, one for each possible choice of the challenge user pair to retrieve the secret keys of the remaining users. It then selects one of these runs uniformly at random and tries to successfully complete only this single one. If the reduction failed to provide a valid secret key for one of the users in this run, $\mathcal{B}$ just outputs $\perp$ and then perfectly simulates the behavior of $\mathcal{A}$ in this case. Otherwise, instead of brute-force searching for a secret key for the challenge user, $\mathcal{B}$ inspects all the other runs of the reduction to see if the reduction has returned a valid secret key for *one of* the challenge users there. If this is the case, it uses this key to compute the shared key between the challenge users. Otherwise, it aborts. The probability of aborting can be shown to be in $\mathcal{O}(1/N^2)$, which implies that the reduction must have a security loss of $\Omega(N^2)$.

To argue that $\mathcal{B}$ correctly simulates $\mathcal{A}$ when it does not abort, we crucially need the following two properties

1. **Perfect correctness:** We need perfect correctness for all key pairs passing the checks of the reduction to argue that the reduction obtains the same shared key for the challenge users, even when it only obtained a secret key for the second challenge user (while $\mathcal{A}$ always uses the secret key for the first challenge user). Even if we allow only a negligible correctness error, this argument would fail, because the reduction could choose key pairs in such a way that they still pass the check, but make a correctness error much more likely. If $\mathcal{B}$ would only work if it received a secret key for the first challenge user, we would only get a linear lower bound.
2. **Unique shared keys:** Secondly, we need that if there are several secret keys that match the same public key (in the sense that they pass the checks done by $\mathcal{A}$ on the public/secret key pairs), they result in the same shared key when we compute the shared key with a valid public key (valid is again in the sense that this public key passes the check that $\mathcal{A}$ does on the public keys).

The lattice-based NIKE constructions in this work (and in the literature) violate both of these conditions. However, the main reason our reduction bypasses this impossibility result is the second one, because the reduction actually does not exploit the correctness error of the NIKE (in the sense that

⁴In their work, they do not mention this check on the secret keys explicitly, but it is clearly needed, because otherwise the reduction could just output nonsense as secret keys and then the metareduction would fail.

⁵In their work, they require this property to hold perfectly, but it can be relaxed similar to how we handle a non-perfect adversary $\mathcal{A}$ to “holds with overwhelming probability” without significantly impacting the result and even with high, but non-overwhelming, probability we get an interesting result as long as the overall advantage of $\mathcal{A}$ stays high.

it would output shared keys that cause correctness errors with higher probability).⁶ The reason our reduction violates the second condition is that right users have many possible secret keys for each public key, and they do lead to significantly⁷ different shared keys when the left user public key is far away from the lattice, and we cannot detect if the public key is close or far from the lattice given just the public key.⁸

Linear lower bound. Hesse, Hofheinz and Kohl [HHK18] presented a metareduction where there is no need to check public keys standalone. Instead, they only need a check for public/secret key pairs. They have a similar requirement for the NIKE as the second requirement above, except that this now only needs to hold if for both users there is a secret key that matches the public key. This requirement is met by their group hash proof system based NIKE.

For the adversary $\mathcal{B}$, it is now important that it obtains the secret key for *both* challenge users to ensure that it still simulates $\mathcal{A}$ correctly. This happens with probability $\mathcal{O}(1/N)$ and thus they get lower bound of $\mathcal{O}(N)$ for the security loss of such NIKES.

They also mention explicitly perfect correctness as requirement for their metareduction. However, one can show that the same result holds when dropping the perfect correctness requirement (essentially because $\mathcal{B}$ now anyway needs to get the secret keys for *both* challenge users).

However, even when we drop perfect correctness, we cannot apply this result to lattice-based NIKE. The reason is that the noise terms in the lattice-based NIKE schemes violate also the weaker unique shared key property that is needed here.

Our new linear lower bound. In this work we show a new metareduction that follows the high-level strategy of [HHK18], but needs only an even weaker variant of the “unique shared key” property that we call “pairwise checkable”. We then give a general structure of lattice-based NIKE schemes, that captures all constructions of this work and previous works for lattice-based NIKES, and show they satisfy the requirements of this metareduction.

Concretely, we replace the validity check for the key pair of a single user with a validity check for each *pair* of users. This check gets as input the public-secret key pair of one user (say Alice) and the public key (but crucially not the secret key!) of another user (say Bob). We now need that when there are two secret keys for Alice, that both pass this check for Alice’s and Bob’s public key and there is a secret key for Bob, that passes this check for Bob’s and Alice’s public key, then both of Alice’s secret keys yield the same shared key for the pair Alice and Bob.

This property is satisfied by the lattice-based NIKE, because the pairwise check can (apart from doing validity checks on the individual key pairs) additionally compute the pre-rounding shared key between Alice and Bob. If this shared key is close to a rounding boundary, the check will fail. If the shared key is sufficiently far away from the rounding boundaries, the check returns true. This gives us robustness against small noise terms, because they can influence the pre-rounding shared key only by a small amount.

In our analysis, we now have to take into account that even honestly generated keys can cause this check to fail. The result we get here is that the reduction needs to have a security loss of at least $\Omega(N(1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda)))$, where $\varepsilon_{\text{PWcheck}}(\lambda)$ is the probability that a pair of honestly generated keys fails the pairwise check. When we select parameters such that $\varepsilon_{\text{PWcheck}}(\lambda) < 1/(2N^2)$, which is only a mild requirement on the parameters that can be achieved even with polynomial modulus-to-noise ratio, this simplifies the lower bound to $\Omega(N)$.

Lower bound for inner-product NIKE. Finally, the work [Hes+21] shows a lower bound that does not require a form of unique shared keys as the metareductions discussed so far. Instead, they require the NIKE to have a certain inner-product structure, where public and secret keys contain (possibly hidden)

⁶In fact, exploiting the correctness error in such a way in the reduction seems not possible for the lattice-based NIKE, because the adversary could detect that the pre-rounding shared keys are more often close to a rounding boundary than expected and stop working.

⁷Significant means here that these differences can be much larger than the noise terms and will likely not disappear in the rounding function.

⁸For the previous lattice-based NIKES where $\mathbf{A}$ is a square matrix, the argument is very similar. When the lattice described by $\mathbf{A}$ has unusually short vectors, there can be many secret keys for one public key. The reduction we sketched for these schemes exploits exactly this: By setting up $\mathbf{A}$ in the “lossy mode” it ensures that such short vectors exist without the adversary being able to detect this.

vectors and the shared key is an *injective* function applied to the inner product of public and secret key vector of the two users. They show that such a NIKE has a security loss of $\Omega(N/d)$, where d is the dimension of the vectors for the inner-product structure. Additionally, they require perfect correctness, but we observe this requirement can be dropped with a similar argument as for the bound by [HHK18].

However, as noted already in [Hes+21], the known lattice-based NIKE schemes do not have the inner-product structure required for their result, because the rounding function is not injective. Moreover, in the scheme with polynomial modulus-to-noise ratio in our work (and similarly in [Lan23]) a pseudorandom noise term is added to the shared key before rounding, which also violates the inner-product structure.

Publication information. The part on lower bounds of this work is based on the master thesis of the second author [Tra25]. This thesis contains formal proofs for the variants of the existing metareductions that we sketched in the technical overview (in particular, the formal properties needed for the [Bad+16] metareduction and variants of the metareductions [HHK18] and [Hes+21] that do not require perfect correctness and support rewinding reductions).

3 Preliminaries⁹

We use $\mathbb{N}_0$ for the set of natural numbers with zero and $\mathbb{N}_+$ for the set of natural number without zero, and $[n]$ (with $n \in \mathbb{N}_+$) for the set $\{1, \dots, n\}$.

We use $\text{Bits} : \mathbb{Z}_{2^\rho} \rightarrow \{0, 1\}^\rho$ to denote the function mapping integers in $\mathbb{Z}_{2^\rho}$ to their binary representation.

For a probability distribution $\mathcal{D}$, we write $X \leftarrow \mathcal{D}$ to denote a random variable X that is distributed according to $\mathcal{D}$ (and independent of all other random variables). For a (probabilistic) algorithm $\mathcal{A}$ we write $X \leftarrow \mathcal{A}(b)$ to denote the random variable X output by $\mathcal{A}$ on input b . For a set S we use $X \xleftarrow{\$} S$ as shortcut for $X \leftarrow \mathcal{U}(S)$, where $\mathcal{U}(S)$ is the uniform distribution on S . We use $\mathsf{T}(\mathcal{A})$ to denote the runtime of $\mathcal{A}$.

Definition 3.1 (Statistical distance). The statistical distance between two random variables X and Y is defined as

$$\text{SD}(X, Y) := \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]|.$$

3.1 Learning with errors

We recall the Learning with errors assumption introduced in [Reg05].

Definition 3.2 (LWE assumption). The (n, m, q, σ) -LWE assumption for $n, m, q \in \mathbb{N}_+$ and $\sigma \in \mathbb{R}_+$ (that all implicitly depend on the security parameter), states that for every PPT adversary $\mathcal{A}$,

$$\text{Adv}_{n,m,q,\sigma}^{\text{lwe}}(\mathcal{A}) := \Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top) - \mathcal{A}(\mathbf{A}, \mathbf{y})]$$

is negligible when the probability is taken over $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^{1 \times m}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma}$, and the internal randomness of $\mathcal{A}$.

Here, $\mathcal{D}_{\mathbb{Z}^m, \sigma}$ is the spherical discrete Gaussian distribution on $\mathbb{Z}^m$ with Gaussian parameter σ . We refer the reader to Section A.5 for more details.

The discrete Gaussian distribution satisfies the following tail bound.

Lemma 3.3 ([Lyu12, Lemma 4.4 (3)]). For any $k > 1, \sigma > 0$, and $n \in \mathbb{N}_+$

$$\Pr_{\mathbf{z} \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma}} [\|\mathbf{z}\|_2 > k\sigma\sqrt{n/2\pi}] < k^n e^{n(1-k^2)/2}.$$

⁹Some standard definitions in this section are taken verbatim from [Lan23].

3.2 Non-interactive key exchange

The notion of NIKE has been formalized in [CKS08]. This notion associates each user with a unique identity, which is necessary for security notions that allow the adversary to introduce maliciously generated public keys. For the security notions considered in this work the identities are not needed (and can be easily removed from our construction and impossibility results).

For convenience, we use for our constructions the notion of asymmetric non-interactive key exchange that has been used in [Lan23] to describe the LWE-based NIKE. This notion can be easily converted to the regular NIKE notion by generating a left and right key pair for every user and deciding based on some fixed rule for each pair of users who should use their left key and who should use their right key.

Definition 3.4 (NIKE). An asymmetric *NIKE* scheme with identity space $\mathcal{IDS}$ and shared key space $\mathcal{K}$ (with $|\mathcal{IDS}|, |\mathcal{K}| \geq 2$) consists of the following five PPT algorithms:

- **Setup** inputs the unary encoded security parameter 1^λ and samples public parameters $\mathbf{pp}$,
- **KeyGenL** inputs the parameters $\mathbf{pp}$ and an identity $\text{id}_L \in \mathcal{IDS}$ and samples a left key pair $(\mathbf{pk}_L, \mathbf{sk}_L) \in \mathcal{LPK} \times \mathcal{LSK}$,
- **KeyGenR** inputs the parameters $\mathbf{pp}$ and an identity $\text{id}_R \in \mathcal{IDS}$ and samples a right key pair $(\mathbf{pk}_R, \mathbf{sk}_R) \in \mathcal{RPK} \times \mathcal{RSK}$,
- **SharedKeyL** inputs the parameters $\mathbf{pp}$, an identity id_R with its corresponding right public key $\mathbf{pk}_R$ and another identity id_L with its corresponding left secret key $\mathbf{sk}_L$ and outputs a shared key K or the failure symbol $\perp$.
- **SharedKeyR** inputs the parameters $\mathbf{pp}$, an identity id_L with its corresponding left public key $\mathbf{pk}_L$ and another identity id_R with its corresponding right secret key $\mathbf{sk}_R$ and outputs a shared key K or the failure symbol $\perp$.

The standard definition of NIKE is obtained by requiring $\text{KeyGenL} = \text{KeyGenR}$ (which is then just called KeyGen for simplicity) and $\text{SharedKeyL} = \text{SharedKeyR}$ (which is then just called SharedKey for simplicity).

Definition 3.5 (Correctness). We say that a NIKE $\text{NIKE} = (\text{Setup}, \text{KeyGenL}, \text{KeyGenR}, \text{SharedKeyL}, \text{SharedKeyR})$ for identity space $\mathcal{IDS}$ has correctness error $\varepsilon_{\text{corr}}$, if for all $\lambda \in \mathbb{N}_+$ and all $\text{id}_L, \text{id}_R \in \mathcal{IDS}$ with $\text{id}_L \neq \text{id}_R$, for

- $\mathbf{pp} \leftarrow \text{Setup}(1^\lambda)$,
- $(\mathbf{pk}_L, \mathbf{sk}_L) \leftarrow \text{KeyGenL}(\mathbf{pp}, \text{id}_L)$,
- $(\mathbf{pk}_R, \mathbf{sk}_R) \leftarrow \text{KeyGenR}(\mathbf{pp}, \text{id}_R)$,
- $K_L \leftarrow \text{SharedKeyL}(\mathbf{pp}, \text{id}_R, \mathbf{pk}_R, \text{id}_L, \mathbf{sk}_L)$, and
- $K_R \leftarrow \text{SharedKeyR}(\mathbf{pp}, \text{id}_L, \mathbf{pk}_L, \text{id}_R, \mathbf{sk}_R)$

it holds

$$\Pr[K_L = K_R \neq \perp] \geq 1 - \varepsilon_{\text{corr}}(\lambda),$$

where the probability is taken over the randomness used to sample $\mathbf{pp}$, $(\mathbf{pk}_L, \mathbf{sk}_L)$, $(\mathbf{pk}_R, \mathbf{sk}_R)$, K_L and K_R .

A NIKE is *statistically correct* if $\varepsilon_{\text{corr}}$ is negligible and *perfectly correct* iff $\varepsilon_{\text{corr}} = 0$.

Definition 3.6 (Adaptive HKR security). An asymmetric NIKE NIKE is $((N_L, N_R)\text{-user})$ adaptively HKR secure if for all PPT adversaries $\mathcal{A}$

$$\text{Adv}_{\mathcal{A}, \text{NIKE}}^{\text{xxx}}(\lambda) := 2|\Pr[\text{Exp}_{\mathcal{A}, \text{NIKE}}^{\text{xxx}}(\lambda) \Rightarrow 1] - 1/2|$$

is negligible for $\text{xxx} = \text{N-user-adaptive-HKR}$ or $\text{xxx} = \text{adaptive-HKR}$, respectively. The games $\text{Exp}_{\mathcal{A}, \text{NIKE}}^{\text{xxx}}$ are defined in Figure 1 and Table 2.

For regular NIKE, $(N\text{-user})$ adaptive HKR security is defined similarly, except that there is no distinction between $\mathcal{O}_{\text{regHL}}/\mathcal{O}_{\text{regHR}}$, $\mathcal{O}_{\text{extrL}}/\mathcal{O}_{\text{extrR}}$, and $\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ queries. The transformation from asymmetric to regular NIKE preserves correctness and tightly translates $((N, N)\text{-user})$ adaptive HKR security to $(N\text{-user})$ adaptive HKR security (defined below).

We recall the definitions for (conditional) Rényi entropy and some of its basic rules, the standard definitions of PRFs, and the background for the module LWE assumption in Appendix A.

$\text{Exp}_{\mathcal{A}, \text{NIKE}}^{(\text{N-user-})\text{adaptive-HKR}}(\lambda):$ $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ $Q_{\text{extrL}} := \emptyset; Q_{\text{extrR}} := \emptyset$ $Q_{\text{rev}} := \emptyset; Q_{\text{test}} := \emptyset$ $\text{lpks} : \mathcal{IDS} \dashrightarrow \mathcal{LPK}$ $\text{rpks} : \mathcal{IDS} \dashrightarrow \mathcal{RPK}$ $\text{lsks} : \mathcal{IDS} \dashrightarrow \mathcal{LSK}$ $\text{rsks} : \mathcal{IDS} \dashrightarrow \mathcal{RSK}$ $b \xleftarrow{\\$} \{0, 1\}$ $b^* \leftarrow \mathcal{A}(\text{pp})$ if $Q_{\text{rev}} \cap Q_{\text{test}} = \emptyset \wedge \forall (\text{id}_1, \text{id}_2) \in Q_{\text{test}} : \text{id}_1 \notin Q_{\text{extrL}} \wedge \text{id}_2 \notin Q_{\text{extrR}}$ then return $b \stackrel{?}{=} b^*$ else return $b' \xleftarrow{\\$} \{0, 1\}$ $\mathcal{O}_{\text{regHL}}(\text{id}_L \in \mathcal{IDS}):$ if $\text{lpks}(\text{id}_L) \neq \perp$ then return $\perp$ $(\text{pk}_L, \text{sk}_L) \leftarrow \text{KeyGenL}(\text{pp}, \text{id}_L)$ $\text{lpks}(\text{id}_L) := \text{pk}_L; \text{lsks}(\text{id}_L) := \text{sk}_L$ return pk_L $\mathcal{O}_{\text{regHR}}(\text{id}_R \in \mathcal{IDS}):$ if $\text{rpks}(\text{id}_R) \neq \perp$ then return $\perp$ $(\text{pk}_R, \text{sk}_R) \leftarrow \text{KeyGenR}(\text{pp}, \text{id}_R)$ $\text{rpks}(\text{id}_R) := \text{pk}_R; \text{rsks}(\text{id}_R) := \text{sk}_R$ return pk_R $\mathcal{O}_{\text{extrL}}(\text{id}_L \in \mathcal{IDS}):$ if $\text{lsks}(\text{id}_L) \neq \perp$ then $Q_{\text{extrL}} := Q_{\text{extrL}} \cup \{\text{id}_L\}$ return $\text{lsks}(\text{id}_L)$ else return $\perp$ $\mathcal{O}_{\text{extrR}}(\text{id}_R \in \mathcal{IDS}):$ if $\text{rsks}(\text{id}_R) \neq \perp$ then $Q_{\text{extrR}} := Q_{\text{extrR}} \cup \{\text{id}_R\}$ return $\text{rsks}(\text{id}_R)$ else return $\perp$	$\mathcal{O}_{\text{revL}}(\text{id}_R \in \mathcal{IDS}, \text{id}_L \in \mathcal{IDS}):$ if $\text{rpks}(\text{id}_R) \neq \perp \wedge \text{lsks}(\text{id}_L) \neq \perp$ then $Q_{\text{rev}} := Q_{\text{rev}} \cup \{(\text{id}_L, \text{id}_R)\}$ $\text{pk}_R := \text{rpks}(\text{id}_R); \text{sk}_L := \text{lsks}(\text{id}_L)$ ret. $\text{SharedKeyL}(\text{pp}, \text{id}_R, \text{pk}_R, \text{id}_L, \text{sk}_L)$ else return $\perp$ $\mathcal{O}_{\text{revR}}(\text{id}_L \in \mathcal{IDS}, \text{id}_R \in \mathcal{IDS}):$ if $\text{lpks}(\text{id}_L) \neq \perp \wedge \text{rsks}(\text{id}_R) \neq \perp$ then $Q_{\text{rev}} := Q_{\text{rev}} \cup \{(\text{id}_L, \text{id}_R)\}$ $\text{pk}_L := \text{lpks}(\text{id}_L); \text{sk}_R := \text{rsks}(\text{id}_R)$ ret. $\text{SharedKeyR}(\text{pp}, \text{id}_L, \text{pk}_L, \text{id}_R, \text{sk}_R)$ else return $\perp$ $\mathcal{O}_{\text{testL}}(\text{id}_R^* \in \mathcal{IDS}, \text{id}_L^* \in \mathcal{IDS}):$ if $\text{rsks}(\text{id}_R^*) \neq \perp \wedge \text{lsks}(\text{id}_L^*) \neq \perp$ then $Q_{\text{test}} := Q_{\text{test}} \cup \{(\text{id}_L^*, \text{id}_R^*)\}$ $\text{pk}_R^* := \text{rpks}(\text{id}_R^*); \text{sk}_L^* := \text{lsks}(\text{id}_L^*)$ $K_0^* \leftarrow \text{SharedKeyL}(\text{pp}, \text{id}_R^*, \text{pk}_R^*, \text{id}_L^*, \text{sk}_L^*)$ $K_1^* \xleftarrow{\\$} \mathcal{K}$ return K_b^* else return $\perp$ $\mathcal{O}_{\text{testR}}(\text{id}_L^* \in \mathcal{IDS}, \text{id}_R^* \in \mathcal{IDS}):$ if $\text{lsks}(\text{id}_L^*) \neq \perp \wedge \text{rsks}(\text{id}_R^*) \neq \perp$ then $Q_{\text{test}} := Q_{\text{test}} \cup \{(\text{id}_L^*, \text{id}_R^*)\}$ $\text{pk}_L^* := \text{lpks}(\text{id}_L^*); \text{sk}_R^* := \text{rsks}(\text{id}_R^*)$ $K_0^* \leftarrow \text{SharedKeyR}(\text{pp}, \text{id}_L^*, \text{pk}_L^*, \text{id}_R^*, \text{sk}_R^*)$ $K_1^* \xleftarrow{\\$} \mathcal{K}$ return K_b^* else return $\perp$
--	---

Figure 1: Experiment for (N_L, N_R) -user adaptive HKR security of an asymmetric NIKE scheme $\text{NIKE} = (\text{Setup}, \text{KeyGenL}, \text{KeyGenR}, \text{SharedKeyL}, \text{SharedKeyR})$ with identity space $\mathcal{IDS}$. $\mathcal{LPK}$ and $\mathcal{RPK}$ denote the left and right public key spaces, and $\mathcal{LSK}$ and $\mathcal{RSK}$ denote the left and right secret key spaces, respectively. The adversary $\mathcal{A}$ has access to the oracles as indicated in Table 2. The partial maps lpks , rpks , lsks , and rsks are initially totally undefined.

	$\mathcal{O}_{\text{regHL}}$	$\mathcal{O}_{\text{regHR}}$	$\mathcal{O}_{\text{extrL}}$	$\mathcal{O}_{\text{extrR}}$	$\mathcal{O}_{\text{revL}}$	$\mathcal{O}_{\text{revR}}$	$\mathcal{O}_{\text{testL}}$	$\mathcal{O}_{\text{testR}}$
(N_L, N_R) -user adapt. HKR	N_L	N_R	∞	∞	∞	∞	1	
adaptive HKR	∞	∞	∞	∞	∞	∞	1	

Table 2: The following table indicates how often the adversary is allowed to query each of the oracles in the (N_L, N_R) -user adaptive HKR security game. The symbol “ ∞ ” stands for arbitrary many allowed queries. The last column shows the bound on the number of $\mathcal{O}_{\text{testL}}$ plus $\mathcal{O}_{\text{testR}}$ queries.

Setup (1^λ): $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ return $\text{pp} := \mathbf{A}$ KeyGenL ($\text{pp} = \mathbf{A}, \text{id}_L$): $\text{sk}_L \xleftarrow{\$} \mathbb{Z}_q^n$; $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma}$ $\text{pk}_L := \text{sk}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ return $(\text{pk}_L, \text{sk}_L)$ KeyGenR ($\text{pp} = \mathbf{A}, \text{id}_R$): $\text{sk}_R \xleftarrow{\$} \{0, 1\}^m$ $\text{pk}_R := \mathbf{A} \text{sk}_R$ return $(\text{pk}_R, \text{sk}_R)$	SharedKeyL ($\text{pp} = \mathbf{A}, \text{id}_R, \text{pk}_R, \text{id}_L, \text{sk}_L$): return $\text{Round}(\text{sk}_L^\top \text{pk}_R)$ SharedKeyR ($\text{pp} = \mathbf{A}, \text{id}_L, \text{pk}_L, \text{id}_R, \text{sk}_R$): return $\text{Round}(\text{pk}_L \text{sk}_R)$ Round : $\mathbb{Z}_q \rightarrow \{0, 1\}$ $x \mapsto \begin{cases} 0 & \text{if } x \in [0, \dots, \frac{q-1}{2}] \\ 1 & \text{otherwise} \end{cases}$
--	---

Figure 2: The new, simplified LWE based NIKE $\text{NIKE}_{\text{sp}}^{\text{simple}}$ we consider in this work in the variant for super-polynomial modulus-to-noise ratio.

4 A simple, tighter LWE-based NIKE.

In this section, we present a simplified version of our tight lattice-based NIKE construction. The construction we present here only works for unstructured LWE with super-polynomial modulus-to-noise ratio and produces only a single-bit shared key.

In Appendix B we generalize this construction to ring and module LWE and present it with flexible shared key length. The main difference with structured LWE is that we need to rely on a ring version of the left-over hash lemma. In this section we also present a variant of that NIKE that can handle polynomial modulus-to-noise ratio. The main challenge for polynomial modulus-to-noise ratio is that we need to adapt the proof strategy, because this NIKE has non-negligible correctness error.

4.1 Warm-up: NIKE for super-polynomial modulus-to-noise ratio

Theorem 4.1 (Correctness). *The NIKE scheme presented in Figure 2 has correctness error*

$$\varepsilon_{\text{corr}}(\lambda) \leq 4m\sigma\sqrt{m/\pi}/q + \sqrt{2}^m e^{-m/2} + 2^{-\lambda/2-1} + 1/q^n$$

if $m \geq (n+1)\log(q) + \lambda$ and q is a prime.

Proof. Let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{s}_L \xleftarrow{\$} \mathbb{Z}_q^n$; $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma}$, and $\text{sk}_R \xleftarrow{\$} \{0, 1\}^m$. When a user with left public key $\mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ and a user with right public key $\text{pk}_R := \mathbf{A} \text{sk}_R$ compute the shared key between them, the unrounded keys are

$$k_L := \mathbf{s}_L^\top \mathbf{A} \text{sk}_R \quad \text{and} \quad k_R := (\mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top) \text{sk}_R = \mathbf{s}_L^\top \mathbf{A} \text{sk}_R + \mathbf{e}_L^\top \text{sk}_R$$

and thus the ∞ -norm of their difference is

$$\|k_L - k_R\|_\infty = \|\mathbf{e}_L^\top \text{sk}_R\|_\infty \leq m \cdot \|\mathbf{e}_L\|_\infty \cdot \|\text{sk}_R\|_\infty \stackrel{(*)}{\leq} m\sigma\sqrt{m/\pi} =: B_K,$$

where the inequality marked with $(*)$ holds except with probability at most $\sqrt{2}^m e^{-m/2}$ (which is negligible in λ) by applying Lemma 3.3 with $k = \sqrt{2}$ to bound $\|\mathbf{e}_L\|_2 \geq \|\mathbf{e}_L\|_\infty$.

Next, we argue that the term $\mathbf{s}_L^\top \mathbf{A} \text{sk}_R$ (and therefore also k_L) is close to uniformly random. First, note that the vector $\mathbf{A} \text{sk}_R$ is within statistical distance $2^{-\lambda/2-1}$ to a uniformly random vector (using that $m \geq (n+1)\log(q) + \lambda$, see Section 4.2). We also have $\Pr[\mathbf{s} = \mathbf{0}] = 1/q^n$. The inner product of a fixed non-zero vector $\mathbf{s}$ and a uniformly random vector is a uniformly random scalar in $\mathbb{Z}_q$. Thus, we get the statistical distance of $\mathbf{s}_L^\top \mathbf{A} \text{sk}_R$ to a uniformly random scalar is at most $2^{-\lambda/2-1} + 1/q^n$.

Assume k_L is uniformly random in $\mathbb{Z}_q$. Then, k_L will be within distance B_K of 0 or $(q-1)/2$ (the “rounding boundaries”) with probability $4B_K/q$. If this does not occur, we have $\text{Round}(k_L) = \text{Round}(k_R)$.

Using that $\mathbf{s}_L^\top \mathbf{Ask}_R$ and therefore k_L is statistically close to uniformly random, as shown above, we get the desired correctness bound. $\square$

Theorem 4.2 (Security). *The NIKE NIKE_{sp} in Figure 2 is adaptive HKR secure if the following requirements for the parameters are satisfied.*

- q is a prime.
- The (n, m, σ) -LWE assumption holds.
- σ/q is negligible (which implies that $\varepsilon_{\text{corr}}$ is negligible).
- $m \geq (n+1)\log(q) + \lambda$.

More precisely, for every PPT adversary $\mathcal{A}$ against the adaptive HKR security of NIKE there exists a PPT adversary $\mathcal{B}$ (with approximately the same run time) such that

$$\text{Adv}_{\mathcal{A}, \text{NIKE}}^{\text{adaptive-HKR}}(\lambda) \leq 2N_L \left(\text{Adv}_{n,m,q,\sigma}^{\text{lwe}}(\mathcal{B}) + N_R 2\varepsilon_{\text{corr}} + 2^{-\lambda/2-1} + 1/(2q) \right)$$

where N_L and N_R are the maximum number of left or right users the adversary registers, respectively.

Proof. The proof proceeds via a hybrid argument. The real adaptive HKR game is G_0 .

The first hybrid game G_1 changes all $\mathcal{O}_{\text{revR}}$ queries and the $\mathcal{O}_{\text{testR}}$ query. These queries are answered in G_1 like an $\mathcal{O}_{\text{revL}}$ or $\mathcal{O}_{\text{testL}}$ query (on the same arguments), respectively.

Lemma 4.3 ($G_0 \rightsquigarrow G_1$).

$$|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]| \leq N_L N_R \varepsilon_{\text{corr}}$$

Proof. The statistical difference between the answer of an $\mathcal{O}_{\text{revR}}(\text{id}_L, \text{id}_R)$ (or $\mathcal{O}_{\text{testR}}(\text{id}_L, \text{id}_R)$) query and an $\mathcal{O}_{\text{revL}}(\text{id}_R, \text{id}_L)$ query (or $\mathcal{O}_{\text{testL}}(\text{id}_R, \text{id}_L)$ query) is bounded by $\varepsilon_{\text{corr}}$. Since there are at most $N_L N_R - 1$ different $\mathcal{O}_{\text{revR}}$ queries and at most one $\mathcal{O}_{\text{testR}}$ queries, the Lemma follows. $\square$

The next hybrid G_2 is identical to G_1 , with the following change: The game samples a random index $i^* \xleftarrow{\$} [N_L]$. If the left user registered in the i^* -th $\mathcal{O}_{\text{regHL}}$ query is the left user used in the $\mathcal{O}_{\text{testR}}$ query, the game proceeds as in G_0 . Otherwise, the game aborts¹⁰ and the adversary wins with probability $1/2$.

Lemma 4.4 ($G_1 \rightsquigarrow G_2$).

$$\left| \Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2} \right| = N_L \left| \Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2} \right|$$

Proof. The index i^* chosen in the game $\Pr[G_2^{\mathcal{A}} \Rightarrow 1]$ is completely independent of the adversary's view. Thus, the probability that the adversary successfully distinguishes a real from a random shared key in the $\mathcal{O}_{\text{testR}}$ query is independent of i^* , and the probability that i^* is the registration index of the left user used for the $\mathcal{O}_{\text{testR}}$ query is $1/N_L$. $\square$

The next hybrid G_3 is identical to G_2 , except that all $\mathcal{O}_{\text{revL}}$ and $\mathcal{O}_{\text{revR}}$ queries as well as the $\mathcal{O}_{\text{testL}}$ or $\mathcal{O}_{\text{testR}}$ query for the left user with index i^* are answered using the right user's secret key, i.e like an $\mathcal{O}_{\text{revR}}$ or $\mathcal{O}_{\text{testR}}$ query in the initial game G_0 .

Lemma 4.5 ($G_2 \rightsquigarrow G_3$).

$$|\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1]| \leq N_R \varepsilon_{\text{corr}}$$

Proof. The statistical difference between the answer of an $\mathcal{O}_{\text{revL}}(\text{id}_R, \text{id}_L)$ query and an $\mathcal{O}_{\text{revR}}(\text{id}_L, \text{id}_R)$ query is bounded by $\varepsilon_{\text{corr}}$. Since there are at most N_R different¹¹ $\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}/\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ queries that are changed for G_3 , the lemma follows. $\square$

The next hybrid game G_4 generates the public key for the i^* -th left user as $\text{pk}_L \xleftarrow{\$} \mathbb{Z}_q^{1 \times m}$ (instead of $\text{pk}_L := \text{sk}_L^\top \mathbf{A} + \mathbf{e}_L^\top$). This means of course that there might be no corresponding secret key for pk_L , but this does not matter because in G_3/G_4 the secret key for the i^* -th challenge user is never used anyway (if the adversary makes an $\mathcal{O}_{\text{extrL}}$ query for this user the game aborts and if the adversary makes an $\mathcal{O}_{\text{revL}}$ query for this user, this is answered like an $\mathcal{O}_{\text{revR}}$ query).

¹⁰We assume that the game aborts if $\mathcal{A}$ makes an $\mathcal{O}_{\text{testR}}$ query with a left user with a different index than i^* or the adversary makes an $\mathcal{O}_{\text{extrL}}$ query for the left user with index i^* .

¹¹We count here an $\mathcal{O}_{\text{revL}}$ and $\mathcal{O}_{\text{revR}}$ query for the same pair of users as one query, because these queries are answered identically in both games.

Lemma 4.6 ($G_3 \rightsquigarrow G_4$). *For every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ with*

$$|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{n,m,q,\sigma}^{\text{lwe}}(\mathcal{B})$$

Proof. Since the secret key of the i^* -th left user is never used ($\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}$ queries never use this key and the game aborts if an $\mathcal{O}_{\text{extrL}}(\text{id}_L^*)$ query is made), the reduction is straightforward: The reduction receives the LWE challenge $(\mathbf{A}, \text{pk}_L^*)$ and uses $\mathbf{A}$ as public parameters and pk_L^* as the i^* -th left user's public key. The reduction generates all remaining parts of the game honestly. When $\text{pk}_L^* = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$, the reduction is simulating G_3 . When $\text{pk}_L^* \xleftarrow{\$} \mathbb{Z}_q^{1 \times m}$, the reduction is simulating G_4 . $\square$

In the final hybrid G_5 , the $\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ query is answered for $b = 0$ by sampling $k^* \xleftarrow{\$} \mathbb{Z}_q$ (instead of $k^* := \text{pk}_L^* \text{sk}_R^*$) and returning $\text{Round}(k^*)$.

Lemma 4.7 ($G_4 \rightsquigarrow G_5$).

$$|\Pr[G_4^{\mathcal{A}} \Rightarrow 1] - \Pr[G_5^{\mathcal{A}} \Rightarrow 1]| \leq 2^{-\lambda/2-1}$$

Proof. We assume the adversary makes an $\mathcal{O}_{\text{testL}}(\text{id}_R, \text{id}_L)$ or $\mathcal{O}_{\text{testR}}(\text{id}_L, \text{id}_R)$ query where id_L is the identity of the i^* -th challenge user. Otherwise, both games G_4 and G_5 abort and the lemma holds trivially in this case.

Let sk_R^* be the right challenge user's secret key. This secret key is not used in any $\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}$ query, because these queries are answered in G_4/G_5 using the left user's secret key unless the left user is id_L^* . We assume that the latter case does not occur (and that the adversary does not make an $\mathcal{O}_{\text{extrR}}(\text{id}_R^*)$ query), because then the adversary would lose the game anyway. Thus, the adversary's view can be simulated by using only $\text{pk}_R^* := \mathbf{A} \text{sk}_R^*$ and $\text{pk}_L^* \text{sk}_R^*$. The left-over hash Lemma thus shows that $\text{pk}_L^* \text{sk}_R^*$ is statistically close to uniformly random (see Section 4.2). $\square$

Lemma 4.8

$$\Pr[G_5^{\mathcal{A}} \Rightarrow 1] \leq \frac{1}{2} + \frac{1}{2q}$$

Proof. Since in G_5 the unrounded challenge shared key k^* is uniformly random in $\mathbb{Z}_q$, by Lemma B.1 the view of the adversary has statistical distance at most $1/q$ to a distribution that is statistically independent of the challenge bit b . This is because in G_5 the challenge shared key is $\text{Round}(k^*)$. The probability that the rounding function Round (for uniformly random input) outputs 0 is slightly higher (by $1/q$) than for outputting 1. Thus, the adversary has advantage at most $1/(2q)$ in guessing b . $\square$

Combining Lemmata 4.3–4.8 yields Theorem 4.2. $\square$

4.2 Left-over hash Lemma.

The correctness proof and the last step of our tighter proof require arguing that $(\mathbf{A}, \mathbf{Au})$ for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \leftarrow \{0, 1\}^m$ is statistically close to a uniformly random value from $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$, even given some linear leakage about $\mathbf{u}$.

For the integer case, we can argue this with the standard left-over hash lemma if q is a prime. Therefore, we recall the definition of universal hash functions and the generalized left-over hash lemma.

Definition 4.9 (Universal hash function). A distribution $\mathcal{H}$ of functions $h : \mathcal{X} \rightarrow \mathcal{Y}$ is called a universal hash family, iff for all $x_1 \neq x_2 \in \mathcal{X}$ we have $\Pr_{h \xleftarrow{\$} \mathcal{H}}[h(x_1) = h(x_2)] \leq 1/|\mathcal{Y}|$.

Lemma 4.10 (Generalized leftover hash Lemma (GLOHL), [DRS04, Lemma 2.4 in the ePrint version]). *For a random variable X over $\mathcal{X}$, a random variable Z (that can be correlated with X), a function $h : \mathcal{X} \rightarrow \mathcal{Y}$ sampled from a universal hash family and $Y \xleftarrow{\$} \mathcal{Y}$,*

$$\text{SD}((h, h(X), Z), (h, Y, Z)) \leq \frac{1}{2} \sqrt{2^{-H_\infty(X|Z)} |\mathcal{Y}|}$$

holds.

Corollary 4.11 *For any prime q , any $k, n \in \mathbb{N}_+$, any $\mathbf{X} \in \mathbb{Z}_q^{k \times m}$ and $m \geq (n + k) \log(q) + \lambda$ we have for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \xleftarrow{\$} \{0, 1\}^m$ and $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^n$*

$$\text{SD}((\mathbf{A}, \mathbf{Au}, \mathbf{Xu}), (\mathbf{A}, \mathbf{x}, \mathbf{Xu})) \leq 2^{-\lambda/2-1}.$$

```

 $\text{Exp}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda):$ 
pp  $\leftarrow$  Setup( $1^\lambda$ )
 $((\text{id}_i)_{i \in [N]}, \text{st}_1) \leftarrow \mathcal{A}_0(\text{pp})$  s.t.  $\text{id}_i \neq \text{id}_j$  for all  $i \neq j$ 
for  $i \in [N]$  do  $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\text{pp}, \text{id}_i)$ 
 $((i^*, j^*), \text{st}_2) \leftarrow \mathcal{A}_1(\text{st}_1, (\text{pk}_i)_{i \in [N]})$ 
 $K^* \leftarrow \mathcal{A}_2(\text{st}_2, (\text{sk}_i)_{i \in [N] \setminus \{i^*, j^*\}})$ 
if  $K^* = \text{SharedKey}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{id}_{j^*}, \text{sk}_{j^*})$  then
  return 1
else
  return 0

```

Figure 3: The N -user 1-step adaptive security game.

Proof. It is easy to verify that for prime q the distribution generating $h_{\mathbf{A}} : \mathbb{Z}_q^m \rightarrow \mathbb{Z}_q^n, \mathbf{u} \mapsto \mathbf{A}\mathbf{u}$ for uniformly random $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ is a universal hash family, and thus we can apply Lemma 4.10 to get

$$\begin{aligned} \text{SD}((\mathbf{A}, \mathbf{A}\mathbf{u}, \mathbf{X}\mathbf{u}), (\mathbf{A}, \mathbf{x}, \mathbf{X}\mathbf{u})) &\leq \frac{1}{2} \sqrt{2^{-H_\infty(\mathbf{u}|\mathbf{X}\mathbf{u})} |\mathbb{Z}_q^n|} \\ &\leq \frac{1}{2} \sqrt{2^{-(m-k \log q)} q^n} = \frac{1}{2} \sqrt{q^{n+k}/2^m} = \frac{1}{2} \sqrt{2^{-\lambda}} = 2^{-\lambda/2-1}. \end{aligned}$$

□

For the ring version of our NIKE, we have to use here the algebraic left-over hash Lemma. We refer the reader to Section B.5.

5 Lower bound

5.1 Preliminaries for metareductions

For this section, we use the regular NIKE notion, that does not distinguish between left and right users, to simplify presentation. Via the tight transformation from asymmetric to regular NIKE sketched in Section 3.2, our results also apply to asymmetric NIKE.

Similar to previous work [Bad+16; HHK18], we consider a weaker and simpler security notion for the metareductions which we call N -user 1-step adaptive security. Since this notion is tightly implied by $(N\text{-user})$ adaptive HKR security, the resulting tightness bound also translates to $(N\text{-user})$ adaptive HKR security.

Definition 5.1 (N -user 1-step adaptive security). A NIKE NIKE is N -user 1-step adaptive secure if for all PPT adversaries $\mathcal{A}$

$$\text{Adv}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda) := \Pr[\text{Exp}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda) \Rightarrow 1]$$

is negligible. The game $\text{Exp}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}$ is defined in Figure 3.

Theorem 5.2 ([HHK18]). *If NIKE NIKE is $(N\text{-user})$ adaptive HKR secure, it is also N -user 1-step adaptive secure. More precisely, for every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ such that*

$$\text{Adv}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda) \leq \text{Adv}_{\mathcal{B}, \text{NIKE}}^{(N\text{-user-})\text{adaptive-HKR}}(\lambda)$$

and $T(\mathcal{A}) \approx T(\mathcal{B})$.

Definition 5.3 (Non-interactive complexity assumption). A *non-interactive complexity assumption* (NICA) NICA consists of three algorithms:

1. Gen inputs the unary encoded security parameter 1^λ and samples a problem instance c with witness w ,

$$\text{Exp}_{\mathcal{B}, \text{NICA}}^{\text{nica}} = (\text{Gen}, \text{Sample}, \text{Verify})(\lambda):$$

$$(c, w) \xleftarrow{\$} \text{Gen}(1^\lambda)$$

$$s \leftarrow \mathcal{B}(c)$$

$$\text{return } \text{Verify}(c, w, s)$$

Figure 4: Security experiment for a non-interactive complexity assumption (NICA).

$$r\text{-}\Lambda^{\mathcal{A}}(c):$$

$$\text{st}_{\Lambda_{1,0}} \leftarrow \Lambda_0(c)$$

$$\text{for } 1 \leq l \leq r \text{ do}$$

$$(\text{pp}^l, \text{st}_{\Lambda_{l,1}}) \leftarrow \Lambda_{l,0}(\text{st}_{\Lambda_{l,0}})$$

$$((\text{id}_i^l)_{i \in [n]}, \text{st}_{\mathcal{A},1}) \leftarrow \mathcal{A}_0(\text{pp}^l) \text{ s.t. } \text{id}_i \neq \text{id}_j \text{ for all } i \neq j$$

$$((\text{pk}_i^l)_{i \in [n]}, \text{st}_{\Lambda_{l,2}}) \leftarrow \Lambda_{l,1}(\text{st}_{\Lambda_{l,1}}, \text{pp}^l, (\text{id}_i^l)_{i \in [n]})$$

$$((\text{id}_{i^*}^l, \text{id}_{j^*}^l), \text{st}_{\mathcal{A},2}) \leftarrow \mathcal{A}_1(\text{st}_{\mathcal{A},1}, (\text{pk}_i^l)_{i \in [n]})$$

$$((\text{sk}_i^l)_{i \in [n] \setminus \{i^{l*}, j^{l*}\}}, \text{st}_{\Lambda_{l,3}}) \leftarrow \Lambda_{l,2}(\text{st}_{\Lambda_{l,2}}, (\text{id}_{i^*}^l, \text{id}_{j^*}^l)).$$

$$K_{i^*, j^*}^l \leftarrow \mathcal{A}_2(\text{st}_{\mathcal{A},2}, (\text{sk}_i^l)_{i \in [n] \setminus \{i^{l*}, j^{l*}\}})$$

$$\text{st}_{\Lambda_{l+1,0}} \leftarrow \Lambda_{l,3}(\text{st}_{\Lambda_{l,3}}, K_{i^*, j^*}^l)$$

$$s \leftarrow \Lambda_4(\text{st}_{\Lambda_{r+1,0}})$$

$$\text{return } s$$

Figure 5: Algorithm $r\text{-}\Lambda^{\mathcal{A}}$, that breaks a non-interactive assumption NICA, constructed from a r -simple reduction $r\text{-}\Lambda = (\Lambda_0, (\Lambda_{l,0}, \Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3})_{l \in [r]}, \Lambda_4)$ and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ against the N -user 1-step adaptive security of NIKE. Note that the state $\text{st}_{\Lambda_{l,1}}$ may be passed to the next iteration of l , while the state $\text{st}_{\mathcal{A},1}$ may not be passed over to the next iteration of l .

2. **Sample** is a PPT algorithm and inputs a problem instance c and outputs a candidate solution s
3. **Verify** inputs a problem instance c with its witness w and a candidate solution s and outputs a bit indicating whether s is a correct solution to the challenge c .

Definition 5.4 The advantage of a PPT adversaries $\mathcal{A}$ against a NICA NICA is defined as

$$\text{Adv}_{\mathcal{A}, \text{NICA}}^{\text{nica}}(\lambda) := |\Pr[\text{Exp}_{\mathcal{A}, \text{NICA}}^{\text{nica}}(\lambda) \Rightarrow 1] - \Pr[\text{Exp}_{\text{Sample}, \text{NICA}}^{\text{nica}}(\lambda) \Rightarrow 1]| \geq \varepsilon(\lambda),$$

where $\text{Exp}_{\mathcal{A}, \text{NICA}}^{\text{nica}}$ is the experiment defined in Figure 4.

The following definition captures a reduction that has only black-box access to the adversary $\mathcal{A}$ and can rewind $\mathcal{A}$ at most r times. The vast majority of reductions used in cryptography fall into this category.

Definition 5.5 (r -simple reduction). An r -simple reduction that reduces the N -user 1-step adaptive security of a NIKE NIKE to a NICA NICA consists of $2 + r \cdot 4$ PPT algorithms $r\text{-}\Lambda = (\Lambda_0, (\Lambda_{l,0}, \Lambda_{l,1}, \Lambda_{l,2}, \Lambda_{l,3})_{l \in [r]}, \Lambda_4)$ with the syntax defined in Figure 5.

We say that an r -simple reduction has runtime $T(r\text{-}\Lambda)$ and security loss $\ell(r\text{-}\Lambda)$ if for any adversary $\mathcal{A}$ the reduction combined with the adversary $r\text{-}\Lambda^{\mathcal{A}}$ has runtime $T(r\text{-}\Lambda^{\mathcal{A}}) \leq r \cdot T(\mathcal{A}) + T(r\text{-}\Lambda)$ and advantage $\text{Adv}_{r\text{-}\Lambda^{\mathcal{A}}, \text{NICA}}^{\text{nica}}(\lambda) \geq r \cdot \text{Adv}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda) / \ell(r\text{-}\Lambda)$.

Note that this definition of security loss is only meaningful if $T(r\text{-}\Lambda)$ is much smaller than $r \cdot T(\mathcal{A})$ (which it typically is, and most previous works ignore the runtime overhead in the analysis even when they incorporate it in the definition).

5.2 Linear lower bound

Definition 5.6 (Pairwise checkable NIKE). We say a NIKE scheme $\text{NIKE} = (\text{Setup}, \text{KeyGen}, \text{SharedKey})$ with identity space $\mathcal{IDS}$ and shared key space $\mathcal{K}$ (with $|\mathcal{IDS}|, |\mathcal{K}| \geq 2$) is $\varepsilon_{\text{PWcheck-pairwise checkable}}$ if

there exists an efficient and deterministic function

$$\text{PairChk} : \mathcal{SPS} \times \mathcal{IDS} \times \mathcal{PK} \times \mathcal{SK} \times \mathcal{IDS} \times \mathcal{PK} \rightarrow \{0, 1\}$$

with the following properties:

1. For all $\lambda \in \mathbb{N}_+$ and all $\text{id}_1, \text{id}_2 \in \mathcal{IDS}$ with $\text{id}_1 \neq \text{id}_2$ for
 - $\text{pp} \leftarrow \text{Setup}(1^\lambda)$,
 - $(\text{pk}_1, \text{sk}_1) \leftarrow \text{KeyGen}(\text{pp}, \text{id}_1)$, and
 - $(\text{pk}_2, \text{sk}_2) \leftarrow \text{KeyGen}(\text{pp}, \text{id}_2)$

we have

$$\Pr[\text{PairChk}(\text{pp}, \text{id}_1, \text{pk}_1, \text{sk}_1, \text{id}_2, \text{pk}_2) = 1] \geq 1 - \varepsilon_{\text{PWcheck}}(\lambda),$$

where $\varepsilon_{\text{PWcheck}}(\lambda) \in \mathcal{O}(\lambda^{-c})$ for some constant $c > 0$, where the probability is taken over the above calls of Setup and KeyGen .

2. For all $\lambda \in \mathbb{N}_+$, all $\text{id}_1, \text{id}_2 \in \mathcal{IDS}$ with $\text{id}_1 \neq \text{id}_2$, all $\text{pk}_1, \text{pk}_2 \in \mathcal{PK}$, and all $\text{sk}_1, \text{sk}'_1, \text{sk}_2 \in \mathcal{SK}$ with
 - $\text{PairChk}(\text{pp}, \text{id}_1, \text{pk}_1, \text{sk}_1, \text{id}_2, \text{pk}_2) = 1$,
 - $\text{PairChk}(\text{pp}, \text{id}_1, \text{pk}_1, \text{sk}'_1, \text{id}_2, \text{pk}_2) = 1$, and
 - $\text{PairChk}(\text{pp}, \text{id}_2, \text{pk}_2, \text{sk}_2, \text{id}_1, \text{pk}_1) = 1$, we have

$$\text{SharedKey}(\text{pp}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{sk}_1) = \text{SharedKey}(\text{pp}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{sk}'_1).$$

6 Lattice-based NIKes are pairwise checkable

6.1 A general rounding function

To make our impossibility result applicable to all existing lattice-based NIKes, we formulate it for general cyclotomic rings $\mathcal{R}$. We refer the reader to Section A.3 for more details.

For our warm-up construction we considered a simplified rounding function that is unsuitable in the regime with polynomial modulus-to-noise ratio. In the following we present a more general rounding function that additionally works for arbitrary cyclotomic rings. This rounding function is also used in our final constructions, and we refer the reader to Section B.1 for the design rationale of this rounding function.

Let q' be the unique integer with $q' \in [q - 2^\rho + 1, q]$ and $2^\rho \mid q'$. We define the following rounding function (for $\beta \in (\{0, 1\}^\rho)^{d'}$, $\gamma \in \{0, 1\}^\rho$)

$$\begin{aligned} \text{Round}_\beta : \mathcal{R}_q &\rightarrow (\{0, 1\}^\rho)^{d'} \\ r &\mapsto (\text{round}_{\beta_1}(\mathbf{x}_1), \dots, \text{round}_{\beta_{d'}}(\mathbf{x}_{d'})) \text{ where } \mathbf{x} = \psi(r) \\ \text{round}_\gamma : \{0, \dots, q-1\} &\rightarrow \{0, 1\}^\rho \\ x &\mapsto \begin{cases} \text{Bits}(x \text{ div } \frac{q'}{2^\rho}) & \text{if } x < q' \\ \gamma & \text{otherwise} \end{cases} \end{aligned}$$

where the coefficients of $\mathcal{R}_q$ (under the fixed embedding ψ) are interpreted as integers in $\{0, \dots, q\}$.

In the following, we call $0, \frac{q'}{2^\rho}, \frac{2q'}{2^\rho}, \dots, q'$ the rounding boundaries and for $B_K \in \mathbb{N}_+$ we call $D_{B_K} := \{x \mid \exists k : |x - \frac{kq'}{2^\rho}| < B_K\} \cup [q', q]$ the B_K -danger zone.

6.2 Lattice based NIKE

In this section we present a general structure of lattice-based NIKE constructions. All known lattice-based NIKE constructions in the literature ([DXL12; Pei14; Lan23; Gaj+24] and this work) have this structure. We will then show that our lower bound holds for all NIKE schemes with this structure.

Definition 6.1 A lattice-based NIKE is a NIKE with the structure depicted in Figure 6 such that for all $\text{id}_1, \text{id}_2 \in \mathcal{IDS}$ with $\text{id}_1 \neq \text{id}_2$,

Setup (1^λ): $\mathbf{A} \xleftarrow{\$} \mathcal{A}$ $\text{pp}_{\text{aux}} \leftarrow \text{Setup}_{\text{aux}}(1^\lambda, \mathbf{A})$ return ($\text{pp} := (\mathbf{A}, \text{pp}_{\text{aux}})$) KeyGen (pp, id): parse ($\mathbf{A}, \text{pp}_{\text{aux}}$) $:= \text{pp}$ $\mathbf{s}_L \xleftarrow{\$} \mathcal{S}_L$; $\mathbf{s}_R \xleftarrow{\$} \mathcal{S}_R$ $\mathbf{e}_L \xleftarrow{\$} \mathcal{E}_L$; $\mathbf{e}_R \xleftarrow{\$} \mathcal{E}_R$ $\text{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$; $\text{pk}_R := \mathbf{A} \mathbf{s}_R + \mathbf{e}_R$ $(\text{pk}_{\text{aux}}, \text{sk}_{\text{aux}}) \leftarrow \text{KeyGen}_{\text{aux}}(\text{pp}, \text{id}, \mathbf{s}_L, \mathbf{s}_R, \mathbf{e}_L, \mathbf{e}_R)$ $\text{pk} := (\text{pk}_L, \text{pk}_R, \text{pk}_{\text{aux}})$ $\text{sk} := (\text{pk}, \mathbf{s}_L, \mathbf{s}_R, \text{sk}_{\text{aux}})$ return (pk, sk)	SharedKey ($\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2$): parse ($\mathbf{A}, \text{pp}_{\text{aux}}$) $:= \text{pp}$ parse ($\text{pk}_{L,1}, \text{pk}_{R,1}, \text{pk}_{\text{aux},1}$) $:= \text{pk}_1$ parse ($\text{pk}_2, \mathbf{s}_{L,2}, \mathbf{s}_{R,2}, \text{sk}_{\text{aux},2}$) $:= \text{sk}_2$ $\mathbf{e}' \leftarrow \text{NoiseGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2)$ $\beta \leftarrow \text{RoundGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2)$ $t \leftarrow \text{OffsetGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2)$ $b \leftarrow \text{ChoiceGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2)$ if $b = 0$ then return $\text{Round}_\beta(\mathbf{s}_{L,2}^\top \cdot \text{pk}_{R,1} + \mathbf{e}' + t)$ else if $b = 1$ then return $\text{Round}_\beta(\text{pk}_{L,1} \cdot \mathbf{s}_{R,2} + \mathbf{e}' + t)$ else return $\perp$
---	--

Figure 6: The general structure of lattice-based NIKE constructions. $\mathcal{A}$ is a distribution on $\mathcal{R}_q^{n \times m}$. $\mathcal{S}_L$ and $\mathcal{E}_R$ are distributions on $\mathcal{R}_q^n$ and likewise $\mathcal{S}_R$ and $\mathcal{E}_L$ are distributions on $\mathcal{R}_q^m$. $\text{Setup}_{\text{aux}}$, $\text{KeyGen}_{\text{aux}}$ are arbitrary PPT algorithms and NoiseGen , RoundGen , OffsetGen , ChoiceGen are deterministic, poly-time algorithms. The rounding function is defined in Section B.1. For simplicity, we depict here $\mathcal{S}_L$, $\mathcal{E}_L$, $\mathcal{E}_R$, and $\mathcal{S}_R$ as separate distributions, but it is possible to use a joined distribution to have correlated samples $\mathbf{s}_L$, $\mathbf{s}_R$, $\mathbf{e}_L$ and $\mathbf{e}_R$ in KeyGen .

- $(\mathbf{A}, \text{pp}_{\text{aux}}) \leftarrow \text{Setup}(1^\lambda)$,
- $(\text{pk}_1 = (\text{pk}_{L,1}, \text{pk}_{R,1}, \text{pk}_{\text{aux},1}), \text{sk}_1 = (\text{pk}_1, \mathbf{s}_{L,1}, \mathbf{s}_{R,1}, \text{sk}_{\text{aux},1})) \leftarrow \text{KeyGen}(\text{pp}, \text{id}_1)$ with $\text{pk}_{L,1} := \mathbf{s}_{L,1}^\top \mathbf{A} + \mathbf{e}_{L,1}^\top$ and $\text{pk}_{R,1} := \mathbf{A} \mathbf{s}_{R,1} + \mathbf{e}_{R,1}$, and
- $(\text{pk}_2 = (\text{pk}_{L,2}, \text{pk}_{R,2}, \text{pk}_{\text{aux},2}), \text{sk}_2 = (\text{pk}_2, \mathbf{s}_{L,2}, \mathbf{s}_{R,2}, \text{sk}_{\text{aux},2})) \leftarrow \text{KeyGen}(\text{pp}, \text{id}_2)$ with $\text{pk}_{L,2} := \mathbf{s}_{L,2}^\top \mathbf{A} + \mathbf{e}_{L,2}^\top$ and $\text{pk}_{R,2} := \mathbf{A} \mathbf{s}_{R,2} + \mathbf{e}_{R,2}$

we have with probability $1 - \varepsilon_{\text{bounded}}(\lambda)$ (where the probability is taken over the above calls of Setup and KeyGen)

- $\text{OffsetGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2) = \text{OffsetGen}(\text{pp}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{pk}_1)$,
- for $b_1 \leftarrow \text{ChoiceGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2)$ and $b_2 \leftarrow \text{ChoiceGen}(\text{pp}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{pk}_1)$ we have $b_1, b_2 \in \{0, 1\}$ and $b_1 = 1 - b_2$,
- $\|\mathbf{s}_{L,1}\|_\infty \leq B_{\mathbf{s},L}$,
- $\|\mathbf{e}_{L,1}\|_\infty \leq B_{\mathbf{e},L}$,
- $\|\mathbf{s}_{R,1}\|_\infty \leq B_{\mathbf{s},R}$,
- $\|\mathbf{e}_{R,1}\|_\infty \leq B_{\mathbf{e},R}$,
- $\|\text{NoiseGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2)\|_\infty \leq B_{\mathbf{e}'}$,
- $\mathbf{s}_{L,2}^\top \cdot \text{pk}_{R,1} + \text{NoiseGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2) + \text{OffsetGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2)$ has at most statistical distance $\varepsilon_{\text{nonuniform}}(\lambda)$ to the uniform distribution on $\mathcal{R}_q$, and
- $\text{pk}_{L,1} \cdot \mathbf{s}_{R,2} + \text{NoiseGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2) + \text{OffsetGen}(\text{pp}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2)$ has at most statistical distance $\varepsilon_{\text{nonuniform}}(\lambda)$ to the uniform distribution on $\mathcal{R}_q$

for parameters $B_{\mathbf{s},L}, B_{\mathbf{e},L}, B_{\mathbf{s},R}, B_{\mathbf{e},R}, B_{\mathbf{e}'} \in \mathbb{R}_{\geq 0}$. For convenience, we define $B_K := n \cdot B_{\mathbf{s},L} \cdot B_{\mathbf{e},R} + m \cdot B_{\mathbf{s},R} \cdot B_{\mathbf{e},L} + B_{\mathbf{e}'}$. Our lower bound only gives meaningful results if $B_K \ll q$.

We briefly sketch how our and previous lattice-based NIKE construction satisfy Definition 6.1. All constructions use as $\mathbf{A}$ the uniform distribution on $\mathcal{R}_q^{n \times m}$.

For NIKE_{sp} from Figure 9 $\mathcal{S}_L$ is the uniform distribution on $\mathcal{R}_q^n$, $\mathcal{S}_R$ is the uniform distribution on $\{0, 1\}^m$ (or any other short distribution), $\mathcal{E}_L$ is a discrete Gaussian distribution, and $\mathcal{E}_R$ is the singleton distribution that always outputs $\mathbf{0} \in \mathcal{R}_q^m$. These satisfy the norm bounds with $B_{\mathbf{s},L} = q$, $B_{\mathbf{s},R} = 1$, $B_{\mathbf{e},L} = m\sqrt{md\sigma_{\max}(\Sigma)/\pi}$ (with probability at least $1 - \sqrt{2}^{md}e^{-md/2}$), $B_{\mathbf{e},R} = 0$, and $B_{\mathbf{e}'} = 0$, resulting in the same B_K as in Theorem B.5 with probability $1 - \varepsilon_{\text{bounded}}(\lambda)$ for negligible $\varepsilon_{\text{bounded}}(\lambda)$. $\text{Setup}_{\text{aux}}$ does nothing and OffsetGen always returns 0. The scheme satisfies the last requirement with $\varepsilon_{\text{nonuniform}}(\lambda) = 2^{-\lambda/2-1} + (gq^{-f})^n$. The algorithm NoiseGen always returns $\mathbf{0} \in \mathcal{R}_q$ and the algorithm RoundGen always returns the bitstring 0^ρ . Finally, ChoiceGen is the function used to make the NIKE

symmetric, for example by comparing the identities. The algorithm $\text{KeyGen}_{\text{aux}}$ does nothing in this case.

For NIKE_p from Figure 10 the argument is similar, we just use additionally $\text{KeyGen}_{\text{aux}}$ to generate the PRF key and store it in sk_{aux} . The algorithms NoiseGen and RoundGen then evaluate the PRF to compute their respective outputs. This NIKE achieves the bounds $B_{e'} = \sqrt{(n+1)d\sigma_{\max}(\mathbf{T})/2\pi}$ (with probability at least $1 - \sqrt{n+1}^d e^{-nd/2}$) and all other bounds as before, resulting in the bound B_K defined in Theorem B.13.

The NIKes considered in [Lan23] also fit this framework, by selecting $n = m$ and using a short distribution for $\mathcal{S}_L$ and $\mathcal{S}_R$ and a discrete Gaussian distribution for $\mathcal{E}_L$ and $\mathcal{E}_R$. Instead of using a random offset t for the rounding function, they show that the pre-rounding shared keys are statistically close to uniform directly, which leads to a negligible $\varepsilon_{\text{nonuniform}}(\lambda)$ in our framework. Thus they do not need $\text{Setup}_{\text{aux}}$ and OffsetGen . In their variant for polynomial modulus-to-noise ratio, we need two PRFs like for NIKE_p and thus $\text{KeyGen}_{\text{aux}}$ and NoiseGen work as for NIKE_p . In their adaptive DKR secure variant with super-polynomial modulus-to-noise ratio NoiseGen ignores the input and outputs a smudging size noise term¹² ($\text{KeyGen}_{\text{aux}}$ does nothing in this case). As before, ChoiceGen is the function used to make the NIKE symmetric.

The NIKE considered in [Gaj+24] fits this framework by using a short binomial distribution for $\mathcal{S}_L$, $\mathcal{S}_R$, $\mathcal{E}_L$, and $\mathcal{E}_R$. Their construction uses OffsetGen to query the random oracle on the identities and public keys and output the result as element in $\mathcal{R}_q$. The algorithms $\text{Setup}_{\text{aux}}$, $\text{KeyGen}_{\text{aux}}$ and NoiseGen do nothing in their variant. As before, ChoiceGen is the function used to make the NIKE symmetric.

To capture the Ring-LWE based constructions used in [DXL12; Pei14] (with $n = m = 1$), that are already a symmetric NIKE without generating two different key pairs for each user, we can use a joint distribution for $\mathbf{s}_L$ and $\mathbf{s}_R$ to enforce $\mathbf{s}_L = \mathbf{s}_R$ and similarly use a joint distribution for $\mathbf{e}_L$ and $\mathbf{e}_R$ to enforce $\mathbf{e}_L = \mathbf{e}_R$. To satisfy our definition, we still need a ChoiceGen that outputs bits as before (for example by comparing the identities), but the output bit has no influence on the resulting shared key.

Finally, for variants where public keys are equipped with a NIZKPoK of the secret key (as in the adaptive DKR secure variants in [Lan23; Gaj+24]), we can use $\text{KeyGen}_{\text{aux}}$ to generate these proofs and store them in the public key. ChoiceGen can then verify these proofs and output $\perp$ if they do not verify, to force SharedKey to output $\perp$, too.

6.3 Pairwise checkability of LWE-based NIKE

```

PairChk(pp = (A, ppaux), id1, pk1, sk1, id2, pk2):
  b ← ChoiceGen(ppaux, id2, pk2, id1, pk1)
  if b ∉ {0, 1} then return 0
  parse (pkL,1, pkR,1, pkaux,1) := pk1 //return 0 if pk1 does not have this format
  parse (pk1, sL,1, sR,1, skaux,1) := sk1 //return 0 if sk1 does not have this format
  e' ← NoiseGen(ppaux, id2, pk2, id1, sk1)
  t ← OffsetGen(ppaux, id2, pk2, id1, pk1)
  if ||sL,1||∞ > Bs,L ∨ ||eL,2||∞ > Be,L ∨ ||sR,1||∞ > Bs,R ∨ ||eR,2||∞ > Be,R ∨ ||e'||∞ > Be' then
    return 0
  if b = 0 then
    | k = sL,2 · pkR,1 + e' + t
  else
    | k = pkL,1 · sR,2 + e' + t
  if k ∈ DBK then //See Section B.1 for the definition of DBK
    | return 0
  else
    | return 1

```

Figure 7: The PairChk algorithm for lattice-based NIKes in the sense of Definition 6.1.

¹²The variant described in [Lan23] samples this noise-term at random. To fit our framework, we have to make this deterministic, e.g. by generating the noise term via a PRF, whose secret key is stored in sk_{aux} .

Here, we show that any lattice-based NIKE in the sense of Definition 6.1 is pairwise checkable. The PairChk algorithm for lattice-based NIKES is defined in Figure 7.

We show that this definition of PairChk satisfies property (1), i.e. honestly generated key pairs pass the check with probability at least $1 - \varepsilon_{\text{PWcheck}}(\lambda)$ for $\varepsilon_{\text{PWcheck}}(\lambda) := \varepsilon_{\text{bounded}}(\lambda) + \varepsilon_{\text{nonuniform}}(\lambda) + 2^\rho d'(2B_K + 1)/q$, where B_K is defined as in Definition 6.1. Let $(\mathbf{A}, \text{pp}_{\text{aux}}) \leftarrow \text{Setup}(1^\lambda)$, $(\text{pk}_1 = (\text{pk}_{L,1}, \text{pk}_{R,1}, \text{pk}_{\text{aux},1}), \text{sk}_1 = (\text{pk}_1, \mathbf{s}_{L,1}, \mathbf{s}_{R,1}, \text{sk}_{\text{aux},1})) \leftarrow \text{KeyGen}(\text{pp}, \text{id})$ (with $\text{pk}_{L,1} := \mathbf{s}_{L,1}^\top \mathbf{A} + \mathbf{e}_{L,1}^\top$ and $\text{pk}_{R,1} := \mathbf{A} \mathbf{s}_{R,1} + \mathbf{e}_{R,1}$) and $(\text{pk}_2 = (\text{pk}_{L,2}, \text{pk}_{R,2}, \text{pk}_{\text{aux},2}), \text{sk}_2 = (\text{pk}_2, \mathbf{s}_{L,2}, \mathbf{s}_{R,2}, \text{sk}_{\text{aux},2})) \leftarrow \text{KeyGen}(\text{pp}, \text{id})$ (with $\text{pk}_{L,2} := \mathbf{s}_{L,2}^\top \mathbf{A} + \mathbf{e}_{L,2}^\top$ and $\text{pk}_{R,2} := \mathbf{A} \mathbf{s}_{R,2} + \mathbf{e}_{R,2}$). Assume from now on that these key pairs satisfy the properties listed in Definition 6.1. This happens with probability $1 - \varepsilon_{\text{bounded}}(\lambda)$.

Next, we show that under this condition $\text{PairChk}(\text{pp} = (\mathbf{A}, \text{pp}_{\text{aux}}), \text{id}_1, \text{pk}_1, \text{sk}_1, \text{id}_2, \text{pk}_2) = 1$ with probability at least $1 - 2^\rho d'(2B_K + 1)/q$. The condition $b := \text{ChoiceGen}(\text{pp}_{\text{aux}}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2) \in \{0, 1\}$ and the norm bounds have direct counterparts in Definition 6.1 and are guaranteed to be satisfied.

For the last condition, assume that $b = 0$ and thus $k = \mathbf{s}_{L,2}^\top \cdot \text{pk}_{R,1} + e' + t$. The case $b = 1$ is analogous. The properties of a lattice-based NIKE guarantee that k is within statistical distance $\varepsilon_{\text{nonuniform}}(\lambda)$ to uniformly random in $\mathcal{R}_q$. Thus, the probability that k is in the B_K -danger zone D_{B_K} is $|D_{B_K}|d'/q \leq 2^\rho d'(2B_K + 1)/q$ by Lemma B.4.

Next, we need to show that this definition of PairChk satisfies property (2). Therefore, let $\text{id}_1, \text{id}_2 \in \mathcal{IDS}$ with $\text{id}_1 \neq \text{id}_2$, $\text{pk}_1, \text{pk}_2 \in \mathcal{PK}$, and $\text{sk}_1, \text{sk}'_1, \text{sk}_2 \in \mathcal{SK}$ such that

- $\text{PairChk}((\mathbf{A}, \text{pp}_{\text{aux}}), \text{id}_1, \text{pk}_1, \text{sk}_1, \text{id}_2, \text{pk}_2) = 1$,
- $\text{PairChk}((\mathbf{A}, \text{pp}_{\text{aux}}), \text{id}_1, \text{pk}_1, \text{sk}'_1, \text{id}_2, \text{pk}_2) = 1$, and
- $\text{PairChk}((\mathbf{A}, \text{pp}_{\text{aux}}), \text{id}_2, \text{pk}_2, \text{sk}_2, \text{id}_1, \text{pk}_1) = 1$.

Then, we have $\text{pk}_1 = (\text{pk}_{L,1}, \text{pk}_{R,1}, \text{pk}_{\text{aux},1}), \text{sk}_1 = (\text{pk}_1, \mathbf{s}_{L,1}, \mathbf{s}_{R,1}, \text{sk}_{\text{aux},1}), \text{sk}'_1 = (\text{pk}_1, \mathbf{s}'_{L,1}, \mathbf{s}'_{R,1}, \text{sk}'_{\text{aux},1}), \text{pk}_2 = (\text{pk}_{L,2}, \text{pk}_{R,2}, \text{pk}_{\text{aux},2}), \text{sk}_2 = (\text{pk}_2, \mathbf{s}_{L,2}, \mathbf{s}_{R,2}, \text{sk}_{\text{aux},2})$. Assume $\text{ChoiceGen}(\text{pp}_{\text{aux}}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2) = 0$. The case $\text{ChoiceGen}(\text{pp}_{\text{aux}}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2) = 1$ is analogous. Now let $k_1 := \mathbf{s}_{L,1}^\top \cdot \text{pk}_{R,2} + \text{NoiseGen}(\text{pp}_{\text{aux}}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{sk}_1) + \text{OffsetGen}(\text{pp}_{\text{aux}}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{pk}_1)$ (the pre-rounding shared key computed with sk_1), $k'_1 := (\mathbf{s}'_{L,1})^\top \cdot \text{pk}_{R,2} + \text{NoiseGen}(\text{pp}_{\text{aux}}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{sk}'_1) + \text{OffsetGen}(\text{pp}_{\text{aux}}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{pk}_1)$ (the pre-rounding shared key computed with sk'_1), and $k_2 := \text{pk}_{L,1}^\top \cdot \mathbf{s}_{R,2} + \text{NoiseGen}(\text{pp}_{\text{aux}}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2) + \text{OffsetGen}(\text{pp}_{\text{aux}}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{pk}_2)$ (the pre-rounding shared key computed with sk_2). Both k_1 and k'_1 (and k_2 , but that is not needed here) are not inside the B_K -danger zone D_{B_K} . Thus, by Lemma B.3, we have $\text{Round}_{\beta_1}(k_1) = \text{Round}_{\beta_2}(k_2)$ and $\text{Round}_{\beta'_1}(k'_1) = \text{Round}_{\beta_2}(k_2)$ (where $\beta_1 \leftarrow \text{RoundGen}(\text{pp}_{\text{aux}}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{sk}_1)$, $\beta'_1 \leftarrow \text{RoundGen}(\text{pp}_{\text{aux}}, \text{id}_2, \text{pk}_2, \text{id}_1, \text{sk}'_1)$, and $\beta_2 \leftarrow \text{RoundGen}(\text{pp}_{\text{aux}}, \text{id}_1, \text{pk}_1, \text{id}_2, \text{sk}_2)$). In particular, we have $\text{Round}_{\beta_1}(k_1) = \text{Round}_{\beta'_1}(k'_1)$, what we needed to show.

6.3.1 Metareduction

Theorem 6.2 *Any r -simple reduction that reduces the N -user 1-step adaptive security of a pairwise checkable NIKE NIKE to a NICA NICA must incur a security loss of $\Omega(N)$. More precisely, for any such reduction $r\text{-}\Lambda$ there exists an adversary $\mathcal{B}$ breaking the NICA NICA with $\mathsf{T}(\mathcal{B}) \approx N(N-1)\mathsf{T}(r\text{-}\Lambda)$ and*

$$\text{Adv}_{\mathcal{B}, \text{NICA}}^{\text{nica}}(\lambda) \geq r((1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda))/\ell(r\text{-}\Lambda) - 2/N).$$

Proof. We first describe an inefficient adversary $\mathcal{A}$ against the N -user 1-step adaptive security of any pairwise checkable NIKE NIKE. In a second step, we will give an efficient adversary $\mathcal{B}$ against the NICA NICA that internally runs the reduction $r\text{-}\Lambda$ and (tries to) efficiently simulate the responses of $\mathcal{A}$. Both adversaries are given in Figure 8.

First, we analyze the advantage of $\mathcal{A}$. Therefore, note that for N honestly generated key pairs $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\text{pp}, \text{id}_i)$ we have $\Pr[\forall i, j \in [N]^2 \setminus \{(k, k) | k \in [N]\} : \text{PairChk}(\text{pp}, \text{id}_i, \text{pk}_i, \text{sk}_i, \text{id}_j, \text{pk}_j) = 1] \geq 1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda)$ by applying the union bound. If this occurs for all key pairs generated by the N -user 1-step adaptive security game, the adversary $\mathcal{A}$ wins this game. Firstly, note the check that $\mathcal{A}_2$ performs on the retrieved public and secret keys passes. Secondly, since for *both* challenge users i^* and j^* there exist corresponding secret keys $\text{sk}_{i^*}, \text{sk}_{j^*} \in \mathcal{SK}$ such that for all $j \in [N]$ $\text{PairChk}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{sk}_{i^*}, \text{id}_j, \text{pk}_j) = 1$ and $\text{PairChk}(\text{pp}, \text{id}_{j^*}, \text{pk}_{j^*}, \text{sk}_{j^*}, \text{id}_j, \text{pk}_j) = 1$, the brute-force search for secrets keys sk_{i^*} and sk_{j^*} succeeds. The pairwise checkability property of NIKE guarantees in this situation that every secret key $\text{sk} \in \mathcal{SK}$ that fulfills the brute-force search criterion of $\mathcal{A}_2$ (namely, $\text{PairChk}(\text{pp}, \text{id}_{j^*}, \text{pk}_{j^*}, \text{sk}, \text{id}_{i^*}, \text{pk}_{i^*}) = 1$) will satisfy $\text{SharedKey}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{id}_{j^*}, \text{sk}) = \text{SharedKey}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{id}_{j^*}, \text{sk}_{j^*})$. In this case, $\mathcal{A}$ computes

$\begin{aligned} &\underline{\mathcal{A}_0(\text{pp})}: \\ &\text{for } i \in [N] \text{ do } \text{id}_i \xleftarrow{\$} \text{IDS}, \text{ with } \text{id}_i \neq \text{id}_j \text{ for } i \neq j \\ &\text{st}_1 := (\text{pp}, (\text{id}_i)_{i \in [N]}). \\ &\text{return } ((\text{id}_i)_{i \in [N]}, \text{st}_1) \\ \\ &\underline{\mathcal{A}_1(\text{st}_1, (\text{pk}_i)_{i \in [N]}):} \\ &(i^*, j^*) \xleftarrow{\$} [N]^2 \setminus \{(k, k) k \in [N]\} \\ &\text{st}_2 := (\text{st}_1, (\text{pk}_i)_{i \in [N]}, (i^*, j^*)) \\ &\text{return } ((i^*, j^*), \text{st}_2) \\ \\ &\underline{\mathcal{A}_2(\text{st}_2, (\text{sk}_i)_{i \in [N] \setminus \{i^*, j^*\}}):} \\ &\text{if } \bigwedge_{i \in [N] \setminus \{i^*, j^*\}, j \in [N] \setminus \{i\}} \text{PairChk}(\text{pp}, \text{id}_i, \text{pk}_i, \text{sk}_i, \text{id}_j, \text{pk}_j) = 1 \\ &\text{then} \\ &\quad \text{sk}_{i^*} := \perp; \text{sk}_{j^*} := \perp \\ &\quad \text{for } \text{sk} \in \mathcal{SK} \text{ do} \\ &\quad \quad \text{if } \bigwedge_{j \in [N] \setminus \{i^*\}} \text{PairChk}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{sk}_{i^*}, \text{id}_j, \text{pk}_j) = 1 \\ &\quad \quad \quad \text{then } \text{sk}_{i^*}^* \leftarrow \text{sk} \\ &\quad \quad \text{if } \bigwedge_{j \in [N] \setminus \{j^*\}} \text{PairChk}(\text{pp}, \text{id}_{j^*}, \text{pk}_{j^*}, \text{sk}_{j^*}, \text{id}_j, \text{pk}_j) = 1 \\ &\quad \quad \quad \text{then } \text{sk}_{j^*}^* \leftarrow \text{sk} \\ &\quad \text{if } \text{sk}_{i^*} = \perp \vee \text{sk}_{j^*} = \perp \text{ then} \\ &\quad \quad \text{return } \perp \\ &\quad \text{else} \\ &\quad \quad \text{return } \text{SharedKey}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{id}_{j^*}, \text{sk}_{i^*}^*). \\ &\text{else} \\ &\quad \text{return } \perp \end{aligned}$	$\begin{aligned} &\underline{\mathcal{B}(c)}: \\ &\text{st}_{\Lambda_{1,0}} \leftarrow \Lambda_0(c) \\ &\text{for } 1 \leq l \leq r \text{ do} \\ &\quad (\text{pp}, \text{st}_{\Lambda_{l,1}}) \leftarrow \Lambda_{l,0}(\text{st}_{\Lambda_{l,0}}) \\ &\quad \text{for } i \in [N] \text{ do } \text{id}_i \xleftarrow{\$} \text{IDS}, \text{ with } \text{id}_i \neq \text{id}_j \text{ for } i \neq j \\ &\quad ((\text{pk}_i)_{i \in [N]}, \text{st}_{\Lambda_{l,2}}) \leftarrow \Lambda_{l,1}(\text{st}_{\Lambda_{l,1}}, \text{pp}, (\text{id}_i^l)_{i \in [N]}) \\ &\quad (i^*, j^*) \xleftarrow{\$} [N]^2 \setminus \{(k, k) k \in [N]\} \\ &\quad \text{for } (x, y) \in [N]^2 \setminus (\{(k, k) k \in [N]\} \cup \{(i^*, j^*), (j^*, i^*)\}) \text{ do} \\ &\quad \quad ((\text{sk}_i)_{i \in [N] \setminus \{x, y\}}, \_) \leftarrow \Lambda_{l,2}(\text{st}_{\Lambda_{l,2}}, (x, y)) \\ &\quad \quad \text{if } \bigwedge_{i \in [N] \setminus \{x, y\}, j \in [N] \setminus \{i\}} \text{PairChk}(\text{pp}, \text{id}_i, \text{pk}_i, \text{sk}_i, \text{id}_j, \text{pk}_j) = 1 \\ &\quad \quad \quad \text{then} \\ &\quad \quad \quad \text{for } i \in [N] \setminus \{x, y\} \text{ do } \text{SK}[i] \leftarrow \text{sk}_i \\ &\quad \quad ((\text{sk}_i)_{i \in [N] \setminus \{i^*, j^*\}}, \text{st}_{\Lambda_{l,3}}) \leftarrow \Lambda_{l,2}(\text{st}_{\Lambda_{l,2}}, (i^*, j^*)) \\ &\quad \quad \text{if } \bigwedge_{i \in [N] \setminus \{i^*, j^*\}, j \in [N] \setminus \{i\}} \text{PairChk}(\text{pp}, \text{id}_i, \text{pk}_i, \text{sk}_i, \text{id}_j, \text{pk}_j) = 1 \\ &\quad \quad \quad \text{then} \\ &\quad \quad \quad \text{for } i \in [N] \setminus \{i^*, j^*\} \text{ do } \text{SK}[i] \leftarrow \text{sk}_i \\ &\quad \quad \quad \text{if } \text{SK}[i^*] \neq \perp \wedge \text{SK}[j^*] \neq \perp \text{ then} \\ &\quad \quad \quad \quad K_{i^*, j^*}^l \leftarrow \text{SharedKey}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{id}_{j^*}, \text{SK}[j^*]) \\ &\quad \quad \quad \quad \text{else} \\ &\quad \quad \quad \quad K_{i^*, j^*}^l := \perp \quad // \text{Simulation failure} \\ &\quad \quad \text{else} \\ &\quad \quad \quad K_{i^*, j^*}^l := \perp \\ &\quad \text{st}_{\Lambda_{l+1,0}} \leftarrow \Lambda_{l,3}(\text{st}_{\Lambda_{l,3}}, K_{i^*, j^*}^l) \\ &\quad s \leftarrow \Lambda_4(\text{st}_{\Lambda_{r+1,0}}) \\ &\text{return } s \end{aligned}$
--	--

Figure 8: The left column shows the inefficient adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, that breaks the N -user 1-step adaptive security of any pairwise checkable NIKE NIKE. The right column shows the adversary $\mathcal{B}$ against the NICA NICA that internally runs the reduction r - Λ and (tries to) efficiently simulate the responses of $\mathcal{A}$.

the challenge shared key using a secret key satisfying this check. Thus, the adversary $\mathcal{A}$ has advantage $\text{Adv}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda) \geq 1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda)$.

To argue about the success probability of $\mathcal{B}$, we define the following two events for each run $l \in [r]$:

check-fails^l: This event occurs iff the secret keys that $\mathcal{B}$ retrieves in the l -th iteration of the main loop in the run of the reduction for the real challenge pair $((\text{sk}_i)_{i \in [N] \setminus \{i^*, j^*\}}, \text{st}_{\Lambda_{l,3}}) \leftarrow \Lambda_{l,2}(\text{st}_{\Lambda_{l,2}}, (i^*, j^*))$ do *not* satisfy

$$\bigwedge_{i \in [N] \setminus \{i^*, j^*\}, j \in [N] \setminus \{i\}} \text{PairChk}(\text{pp}, \text{id}_i, \text{pk}_i, \text{sk}_i, \text{id}_j, \text{pk}_j) = 1$$

sks-retrieved^l: This event occurs if $\text{SK}[i^*] \neq \perp \wedge \text{SK}[j^*] \neq \perp$ at the end of the l -th iteration of the main loop in $\mathcal{B}$.

First, we argue that if $\text{check-fails}^l \vee \text{sks-retrieved}^l$, $\mathcal{B}$ perfectly simulates $\mathcal{A}$ in the l -th run of the reduction. If check-fails^l occurs, $\mathcal{A}$ and $\mathcal{B}$ both output $\perp$ thus behave identically. If sks-retrieved^l occurs, we know that for both challenge users i^* and j^* there exist corresponding secret keys $\text{sk}_{i^*}, \text{sk}_{j^*} \in \mathcal{SK}$ such that for all $j \in [N]$ $\text{PairChk}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{sk}_{i^*}, \text{id}_j, \text{pk}_j) = 1$ and $\text{PairChk}(\text{pp}, \text{id}_{j^*}, \text{pk}_{j^*}, \text{sk}_{j^*}, \text{id}_j, \text{pk}_j) = 1$. Thus, if $\neg \text{check-fails}^l \wedge \text{sks-retrieved}^l$ occurs, $\mathcal{A}$ outputs a shared key computed with a secret key sk obtained via the brute-force search. $\mathcal{B}$ outputs a shared key computed with $\text{SK}[j^*]$. Since these keys satisfy $\text{PairChk}(\text{pp}, \text{id}_{j^*}, \text{pk}_{j^*}, \text{sk}, \text{id}_{i^*}, \text{pk}_{i^*}) = 1$ and $\text{PairChk}(\text{pp}, \text{id}_{j^*}, \text{pk}_{j^*}, \text{SK}[j^*], \text{id}_{i^*}, \text{pk}_{i^*}) = 1$ and there exists a secret key sk_{i^*} with $\text{PairChk}(\text{pp}, \text{id}_{i^*}, \text{pk}_{i^*}, \text{sk}_{i^*}, \text{id}_{j^*}, \text{pk}_{j^*}) = 1$, the pairwise checkability of NIKE guarantees that the shared keys that $\mathcal{A}$ and $\mathcal{B}$ output are identical.

Next, we bound the probability of $\neg \text{check-fails}^l \wedge \neg \text{sks-retrieved}^l$ occurring. Therefore, observe that $\neg \text{sks-retrieved}^l$ can only occur when there is an index $i \in [N]$ such that $\text{SK}[i] = \perp$ at the end of the l -th iteration of the main loop in $\mathcal{B}$. Thus, we assume in the following that such an index exists. Let i be the smallest index with $\text{SK}[i] = \perp$. Since $\mathcal{B}$ samples (i^*, j^*) uniformly at random and the view of the reduction is independent of (i^*, j^*) until $\mathcal{B}$ invokes $\Lambda_{l,3}$, i is independent of i^* and j^* . Thus, we have $\Pr[i \in [N] \setminus \{i^*, j^*\}] = 1 - 2/N$. However, if $i \in [N] \setminus \{i^*, j^*\}$, check-fails^l must occur, because the adversary would add secret keys for all users in $[N] \setminus \{i^*, j^*\}$ to SK in the final iteration when check-fails^l does not occur.

This shows $\Pr[\neg \text{check-fails}^l \wedge \neg \text{sks-retrieved}^l] \leq 2/N$ and, via a union bound, $\Pr[\forall l \in [r] : \neg \text{check-fails}^l \wedge \neg \text{sks-retrieved}^l] \leq 2r/N$. Thus $\mathcal{B}$ has advantage

$$\begin{aligned} \text{Adv}_{\mathcal{B}, \text{NICA}}^{\text{nica}}(\lambda) &\geq \text{Adv}_{r-\Lambda^{\mathcal{A}}, \text{NICA}}^{\text{nica}}(\lambda) - 2r/N \geq r \cdot \text{Adv}_{\mathcal{A}, \text{NIKE}}^{N\text{-user-1-step-adaptive}}(\lambda) / \ell(r-\Lambda) \\ &\quad - 2r/N \geq r((1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda)) / \ell(r-\Lambda) - 2/N). \end{aligned}$$

The runtime of the adversary consists of running the reduction in total $N(N-1)$ times and some overhead for bookkeeping and running PairChk $\mathcal{O}(r \cdot N^4)$ many times. $\square$

Interpretation of Theorem 6.2. The result of Theorem 6.2 can be rearranged to

$$\ell(r-\Lambda) \geq \frac{1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda)}{\frac{\text{Adv}_{\mathcal{B}, \text{NICA}}^{\text{nica}}(\lambda)}{r} + \frac{2}{N}}.$$

If NICA is secure, then for every PPT adversary $\mathcal{B}$ (in particular the one given in Figure 8) $\text{Adv}_{\mathcal{B}, \text{NICA}}^{\text{nica}}(\lambda)$ is negligible, and thus $\frac{\text{Adv}_{\mathcal{B}, \text{NICA}}^{\text{nica}}(\lambda)}{r} + \frac{2}{N} \in \Theta(\frac{1}{N})$ which implies $\ell(r-\Lambda) \in \Omega(N(1 - N^2 \varepsilon_{\text{PWcheck}}(\lambda)))$. Moreover, in the lattice-based NIKE schemes we present in this work it is easy to set parameters such that $\varepsilon_{\text{PWcheck}}(\lambda) \leq \frac{1}{2N^2}$ (even with polynomial modulus-to-noise ratio) and in this case the theorem shows $\ell(r-\Lambda) \in \Omega(N)$.

Acknowledgments. Roman Langrehr was supported by SNSF Postdoc.Mobility grant no. 235425.

References

- [Alw+23] Joël Alwen, Jonas Janneck, Eike Kiltz, and Benjamin Lipp. “The Pre-Shared Key Modes of HPKE”. in: *ASIACRYPT 2023, Part VI*. ed. by Jian Guo and Ron Steinfeld. Vol. 14443. LNCS. Springer, Singapore, Dec. 2023, pp. 329–360. DOI: [10.1007/978-981-99-8736-8_11](https://doi.org/10.1007/978-981-99-8736-8_11).
- [Ang+22] Yawning Angel, Benjamin Dowling, Andreas Hülsing, Peter Schwabe, and Fiona Johanna Weber. “Post Quantum Noise”. In: *ACM CCS 2022*. Ed. by Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi. ACM Press, Nov. 2022, pp. 97–109. DOI: [10.1145/3548606.3560577](https://doi.org/10.1145/3548606.3560577).
- [Bad+16] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. “On the Impossibility of Tight Cryptographic Reductions”. In: *EUROCRYPT 2016, Part II*. ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Berlin, Heidelberg, May 2016, pp. 273–304. DOI: [10.1007/978-3-662-49896-5_10](https://doi.org/10.1007/978-3-662-49896-5_10).
- [Bas+26] Andrea Basso, Giacomo Borin, Ryan Rueger, and Sina Schaeffler. *CORAL Faster Isogeny Group Action for Post-Quantum NIKE*. to appear at Crypto 2026. 2026. URL: <https://eprint.iacr.org/2026/896>.
- [Bel+24] Mihir Bellare, Doreen Riepel, Stefano Tessaro, and Yizhao Zhang. “Count Corruptions, Not Users: Improved Tightness for Signatures, Encryption and Authenticated Key Exchange”. In: *ASIACRYPT 2024, Part II*. ed. by Kai-Min Chung and Yu Sasaki. Vol. 15485. LNCS. Springer, Singapore, Dec. 2024, pp. 326–360. DOI: [10.1007/978-981-96-0888-1_11](https://doi.org/10.1007/978-981-96-0888-1_11).
- [Ben+18] Fabrice Benhamouda, Olivier Blazy, Léo Ducas, and Willy Quach. “Hash Proof Systems over Lattices Revisited”. In: *PKC 2018, Part II*. ed. by Michel Abdalla and Ricardo Dahab. Vol. 10770. LNCS. Springer, Cham, Mar. 2018, pp. 644–674. DOI: [10.1007/978-3-319-76581-5_22](https://doi.org/10.1007/978-3-319-76581-5_22).
- [Bet+22] Slim Bettaiieb, Loïc Bidoux, Olivier Blazy, Yann Connan, and Philippe Gaborit. “A gapless code-based hash proof system based on RQC and its applications”. In: *DCC 90.12 (2022)*, pp. 3011–3044. DOI: [10.1007/s10623-022-01075-7](https://doi.org/10.1007/s10623-022-01075-7).
- [BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) fully homomorphic encryption without bootstrapping”. In: *ITCS 2012*. Ed. by Shafi Goldwasser. ACM, Jan. 2012, pp. 309–325. DOI: [10.1145/2090236.2090262](https://doi.org/10.1145/2090236.2090262).
- [BMP04] Colin Boyd, Wenbo Mao, and Kenneth G. Paterson. “Key Agreement Using Statically Keyed Authenticators”. In: *ACNS 2004*. Ed. by Markus Jakobsson, Moti Yung, and Jianying Zhou. Vol. 3089. LNCS. Springer, Berlin, Heidelberg, June 2004, pp. 248–262. DOI: [10.1007/978-3-540-24852-1_18](https://doi.org/10.1007/978-3-540-24852-1_18).
- [Bra+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. “Classical hardness of learning with errors”. In: *45th ACM STOC*. ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 575–584. DOI: [10.1145/2488608.2488680](https://doi.org/10.1145/2488608.2488680).
- [Cas+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. “CSIDH: An Efficient Post-Quantum Commutative Group Action”. In: *ASIACRYPT 2018, Part III*. ed. by Thomas Peyrin and Steven Galbraith. Vol. 11274. LNCS. Springer, Cham, Dec. 2018, pp. 395–427. DOI: [10.1007/978-3-030-03332-3_15](https://doi.org/10.1007/978-3-030-03332-3_15).
- [Cho+20] Jérémy Chotard, Edouard Dufour-Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. “Dynamic Decentralized Functional Encryption”. In: *CRYPTO 2020, Part I*. ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12170. LNCS. Springer, Cham, Aug. 2020, pp. 747–775. DOI: [10.1007/978-3-030-56784-2_25](https://doi.org/10.1007/978-3-030-56784-2_25).
- [CKS08] David Cash, Eike Kiltz, and Victor Shoup. “The Twin Diffie-Hellman Problem and Applications”. In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Berlin, Heidelberg, Apr. 2008, pp. 127–145. DOI: [10.1007/978-3-540-78967-3_8](https://doi.org/10.1007/978-3-540-78967-3_8).
- [Cor02] Jean-Sébastien Coron. “Optimal Security Proofs for PSS and Other Signature Schemes”. In: *EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Vol. 2332. LNCS. Springer, Berlin, Heidelberg, 2002, pp. 272–287. DOI: [10.1007/3-540-46035-7_18](https://doi.org/10.1007/3-540-46035-7_18).

- [CS02] Ronald Cramer and Victor Shoup. “Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption”. In: *EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Vol. 2332. LNCS. Springer, Berlin, Heidelberg, 2002, pp. 45–64. DOI: [10.1007/3-540-46035-7_4](https://doi.org/10.1007/3-540-46035-7_4).
- [Dam+20] Ivan Damgård, Helene Haagh, Rebekah Mercer, Anca Nitulescu, Claudio Orlandi, and Sophia Yakoubov. “Stronger Security and Constructions of Multi-designated Verifier Signatures”. In: *TCC 2020, Part II*. ed. by Rafael Pass and Krzysztof Pietrzak. Vol. 12551. LNCS. Springer, Cham, Nov. 2020, pp. 229–260. DOI: [10.1007/978-3-030-64378-2_9](https://doi.org/10.1007/978-3-030-64378-2_9).
- [DH76] Whitfield Diffie and Martin E. Hellman. “New Directions in Cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: [10.1109/TIT.1976.1055638](https://doi.org/10.1109/TIT.1976.1055638).
- [Dod+09] Yevgeniy Dodis, Jonathan Katz, Adam Smith, and Shabsi Walfish. “Composability and On-Line Deniability of Authentication”. In: *TCC 2009*. Ed. by Omer Reingold. Vol. 5444. LNCS. Springer, Berlin, Heidelberg, Mar. 2009, pp. 146–162. DOI: [10.1007/978-3-642-00457-5_10](https://doi.org/10.1007/978-3-642-00457-5_10).
- [Döt+23] Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. “Efficient Laconic Cryptography from Learning with Errors”. In: *EUROCRYPT 2023, Part III*. ed. by Carmit Hazay and Martijn Stam. Vol. 14006. LNCS. Springer, Cham, Apr. 2023, pp. 417–446. DOI: [10.1007/978-3-031-30620-4_14](https://doi.org/10.1007/978-3-031-30620-4_14).
- [DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data”. In: *EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. LNCS. Springer, Berlin, Heidelberg, May 2004, pp. 523–540. DOI: [10.1007/978-3-540-24676-3_31](https://doi.org/10.1007/978-3-540-24676-3_31).
- [DXL12] Jintai Ding, Xiang Xie, and Xiaodong Lin. *A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem*. Cryptology ePrint Archive, Report 2012/688. 2012. URL: <https://eprint.iacr.org/2012/688>.
- [FL25] Rune Fiedler and Roman Langrehr. *On Deniable Authentication against Malicious Verifiers*. Cryptology ePrint Archive, Report 2025/470. 2025. URL: <https://eprint.iacr.org/2025/470>.
- [Fre+13] Eduarda S. V. Freire, Dennis Hofheinz, Eike Kiltz, and Kenneth G. Paterson. “Non-Interactive Key Exchange”. In: *PKC 2013*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Vol. 7778. LNCS. Springer, Berlin, Heidelberg, 2013, pp. 254–271. DOI: [10.1007/978-3-642-36362-7_17](https://doi.org/10.1007/978-3-642-36362-7_17).
- [Gaj+24] Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. “SWOOSH: Efficient Lattice-Based Non-Interactive Key Exchange”. In: *USENIX Security 2024*. Ed. by Davide Balzarotti and Wenyan Xu. USENIX Association, Aug. 2024. URL: <https://www.usenix.org/conference/usenixsecurity24/presentation/gajland>.
- [GJK24] Phillip Gajland, Jonas Janneck, and Eike Kiltz. “Ring Signatures for Deniable AKEM: Gandalf’s Fellowship”. In: *CRYPTO 2024, Part I*. ed. by Leonid Reyzin and Douglas Stebila. Vol. 14920. LNCS. Springer, Cham, Aug. 2024, pp. 305–338. DOI: [10.1007/978-3-031-68376-3_10](https://doi.org/10.1007/978-3-031-68376-3_10).
- [Guo+20] Siyao Guo, Pritish Kamath, Alon Rosen, and Katerina Sotiraki. “Limits on the Efficiency of (Ring) LWE Based Non-interactive Key Exchange”. In: *PKC 2020, Part I*. ed. by Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas. Vol. 12110. LNCS. Springer, Cham, May 2020, pp. 374–395. DOI: [10.1007/978-3-030-45374-9_13](https://doi.org/10.1007/978-3-030-45374-9_13).
- [GY24] Rishab Goyal and Saikumar Yadugiri. “Multi-authority Functional Encryption with Bounded Collusions from Standard Assumptions”. In: *TCC 2024, Part III*. ed. by Elette Boyle and Mohammad Mahmoody. Vol. 15366. LNCS. Springer, Cham, Dec. 2024, pp. 3–30. DOI: [10.1007/978-3-031-78020-2_1](https://doi.org/10.1007/978-3-031-78020-2_1).
- [Hes+21] Julia Hesse, Dennis Hofheinz, Lisa Kohl, and Roman Langrehr. “Towards Tight Adaptive Security of Non-interactive Key Exchange”. In: *TCC 2021, Part III*. ed. by Kobbi Nissim

- and Brent Waters. Vol. 13044. LNCS. Springer, Cham, Nov. 2021, pp. 286–316. DOI: [10.1007/978-3-030-90456-2_10](https://doi.org/10.1007/978-3-030-90456-2_10).
- [HHK18] Julia Hesse, Dennis Hofheinz, and Lisa Kohl. “On Tightly Secure Non-Interactive Key Exchange”. In: *CRYPTO 2018, Part II*. ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. LNCS. Springer, Cham, Aug. 2018, pp. 65–94. DOI: [10.1007/978-3-319-96881-0_3](https://doi.org/10.1007/978-3-319-96881-0_3).
- [Hül+21] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Fiona Johanna Weber, and Philip R. Zimmermann. “Post-quantum WireGuard”. In: *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2021, pp. 304–321. DOI: [10.1109/SP40001.2021.00030](https://doi.org/10.1109/SP40001.2021.00030).
- [IS14] Mitsugu Iwamoto and Junji Shikata. “Information Theoretic Security for Encryption Based on Conditional Rényi Entropies”. In: *ICITS 13*. Ed. by Carles Padró. Vol. 8317. LNCS. Springer, Cham, 2014, pp. 103–121. DOI: [10.1007/978-3-319-04268-8_7](https://doi.org/10.1007/978-3-319-04268-8_7).
- [Jin+25] Haoxiang Jin, Feng-Hao Liu, Zhedong Wang, and Dawu Gu. “Discrete Gaussians Modulo Sub-Lattices: New Leftover Hash Lemmas for Discrete Gaussians”. In: *PKC 2025, Part II*. ed. by Tibor Jager and Jiaxin Pan. Vol. 15675. LNCS. Springer, Cham, May 2025, pp. 301–330. DOI: [10.1007/978-3-031-91823-0_10](https://doi.org/10.1007/978-3-031-91823-0_10).
- [KV09] Jonathan Katz and Vinod Vaikuntanathan. “Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices”. In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Berlin, Heidelberg, Dec. 2009, pp. 636–652. DOI: [10.1007/978-3-642-10366-7_37](https://doi.org/10.1007/978-3-642-10366-7_37).
- [KWZ22] Venkata Koppula, Brent Waters, and Mark Zhandry. “Adaptive Multiparty NIKE”. in: *TCC 2022, Part II*. ed. by Eike Kiltz and Vinod Vaikuntanathan. Vol. 13748. LNCS. Springer, Cham, Nov. 2022, pp. 244–273. DOI: [10.1007/978-3-031-22365-5_9](https://doi.org/10.1007/978-3-031-22365-5_9).
- [Lan23] Roman Langrehr. “On the Multi-user Security of LWE-Based NIKE”. in: *TCC 2023, Part IV*. ed. by Guy N. Rothblum and Hoeteck Wee. Vol. 14372. LNCS. Springer, Cham, 2023, pp. 33–62. DOI: [10.1007/978-3-031-48624-1_2](https://doi.org/10.1007/978-3-031-48624-1_2).
- [LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “On Ideal Lattices and Learning with Errors over Rings”. In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Berlin, Heidelberg, 2010, pp. 1–23. DOI: [10.1007/978-3-642-13190-5_1](https://doi.org/10.1007/978-3-642-13190-5_1).
- [LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “A Toolkit for Ring-LWE Cryptography”. In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Berlin, Heidelberg, May 2013, pp. 35–54. DOI: [10.1007/978-3-642-38348-9_3](https://doi.org/10.1007/978-3-642-38348-9_3).
- [LS15] Adeline Langlois and Damien Stehlé. “Worst-case to average-case reductions for module lattices”. In: *DCC 75.3* (2015), pp. 565–599. DOI: [10.1007/s10623-014-9938-4](https://doi.org/10.1007/s10623-014-9938-4).
- [LW20] Feng-Hao Liu and Zhedong Wang. “Rounding in the Rings”. In: *CRYPTO 2020, Part II*. ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 296–326. DOI: [10.1007/978-3-030-56880-1_11](https://doi.org/10.1007/978-3-030-56880-1_11).
- [Lyu12] Vadim Lyubashevsky. “Lattice Signatures without Trapdoors”. In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Berlin, Heidelberg, Apr. 2012, pp. 738–755. DOI: [10.1007/978-3-642-29011-4_43](https://doi.org/10.1007/978-3-642-29011-4_43).
- [MR04] Daniele Micciancio and Oded Regev. “Worst-Case to Average-Case Reductions Based on Gaussian Measures”. In: *45th FOCS*. IEEE Computer Society Press, Oct. 2004, pp. 372–381. DOI: [10.1109/FOCS.2004.72](https://doi.org/10.1109/FOCS.2004.72).
- [NPS25] Ky Nguyen, David Pointcheval, and Robert Schädlich. “Dynamic Decentralized Functional Encryption: Generic Constructions with Strong Security”. In: *PKC 2025, Part IV*. ed. by Tibor Jager and Jiaxin Pan. Vol. 15677. LNCS. Springer, Cham, May 2025, pp. 344–379. DOI: [10.1007/978-3-031-91829-2_11](https://doi.org/10.1007/978-3-031-91829-2_11).
- [Pei09] Chris Peikert. “Public-key cryptosystems from the worst-case shortest vector problem: extended abstract”. In: *41st ACM STOC*. ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 333–342. DOI: [10.1145/1536414.1536461](https://doi.org/10.1145/1536414.1536461).

- [Pei14] Chris Peikert. “Lattice Cryptography for the Internet”. In: *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*. Ed. by Michele Mosca. Springer, Cham, Oct. 2014, pp. 197–219. DOI: [10.1007/978-3-319-11659-4_12](https://doi.org/10.1007/978-3-319-11659-4_12).
- [Rao14] Vanishree Rao. *Adaptive Multiparty Non-interactive Key Exchange Without Setup In The Standard Model*. Cryptology ePrint Archive, Report 2014/910. 2014. URL: <https://eprint.iacr.org/2014/910>.
- [Reg05] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *37th ACM STOC*. ed. by Harold N. Gabow and Ronald Fagin. ACM Press, May 2005, pp. 84–93. DOI: [10.1145/1060590.1060603](https://doi.org/10.1145/1060590.1060603).
- [Sch+22] Peter Schwabe et al. *CRYSTALS-KYBER*. tech. rep. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>. National Institute of Standards and Technology, 2022.
- [SSW20] Peter Schwabe, Douglas Stebila, and Thom Wiggers. “Post-Quantum TLS Without Handshake Signatures”. In: *ACM CCS 2020*. Ed. by Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna. ACM Press, Nov. 2020, pp. 1461–1480. DOI: [10.1145/3372297.3423350](https://doi.org/10.1145/3372297.3423350).
- [Tra25] Si An Oliver Tran. “Lower bounds for NIKE with non-perfect correctness”. Available at <https://doi.org/10.3929/ethz-b-000731885>. Master’s thesis. Zurich, Switzerland: ETH Zurich, Apr. 2025.
- [TV23] B  n  dikt Tran and Serge Vaudenay. “A Gapless Post-quantum Hash Proof System in the Hamming Metric”. In: *ACNS 2023, Part I*. ed. by Mehdi Tibouchi and Xiaofeng Wang. Vol. 13905. LNCS. Springer, Cham, June 2023, pp. 664–694. DOI: [10.1007/978-3-031-33488-7_25](https://doi.org/10.1007/978-3-031-33488-7_25).
- [ZY17] Jiang Zhang and Yu Yu. “Two-Round PAKE from Approximate SPH and Instantiations from Lattices”. In: *ASIACRYPT 2017, Part III*. ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10626. LNCS. Springer, Cham, Dec. 2017, pp. 37–67. DOI: [10.1007/978-3-319-70700-6_2](https://doi.org/10.1007/978-3-319-70700-6_2).

A Additional preliminaries⁹

A.1 (Conditional) Rényi entropy

There are several ways to define conditional Rényi entropy, see [IS14] for an overview. For this work we will use the following definitions for conditional collision and min-entropy:

Definition A.1 (Conditional min-entropy). The average min-entropy of a random variable X with finite domain $\mathcal{X}$ conditioned on a (correlated) random variable Z is defined as

$$H_\infty(X | Z) := -\log \left(\mathbb{E}_{z \leftarrow Z} \left[\max_{x \in \mathcal{X}} \Pr[X = x | Z = z] \right] \right).$$

Definition A.2 (Conditional collision entropy). The average collision entropy of a random variable X with finite domain $\mathcal{X}$ conditioned on a (correlated) random variable Z is defined as

$$H_2(X | Z) := -\log \left(\mathbb{E}_{z \leftarrow Z} \left[\sum_{x \in \mathcal{X}} \Pr[X = x | Z = z]^2 \right] \right).$$

We write $H_\infty(X)$ and $H_2(X)$ for the variants of the above definitions without conditioning on Z .

We will make use of the following simplistic form of a chain rule.

Lemma A.3 ([DRS04, Lemma 2.2]). *Let X, Y, Z be (possibly correlated) random variable on $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively. Then*

$$H_\infty(X | (Y, Z)) \geq H_\infty((X, Y) | Z) - \log |\mathcal{Y}|$$

and in particular

$$H_\infty(X | Y) \geq H_\infty((X, Y)) - \log |\mathcal{Y}|.$$

In [DRS04] this is stated and proven for the special case where $|\mathcal{Y}|$ is a power of two, but the proof trivially generalizes to any finite size set $\mathcal{Y}$.

Lemma A.4 *For random variables X (with finite domain $\mathcal{X}$) and Z we have $H_\infty(X | Z) \leq H_2(X | Z)$*

Proof. Fix z in the support of Z and let $X_{|Z=z}$ be a random variable distributed according to the marginal distribution of X conditioned on $Z = z$. Then

$$\max_{x \in \mathcal{X}} \Pr[X_{|Z=z} = x] = \left(\sum_{x \in \mathcal{X}} \Pr[X_{|Z=z} = x] \right) \max_{x \in \mathcal{X}} \Pr[X_{|Z=z} = x] \geq \sum_{x \in \mathcal{X}} \Pr[X_{|Z=z} = x]^2.$$

By taking $-\log(\mathbb{E}_{z \leftarrow Z}[\cdot])$ on both sides we get the statement. $\square$

A.2 Pseudorandom function (PRF)

Definition A.5 (PRF). A PRF for domain $\mathcal{D}$ and range $\mathcal{E}$ consists of two PPT algorithms

- $\text{Gen}(1^\lambda)$ inputs the unary encoded security parameter and outputs a PRF key k and
- $\text{PRF}(k, x)$ inputs a PRF key k and $x \in \mathcal{D}$ and outputs $y \in \mathcal{E}$.

Definition A.6 (Pseudorandomness of PRFs). A PRF is *pseudorandom* if for every PPT adversary $\mathcal{A}$

$$\text{Adv}_{\mathcal{A}, \text{PRF}}^{\text{prf}}(\lambda) := \left| \Pr_{k \leftarrow \text{Gen}(1^\lambda)} [\mathcal{A}^{\text{PRF}(k, \cdot)}(1^\lambda) = 1] - \Pr_{\text{RF} \xleftarrow{\$} \mathcal{E}^{\mathcal{D}}} [\mathcal{A}^{\text{RF}(\cdot)}(1^\lambda) = 1] \right|.$$

Here, $\mathcal{E}^{\mathcal{D}}$ denotes the set of all functions from $\mathcal{D}$ to $\mathcal{E}$.

A.3 Cyclotomic rings

We recall some basics about cyclotomic number fields and its ring of algebraic integers. For more details we refer the reader to [LPR13].

The ℓ -th cyclotomic number field is the field $\mathbb{Q}(\zeta_\ell)$, where ζ_ℓ is an element of order ℓ (i.e. an ℓ -th root of unity). The minimal polynomial of ζ_ℓ has degree $d := \phi(\ell)$ (where ϕ is Euler's totient function), which we also call the degree of $\mathbb{Q}(\zeta_\ell)$.

Geometry. To define a geometry on number fields, we use the canonical embedding. Every number field $\mathbb{Q}(\zeta)$ of degree d has d embeddings (injective field homomorphisms) in $\mathbb{C}$ denoted by $\sigma_i : \mathbb{Q}(\zeta) \rightarrow \mathbb{C}$, each of them sending ζ to one of the d primitive ℓ -th roots of unity in $\mathbb{C}$. With them, we can construct a ring-homomorphism (called the canonical embedding)

$$\begin{aligned} \sigma : \mathbb{Q}(\zeta) &\rightarrow H := \{(x_1, \dots, x_d) \in \mathbb{R}^{s_1} \times \mathbb{C}^{s_2} \mid \forall i \in [s_2] : x_{s_1+i} = \overline{x_{s_1+s_2+i}}\} \\ x &\mapsto (\sigma_1(x), \dots, \sigma_d(x)) \end{aligned}$$

by ordering these embeddings appropriately and setting $s_1, s_2 \in \mathbb{N}_0$ satisfying $d = s_1 + 2s_2$ appropriately.

As inner-product space, H is isomorphic to $\mathbb{R}^d$, i.e. there exists an isomorphism $\Theta : H \rightarrow \mathbb{R}^d$. With this we get the embedding $\psi := (\Theta \circ \sigma)$. This embedding induces a norm on ring elements $\mathbf{r} \in \mathbb{Q}(\zeta)$: For $p \in \mathbb{N}_+ \cup \{\infty\}$ we define $\|\mathbf{r}\|_p := \|\psi(\mathbf{r})\|_p$. This norm can be extended to vectors $\mathbf{r} \in \mathbb{Q}(\zeta)^n$ in the canonical way.

There are different methods to induce a geometry on $\mathbb{R}^d$, for example by using the coefficient embedding. Our results also hold for different geometries when taking into account ring-expansion factors [LPR10].

Ring of integers. The ring of algebraic integers of a cyclotomic field $\mathbb{Q}(\zeta_\ell)$ is $\mathbb{Z}[\zeta_\ell]$, and we call this (the ℓ -th) cyclotomic ring for brevity. We call $d := \phi(\ell)$ the degree of the ring.

Ideals. The norm of an ideal $\mathcal{I}$ of a $\mathcal{R}$ is $\mathcal{N}(\mathcal{I}) := |\mathcal{R}/\mathcal{I}|$ (i.e. the index of $\mathcal{I}$ as additive subgroup of $\mathcal{R}$).

In $\mathbb{Z}[\zeta_\ell]$ every ideal can be expressed uniquely as a product of prime ideals. For an integer prime $q \in \mathbb{Z}$ we give the explicit prime factorization of the principal ideal $\langle q \rangle$. Therefore, let $\alpha \in \mathbb{N}_0$ be the largest integer with $q^\alpha \mid \ell$, $h := \phi(q^\alpha)$, and f be the multiplicative order of q in $\mathbb{Z}_{\ell/q^\alpha}$. Then $\langle q \rangle = \mathfrak{p}_1^h \cdots \mathfrak{p}_g^h$ where $g = d/(hf)$ and $\mathfrak{p}_1, \dots, \mathfrak{p}_g$ are distinct prime ideals with $\mathcal{N}(\mathfrak{p}_i) = q^f$ for all $i \in [g]$.

An important special case is $q = 1 \pmod{\ell}$. In this case the ideal $\langle q \rangle$ factors into d distinct prime ideals, each with norm q (“fully-splitting ring”).

For our NIKE construction the following Lemma about the probability of hitting an invertible element in $\mathcal{R}_q$ is useful.

Lemma A.7 *Let $\mathcal{R}$ be a cyclotomic ring, q a prime and $\langle q \rangle = \mathfrak{p}_1^h \cdots \mathfrak{p}_g^h$ the prime factorization of $\langle q \rangle$. For $n \in \mathbb{N}_+$ we have*

$$\Pr_{r_1, \dots, r_n \xleftarrow{\$} \mathcal{R}_q} [\exists i \in [n] : r_i \in \mathcal{R}_q^\times := \{r \in \mathcal{R}_q \mid \exists r^{-1} \in \mathcal{R}_q : r \cdot r^{-1} = 1\}] \geq 1 - (gq^{-f})^n,$$

where $f := [\mathcal{R}/\mathfrak{p} : \mathbb{F}_q]$ is the residue degree. For fully-splitting rings we have $f = 1$.

Proof. By the Chinese Remainder Theorem $\mathcal{R}_q$ is isomorphic to $\prod_{i=1}^g \mathcal{R}/\mathfrak{p}_i^h$. For the next step, fix i .

Next, we argue that $\mathcal{R}/\mathfrak{p}_i^h$ has the unique maximal ideal $\mathfrak{p}/\mathfrak{p}_i^h$. Each ideal $\tilde{I} \subseteq \mathcal{R}/\mathfrak{p}_i^h$ corresponds to an ideal $I \subseteq \mathcal{R}$ with $\mathfrak{p}_i^h \subseteq I$. Since $\mathcal{R}$ is a Dedekind domain, this enforces $I \mid \mathfrak{p}_i^h$ which means $I = \mathfrak{p}^j$ for some $0 \leq j \leq h$. The only maximal ideal among these is $I = \mathfrak{p}$.

Next, we argue that each element $x \notin \mathfrak{p}/\mathfrak{p}_i^h$ is a unit. If $x \notin \mathfrak{p}/\mathfrak{p}_i^h$, then $\langle x \rangle \not\subseteq \mathfrak{p}/\mathfrak{p}_i^h$. Since $\mathfrak{p}/\mathfrak{p}_i^h$ is the only maximal ideal, we have $\langle x \rangle = \mathcal{R}/\mathfrak{p}_i^h$, which means x is a unit. Conversely, each element in $\mathfrak{p}/\mathfrak{p}_i^h$ is not a unit.

Finally, we have to count the size of $\mathfrak{p}/\mathfrak{p}_i^h$ and $\mathcal{R}/\mathfrak{p}_i^h$. Therefore, consider the chain

$$\{0\} = \mathfrak{p}^h/\mathfrak{p}^h \subseteq \mathfrak{p}^{h-1}/\mathfrak{p}^h \subseteq \cdots \subseteq \mathfrak{p}/\mathfrak{p}^h \subseteq \mathcal{R}/\mathfrak{p}^h.$$

In this chain each successive quotient $\mathfrak{p}^k/\mathfrak{p}^{k+1}$ is isomorphic to $\mathcal{R}/\mathfrak{p}$. To see this, pick $x \in \mathfrak{p}^k \setminus \mathfrak{p}^{k+1}$. Then the ideal $\langle x \rangle + \mathfrak{p}^{k+1}$ satisfies $\mathfrak{p}^{k+1} \not\subseteq J \subseteq \mathfrak{p}^k$. Since there is no ideal lying strictly between $\mathfrak{p}^{k+1}$ and $\mathfrak{p}^k$ we must have $J = \mathfrak{p}^k$ which means x generates $\mathfrak{p}^k/\mathfrak{p}^{k+1}$ as $\mathcal{R}$ -module. Thus the $\mathcal{R}$ -linear map $\varphi : \mathcal{R} \rightarrow \mathfrak{p}^k/\mathfrak{p}^{k+1}, r \mapsto [rx]_{\mathfrak{p}^{k+1}}$ is surjective. Moreover, the kernel of φ satisfies $\mathfrak{p} \subseteq \ker \varphi \not\subseteq \mathcal{R}$. Since $\mathfrak{p}$ is a maximal ideal, this forces $\ker \varphi = \mathfrak{p}$. Thus, $\psi : \mathcal{R}/\mathfrak{p} \rightarrow \mathfrak{p}^k/\mathfrak{p}^{k+1}, [r]_{\mathfrak{p}} \mapsto [rx]_{\mathfrak{p}^{k+1}}$ is the desired isomorphism.

We have $|\mathcal{R}/\mathfrak{p}| = q^f$ where $f := [\mathcal{R}/\mathfrak{p} : \mathbb{F}_q]$. The index of $\mathfrak{p}^{k+1}/\mathfrak{p}^h$ in $\mathfrak{p}^k/\mathfrak{p}^h$ is $|\mathfrak{p}^k/\mathfrak{p}^{k+1}| = q^f$. Starting with $|\mathfrak{p}^h/\mathfrak{p}^h| = 1$ we can apply Lagrange’s theorem to get the cardinality of each element in the chain. Concretely, we get $|\mathfrak{p}^k/\mathfrak{p}^h| = q^{(h-k)f}$.

To put things together, we have

$$\Pr_{r \leftarrow \mathbb{S}/\mathcal{R}/\mathfrak{p}_i^h} [r \in (\mathcal{R}/\mathfrak{p}_i^h)^\times] = 1 - |\mathfrak{p}/\mathfrak{p}_i^h| |\mathcal{R}/\mathfrak{p}_i^h| = 1 - q^{hf} q^{(h-1)f} = 1 - q^{-f}.$$

With the Chinese Remainder Theorem we get

$$\Pr_{r \leftarrow \mathbb{S}/\mathcal{R}_q} [r \in \mathcal{R}_q^\times] = \prod_{i=1}^g (1 - q^{-f}) = (1 - q^{-f})g \geq 1 - gq^{-f},$$

where the last inequality follows from Bernoulli's inequality.

Using that $r_1, \dots, r_n$ are sampled independently, the statement follows. $\square$

A.4 Linear algebra

Every real matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$ can be written as $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{V} \in \mathbb{R}^{m \times m}$ is a orthogonal matrix and $\mathbf{D} \in \mathbb{R}^{n \times m}$ is an upper diagonal matrix (singular value decomposition). The entries of $\mathbf{D}$ are called the singular values of $\mathbf{M}$ and we denote the smallest singular value by $\sigma_{\min}(\mathbf{M})$ and the largest singular value by $\sigma_{\max}(\mathbf{M})$. The largest singular value $\sigma_{\max}(\mathbf{M})$ is equal to the Euclidean spectral norm $\|\mathbf{M}\|_2 := \max_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2$.

If $\Sigma \in \mathbb{R}^{n \times n}$ is a symmetric positive semi-definite matrix, its singular value decomposition is of the form $\Sigma = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with non-negative entries. Let $\sqrt{\mathbf{D}} \in \mathbb{R}^{n \times n}$ be the matrix obtained by applying the square root function component-wise to all entries of $\mathbf{D}$. With this, we define $\sqrt{\Sigma} = \mathbf{U}\sqrt{\mathbf{D}}\mathbf{U}^\top$.

We will use the following bound for the largest singular value of a short matrix.

Lemma A.8 ([Lan23, Lemma 2.2]). *Let $\mathbf{M} \in [-B, B]^{n \times m}$ for $B > 0$. Then $\sigma_{\max}(\mathbf{M}) \leq B\sqrt{n}$.*

A.5 Discrete Gaussian distribution

Let $\Sigma \in \mathbb{R}^{n \times n}$ be a symmetric positive semi-definite matrix. Then the *Gaussian function* on $\mathbb{R}^n$ is defined as $\rho_{\sqrt{\Sigma}}(\mathbf{x}) := \exp(-\pi \mathbf{x}^\top \Sigma^{-1} \mathbf{x})$. The function extends to sets in the usual way. That is, for any countable set $A \subseteq \mathbb{R}^n$, $\rho_{\sqrt{\Sigma}}(A) := \sum_{\mathbf{x} \in A} \rho_{\sqrt{\Sigma}}(\mathbf{x})$.

Moreover, for every countable set $A \subseteq \mathbb{R}^n$ and any $\mathbf{x} \in A$, the *discrete Gaussian function* is defined by $\rho_{A, \sqrt{\Sigma}}(\mathbf{x}) := \rho_{\sqrt{\Sigma}}(\mathbf{x}) / \rho_{\sqrt{\Sigma}}(A)$ and we denote the corresponding *discrete Gaussian distribution* as $\mathcal{D}_{A, \sqrt{\Sigma}}$.

For cyclotomic rings, we write $\mathcal{D}_{\mathcal{R}^n, \sqrt{\Sigma}}$ for the distribution that samples $\mathbf{r} \in \mathcal{R}^n$ with probability $\rho_{\psi(\mathcal{R})^n, \sqrt{\Sigma}}(\psi(\mathbf{r}))$, where ψ is the canonical embedding.

If $\Sigma = \sigma^2 \cdot \mathbf{I}_n$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix, we denote the Gaussian function by ρ_σ , the discrete Gaussian function by $\rho_{A, \sigma}$ and the discrete Gaussian distribution by $\mathcal{D}_{A, \sigma}$ for short.

We use the following tail-bound for (possibly non-spherical) discrete Gaussian distributions.

Lemma A.9 ([Lan23, Lemma 2.4], based on [Lyu12, Lemma 4.4 (3)]). *For any $k > 1$, $n \in \mathbb{N}_+$ and any symmetric positive semi-definite matrix $\Sigma \in \mathbb{R}^{n \times n}$*

$$\Pr_{\mathbf{z} \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sqrt{\Sigma}}} [\|\mathbf{z}\|_2 > k\sqrt{\sigma_{\max}(\Sigma)n/2\pi}] < k^n e^{n(1-k^2)/2}.$$

A.6 Lattices

A lattice $\Lambda \subseteq \mathbb{R}^n$ is a set of all integer linear combinations of a set of k linearly independent vectors of $\mathbb{R}^n$, the basis of Λ . We call k the rank of the lattice. The dual of a lattice Λ is $\Lambda^* := \{\mathbf{w} \in \mathbb{R}^n \mid \forall \mathbf{v} \in \Lambda : \langle \mathbf{v}, \mathbf{w} \rangle \in \mathbb{Z}\}$.

We recall the definition of the smoothing parameter for lattices.

Definition A.10 ([MR04]). Let Λ be a lattice, and $\varepsilon > 0$. Then $\eta_\varepsilon(\Lambda) := \min\{s > 0 \mid \rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \varepsilon\}$.

For an invertible matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ and a lattice $\Lambda \subseteq \mathbb{R}^n$, we write $\mathbf{M} \geq \eta_\varepsilon(\Lambda)$ iff $1 \geq \eta_\varepsilon(\Lambda \mathbf{M}^{-1})$.

A.7 (Module) learning with errors

We will use the module learning with errors problem (M-LWE) in this work. This includes the interesting special cases of unstructured LWE and ring LWE (R-LWE).

Definition A.11 (M-LWE assumption). The $(\mathcal{R}, n, m, q, \sqrt{\Sigma})$ -LWE assumption for a ring $\mathcal{R}$ of degree d , $n, m, q \in \mathbb{N}_+$, the symmetric positive semi-definite error distribution matrix $\sqrt{\Sigma} \in \mathbb{R}^{m \times m}$ on $\mathcal{R}_q^m$ (that all implicitly depend on the security parameter) states that for every PPT adversary $\mathcal{A}$,

$$\text{Adv}_{\mathcal{R}, n, m, q, \sqrt{\Sigma}}^{\text{lwe}}(\mathcal{A}) := \Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top) - \mathcal{A}(\mathbf{A}, \mathbf{y})]$$

is negligible when the probability is taken over $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathcal{R}_q^n$, $\mathbf{y} \xleftarrow{\$} \mathcal{R}_q^{1 \times m}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$, and the internal randomness of $\mathcal{A}$.

Unstructured LWE. The case of $d = 1$ (where, without loss of generality $\mathcal{R} = \mathbb{Z}$) is called (unstructured) LWE. In this setting, the hardness of the problem is essentially controlled by n . The assumption was introduced by [Reg05], who showed that for $\sqrt{\Sigma} = \sigma \mathbf{I}_m$ with $\sigma > \max\{2\sqrt{n}, q/2^{n^c}\}$ for a constant $c \in (0, 1)$ the hardness of LWE can be reduced quantumly to hard worst-case lattice problems, namely approximating the decision version of the shortest vector problem (GapSVP) and the shortest independent vector problem (SIVP) within $\tilde{O}(n2^{n^c})$. Later works also gave classical reductions for $q \geq 2^{n/2}$ consisting of small primes [Pei09] and polynomial moduli [Bra+13].

This gives strong indication that the LWE assumption holds for both polynomial and super-polynomial modulus(q)-to-noise(σ) ratios, while the former is preferable because it has qualitatively better reductions to lattice problems and arithmetic with polynomial moduli is more efficient than with super-polynomial moduli.

Ring & module LWE. Ring LWE refers to the special case where $n = 1$. In this setting, the hardness of the problem is essentially controlled by the degree d of the ring $\mathcal{R}$. The problem was introduced in [LPR10; LPR13] to improve the concrete efficiency of LWE-based schemes.

Module LWE was introduced in [BGV12; LS15] and generalizes both ring and unstructured LWE. In this setting, the hardness of the problem is essentially controlled by the product dn .

A.8 Noisy hints

We recall (a special case of) a theorem of [Döt+23] showing that noisy hints about the LWE error can be simulated, for suitable parameters, without knowing the error (at the cost of increasing the error size).

Lemma A.12 ([Döt+23, Theorem 2], simplified version of [Lan23]). *Let $\mathcal{R}$ be a ring of degree d , $m, k \in \mathbb{N}_+$, $\mathbf{Z} \in \mathbb{R}^{k \times m}$ and $\varepsilon > 0$. Let $\Sigma_0 \in \mathbb{R}^{dm \times dm}$, $\mathbf{T}_0 \in \mathbb{R}^{dk \times dk}$ and $s, t \geq 2\sqrt{2}$ such that*

$$\begin{aligned} \sqrt{\Sigma_0} &\geq \eta_\varepsilon(\mathcal{R}^m), & \sqrt{\mathbf{T}_0} &\geq \eta_\varepsilon(\mathcal{R}^k), \text{ and} \\ t^2 \sigma_{\min}(\mathbf{T}_0) &\geq (\mathfrak{s}^2 + 1)(\mathfrak{s}^2 + 2)s^2 \sigma_{\max}(\Sigma_0) \sigma_{\max}(\psi(\mathbf{Z}))^2. \end{aligned}$$

Then, for $\Sigma := (s^2 + 1)\Sigma_0$, $\mathbf{T} := (t^2 + 1)\mathbf{T}_0$, and $\bar{\Sigma} := s^2/4 \cdot \Sigma_0$ there exists an efficiently sampleable distribution $\mathcal{F}$ on $\mathcal{R}^m \times \mathcal{R}^k$ such that for $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$, $\mathbf{e}_2 \leftarrow \mathcal{D}_{\mathcal{R}^k, \sqrt{\mathbf{T}}}$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\bar{\Sigma}}}$, and $(\mathbf{f}_1, \mathbf{f}_2) \leftarrow \mathcal{F}$ we have

$$\text{SD}((\mathbf{e}_1, \mathbf{Z}\mathbf{e}_1 + \mathbf{e}_2), (\mathbf{e} + \mathbf{f}_1, \mathbf{f}_2)) \leq 22\varepsilon.$$

B A tighter LWE-based NIKE. (Full details)

B.1 The rounding function

We need a rounding function to reconcile the shared keys Alice and Bob receive, that are close to each other but unlikely to be the exact same value. The unrounded shared key is an element in $\mathcal{R}_q$, and we can use every coefficient of $\mathcal{R}_q$ for rounding. It might be necessary to select a ring degree that is larger than the desired number of bits in the shared key to achieve the desired security level. Therefore, we introduce

a parameter $d' \leq d$ that controls how many coefficients are used to derive a shared key. It might also be desired to use a ring degree that is smaller than the number of bits, for example when using unstructured LWE (where $\rho = 1$). Therefore, we introduce the parameter $\rho \in \mathbb{N}_+$ controlling how many shared key bits we are extracting from each coefficient. Of course, we need $2^\rho \ll q$ and for fast arithmetic operations it seems best to set ρ to a small constant (e.g. to avoid that the modulus q exceeds the machine word length, which is currently typically 64 bits) and use several copies of the NIKE to achieve the desired bit length.

In the setting with polynomial modulus it is also important to ensure that the rounding function does not introduce a bias in the resulting shared key. For example, when we round each coefficient to one bit ($\rho = 1$) and the modulus q is odd, we cannot round $(q-1)/2$ values to 0 and $(q+1)/2$ values 1, since this would slightly favor 1-bits in the resulting shared key. Therefore, we use in this case size $(q-1)/2$ for both “rounding intervals” and map the remaining value to a default value γ , that will be instantiated in the NIKE with a pseudorandom bit.

Let q' be the unique integer with $q' \in [q - 2^\rho + 1, q]$ and $2^\rho \mid q'$. We define the following rounding function (for $\beta \in (\{0, 1\}^\rho)^{d'}$, $\gamma \in \{0, 1\}^\rho$)

$$\begin{aligned} \text{Round}_\beta : \mathcal{R}_q &\rightarrow (\{0, 1\}^\rho)^{d'} \\ r &\mapsto (\text{round}_{\beta_1}(x_1), \dots, \text{round}_{\beta_{d'}}(x_{d'})) \text{ where } \mathbf{x} = \psi(r) \\ \text{round}_\gamma : \{0, \dots, q-1\} &\rightarrow \{0, 1\}^\rho \\ x &\mapsto \begin{cases} \text{Bits}(x \text{ div } \frac{q'}{2^\rho}) & \text{if } x < q' \\ \gamma & \text{otherwise} \end{cases} \end{aligned}$$

where the coefficients of $\mathcal{R}_q$ (under the fixed embedding ψ) are interpreted as integers in $\{0, \dots, q\}$.

In the following, we call $0, \frac{q'}{2^\rho}, \frac{2q'}{2^\rho}, \dots, q'$ the rounding boundaries and for $B_K \in \mathbb{N}_+$ we call $D_{B_K} := \{x \mid \exists k : |x - \frac{kq'}{2^\rho}| < B_K\} \cup [q', q]$ the B_K -danger zone.

The rounding function has the following two properties, that are needed for correctness and security of our lattice-based NIKE construction.

Lemma B.1 *For $r \xleftarrow{\$} \mathcal{R}_q$ and fixed $\beta \in (\{0, 1\}^\rho)^{d'}$, $\text{Round}_\beta(r)$ has statistical distance at most $d'(q - q')/q \leq d' \cdot 2^\rho/q$ to the uniformly random distribution in $(\{0, 1\}^\rho)^{d'}$.*

Proof. For $x \xleftarrow{\$} \mathbb{Z}_q$ and any fixed $\gamma \in \{0, 1\}^\rho$ $\text{round}_\gamma(x)$ is uniformly random in $\{0, 1\}^\rho$. For $x \xleftarrow{\$} \mathbb{Z}_q$, we have $x < q'$ except with probability $(q - q')/q$. Thus, for $x \xleftarrow{\$} \mathbb{Z}_q$ and any fixed $\gamma \in \{0, 1\}^\rho$ $\text{round}_\gamma(x)$ has statistical distance $(q - q')/q$ to uniformly random. Extending this to $\text{Round}_\beta(r)$ yields the lemma. $\square$

Lemma B.2 *For $r \xleftarrow{\$} \mathcal{R}_q$ and $\beta \xleftarrow{\$} (\{0, 1\}^\rho)^{d'}$, $\text{Round}_\beta(r)$ is uniformly random in $(\{0, 1\}^\rho)^{d'}$.*

Proof. For $x \xleftarrow{\$} \mathbb{Z}_q$ and $\gamma \xleftarrow{\$} \{0, 1\}^\rho$ $\text{round}_\gamma(x)$ is uniformly random in $\{0, 1\}^\rho$. If $x < q'$ this can be argued as in Lemma B.1. If $x \geq q'$, this holds because in this case $\text{round}_\gamma(x) = \gamma$ and γ is uniformly random in $\{0, 1\}^\rho$. Extending this to $\text{Round}_\beta(r)$ yields the lemma. $\square$

The following lemma guarantees that when vectors that are within distance B_K are rounded to the same value, when at least one of them is not in the danger zone.

Lemma B.3 *For all $k_L, k_R \in \mathcal{R}_q$ with $\|k_L - k_R\|_\infty \leq B_K$ and $\psi(k_L)_i \notin D_{B_K}$, for all $i \in [d]$, and all $\beta \in \{0, 1\}^\rho$ we have $\text{Round}_\beta(k_L) = \text{Round}_\beta(k_R)$.*

Proof. For $x, x' \in \mathbb{Z}_q$ with $|x - x'| < B_K$ and $x \notin D_{B_K}$ we have $x \text{ div } \frac{q'}{2^\rho} = x' \text{ div } \frac{q'}{2^\rho}$ and thus $\text{round}_\gamma(x) = \text{round}_\gamma(x')$. Extending this to $\text{Round}_\beta(r)$ yields the lemma. $\square$

Lemma B.4 *For all $B_K \in \mathbb{N}_+$ we have $|D_{B_K}| \leq 2^\rho(2B_K + 1)$.*

Proof. The set D_{B_K} contains for each of the 2^ρ rounding boundaries the interval of size B_K to both sides of the boundary and the area of size $q' - q$ where the rounding function returns the default value γ . Thus,

$$|D_{B_K}| \leq 2^\rho \cdot 2B_K + (q' - q) \leq 2^\rho \cdot 2B_K + 2^\rho \leq 2^\rho(2B_K + 1). \quad \square$$

Setup (1^λ): $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$ return $\text{pp} := \mathbf{A}$ KeyGenL ($\text{pp} = \mathbf{A}, \text{id}_L$): $\text{sk}_L \xleftarrow{\$} \mathcal{R}_q^n$; $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$ $\text{pk}_L := \text{sk}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ return $(\text{pk}_L, \text{sk}_L)$ KeyGenR ($\text{pp} = \mathbf{A}, \text{id}_R$): $\text{sk}_R \xleftarrow{\$} \{0, 1\}^m$ $\text{pk}_R := \mathbf{Ask}_R$ return $(\text{pk}_R, \text{sk}_R)$	SharedKeyL ($\text{pp} = \mathbf{A}, \text{id}_R, \text{pk}_R, \text{id}_L, \text{sk}_L$): return $\text{Round}_{(0^\rho)^{d'}}(\text{sk}_L^\top \text{pk}_R)$ SharedKeyR ($\text{pp} = \mathbf{A}, \text{id}_L, \text{pk}_L, \text{id}_R, \text{sk}_R$): return $\text{Round}_{(0^\rho)^{d'}}(\text{pk}_L \text{sk}_R)$
--	---

Figure 9: The new LWE based NIKE NIKE_{sp} we consider in this work in the variant for super-polynomial modulus-to-noise ratio.

B.2 NIKE for super-polynomial modulus-to-noise ratio

For correctness, we need the following parameter requirements. All of these restrictions are needed for security as well.

- The modulus q is a prime.
- If $\mathcal{R} = \mathbb{Z}$, we need $m \geq (n+1)\log(q) + \lambda$.
- If $\mathcal{R}$ is a cyclotomic ring other than $\mathbb{Z}$, we need $m \geq (n+1) \cdot d \log(q) + g \log(h+1) + \lambda$, where $\langle q \rangle = \mathfrak{p}_1^h \cdots \mathfrak{p}_g^h$ with $\mathcal{N}(\mathfrak{p}_i) = q^f$ is the prime ideal factorization of $\langle q \rangle$ (see Section A.3)

Theorem B.5 (Correctness). *The NIKE scheme presented in Figure 9 has correctness error*

$$\varepsilon_{\text{corr}}(\lambda) \leq 2^\rho d' (2m \sqrt{md \sigma_{\max}(\Sigma)/\pi} + 1)/q + \sqrt{2}^{md} e^{-md/2} + 2^{-\lambda/2-1} + (gq^{-f})^n$$

Proof. Let $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$, $\mathbf{s}_L \xleftarrow{\$} \mathcal{R}_q^n$; $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$, and $\text{sk}_R \xleftarrow{\$} \{0, 1\}^m$. When a user with left public key $\mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ and a user with right public key $\text{pk}_R := \mathbf{Ask}_R$ compute the shared key between them, the unrounded keys are

$$k_L := \mathbf{s}_L^\top \mathbf{Ask}_R \quad \text{and} \quad k_R := (\mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top) \text{sk}_R = \mathbf{s}_L^\top \mathbf{Ask}_R + \mathbf{e}_L^\top \text{sk}_R$$

and thus the ∞ -norm of their difference is

$$\|k_L - k_R\|_\infty = \|\mathbf{e}_L^\top \text{sk}_R\|_\infty \leq m \cdot \|\mathbf{e}_L\|_\infty \cdot \|\text{sk}_R\|_\infty \stackrel{(*)}{\leq} m \sqrt{md \sigma_{\max}(\Sigma)/\pi} =: B_K,$$

where the inequality marked with $(*)$ holds except with probability at most $\sqrt{2}^{md} e^{-md/2}$ (which is negligible in λ) by applying Lemma A.9 with $k = \sqrt{2}$ to bound $\|\mathbf{e}_L\|_2 \geq \|\mathbf{e}_L\|_\infty$.

Next, we argue that the term $\mathbf{s}_L^\top \mathbf{Ask}_R$ (and therefore also k_L) is close to uniformly random. First, note that the vector $\mathbf{Ask}_R$ is within statistical distance $2^{-\lambda/2-1}$ to a uniformly random vector (using the left-over hash Lemma, see Corollary 4.11 for the case $\mathcal{R} = \mathbb{Z}$ and Corollary B.23 for the general case that $\mathcal{R}$ is a cyclotomic ring). We also have $\Pr[\exists i \in [n] : \mathbf{s}_i \in \mathcal{R}_q^\times] = 1 - (gq^{-f})^n$ by Lemma A.7. The inner product of a fixed vector $\mathbf{s}$ which contains at least one unit and a uniformly random vector is a uniformly random scalar in $\mathbb{Z}_q$. Thus, we get the statistical distance of $\mathbf{s}_L^\top \mathbf{Ask}_R$ to a uniformly random scalar is at most $2^{-\lambda/2-1} + (gq^{-f})^n$.

Assume k_L is uniformly random in $\mathcal{R}_q$. Then, k_L will be in the B_K -danger zone with probability $|D_{B_K}|d'/q \leq 2^\rho d' (2B_K + 1)/q$ by Lemma B.4. If this does not occur, we have $\text{Round}_\beta(k_L) = \text{Round}_\beta(k_R)$ by Lemma B.3.

This shows that a correctness error can only occur with probability

$$\varepsilon_{\text{corr}}(\lambda) \leq |D_{B_K}|d'/q + \sqrt{2}^{md} e^{-md/2} \leq 2^\rho d' (2m \sqrt{md \sigma_{\max}(\Sigma)/\pi} + 1)/q + \sqrt{2}^{md} e^{-md/2} + 2^{-\lambda/2-1} + (gq^{-f})^n. \quad \square$$

For the security proof we have the following requirements on the parameters:

- The modulus q is a prime.
- The $(\mathcal{R}, n, m, \sqrt{\Sigma})$ -LWE assumption holds.
- The correctness error $\varepsilon_{\text{corr}}(\lambda) \leq 2^{\rho+1} d' m \sqrt{m d \sigma_{\max}(\Sigma)} / \pi / q + 2^{-\lambda/2-1} + (g q^{-f})^n$ is negligible.
- If $\mathcal{R} = \mathbb{Z}$, we need $m \geq (n+1) \log(q) + \lambda$.
- If $\mathcal{R}$ is a cyclotomic ring other than $\mathbb{Z}$, we need $m \geq (n+1) \cdot d \log(q) + g \log(h+1) + \lambda$, where $\langle q \rangle = \mathfrak{p}_1^h \cdots \mathfrak{p}_g^h$ with $\mathcal{N}(\mathfrak{p}_i) = q^f$ is the prime ideal factorization of $\langle q \rangle$ (see Section A.3).

Theorem B.6 (Security). *The NIKE NIKE_{sp} in Figure 9 is adaptive HKR secure if all the requirements mentioned for the above parameters are satisfied. More precisely, for every PPT adversary $\mathcal{A}$ against the adaptive HKR security of NIKE there exists a PPT adversary $\mathcal{B}$ (with approximately the same run time) such that*

$$\text{Adv}_{\mathcal{A}, \text{NIKE}}^{\text{adaptive-HKR}}(\lambda) \leq 2N_L \left(\text{Adv}_{\mathcal{R}, n, m, q, \sqrt{\Sigma}}^{\text{lwe}}(\mathcal{B}) + N_R 2\varepsilon_{\text{corr}} + 2^{-\lambda/2-1} + d' \cdot 2^\rho / q \right)$$

where N_L and N_R are the maximum number of left or right users the adversary registers, respectively.

Proof. The proof proceeds via a hybrid argument. The real adaptive HKR game is G_0 .

The first hybrid game G_1 changes all $\mathcal{O}_{\text{revR}}$ queries and the $\mathcal{O}_{\text{testR}}$ query (if the adversary makes an $\mathcal{O}_{\text{testR}}$ query instead of an $\mathcal{O}_{\text{testL}}$ query). These queries are answered in G_1 like an $\mathcal{O}_{\text{revL}}$ or $\mathcal{O}_{\text{testL}}$ query (on the same arguments), respectively.

Lemma B.7 ($G_0 \rightsquigarrow G_1$).

$$|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \Pr[G_1^{\mathcal{A}} \Rightarrow 1]| \leq N_L N_R \varepsilon_{\text{corr}}$$

Proof. The statistical difference between the answer of an $\mathcal{O}_{\text{revR}}(\text{id}_L, \text{id}_R)$ (or $\mathcal{O}_{\text{testR}}(\text{id}_L, \text{id}_R)$) query and an $\mathcal{O}_{\text{revL}}(\text{id}_R, \text{id}_L)$ query (or $\mathcal{O}_{\text{testL}}(\text{id}_R, \text{id}_L)$ query) is bounded by $\varepsilon_{\text{corr}}$. Since there are at most $N_L N_R - 1$ different $\mathcal{O}_{\text{revR}}$ queries and at most one $\mathcal{O}_{\text{testR}}$ queries, the Lemma follows. $\square$

The next hybrid G_2 is identical to G_1 , with the following change: The game samples a random index $i^* \xleftarrow{\$} [N_L]$. If the left user registered in the i^* -th $\mathcal{O}_{\text{regHL}}$ query is the left user used in the $\mathcal{O}_{\text{testR}}$ query, the game proceeds as in G_0 . Otherwise, the game aborts¹³ and the adversary wins with probability $1/2$.

Lemma B.8 ($G_1 \rightsquigarrow G_2$).

$$\left| \Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2} \right| = N_L \left| \Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2} \right|$$

Proof. The index i^* chosen in the game $\Pr[G_2^{\mathcal{A}} \Rightarrow 1]$ is completely independent of the adversary's view. Thus, the probability that the adversary successfully distinguishes a real from a random shared key in the $\mathcal{O}_{\text{testR}}$ query is independent of i^* , and the probability that i^* is the registration index of the left user used for the $\mathcal{O}_{\text{testR}}$ query is $1/N_L$. $\square$

The next hybrid G_3 is identical to G_2 , except that all $\mathcal{O}_{\text{revL}}$ and $\mathcal{O}_{\text{revR}}$ queries as well as the $\mathcal{O}_{\text{testL}}$ or $\mathcal{O}_{\text{testR}}$ query for the left user with index i^* are answered using the right user's secret key, i.e like an $\mathcal{O}_{\text{revR}}$ or $\mathcal{O}_{\text{testR}}$ query in the initial game G_0 .

Lemma B.9 ($G_2 \rightsquigarrow G_3$).

$$|\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1]| \leq N_R \varepsilon_{\text{corr}}$$

Proof. The statistical difference between the answer of an $\mathcal{O}_{\text{revL}}(\text{id}_R, \text{id}_L)$ query and an $\mathcal{O}_{\text{revR}}(\text{id}_L, \text{id}_R)$ query is bounded by $\varepsilon_{\text{corr}}$. Since there are at most N_R different¹⁴ $\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}/\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ queries that are changed for G_3 , the lemma follows. $\square$

¹³We assume that the game aborts if $\mathcal{A}$ makes an $\mathcal{O}_{\text{testR}}$ query with a left user with a different index than i^* or the adversary makes an $\mathcal{O}_{\text{extrL}}$ query for the left user with index i^* .

¹⁴We count here an $\mathcal{O}_{\text{revL}}$ and $\mathcal{O}_{\text{revR}}$ query for the same pair of users as one query, because these queries are answered identically in both games.

The next hybrid game G_4 generates the public key for the i^* -th left user as $\mathbf{pk}_L \stackrel{\$}{\leftarrow} \mathcal{R}_q^{1 \times m}$ (instead of $\mathbf{pk}_L := \mathbf{sk}_L^\top \mathbf{A} + \mathbf{e}_L^\top$). This means of course that there might be no corresponding secret key for $\mathbf{pk}_L$, but this does not matter because in G_3/G_4 the secret key for the i^* -th challenge user is never used anyway (if the adversary makes an $\mathcal{O}_{\text{extrL}}$ query for this user the game aborts and if the adversary makes an $\mathcal{O}_{\text{revL}}$ query for this user, this is answered like an $\mathcal{O}_{\text{revR}}$ query).

Lemma B.10 ($G_3 \rightsquigarrow G_4$). *For every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ with*

$$|\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{\mathcal{R}, n, m, q, \sqrt{\Sigma}}^{\text{lwe}}(\mathcal{B})$$

Proof. Since the secret key of the i^* -th left user is never used ($\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}$ queries never use this key and the game aborts if an $\mathcal{O}_{\text{extrL}}(\text{id}_L^*)$ query is made), the reduction is straightforward: The reduction receives the LWE challenge $(\mathbf{A}, \mathbf{pk}_L^*)$ and uses $\mathbf{A}$ as public parameters and $\mathbf{pk}_L^*$ as the i^* -th left user's public key. The reduction generates all remaining parts of the game honestly. When $\mathbf{pk}_L^* = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$, the reduction is simulating G_3 . When $\mathbf{pk}_L^* \stackrel{\$}{\leftarrow} \mathcal{R}_q^{1 \times m}$, the reduction is simulating G_4 . $\square$

In the final hybrid G_5 , the $\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ query is answered for $b = 0$ by sampling $k^* \stackrel{\$}{\leftarrow} \mathcal{R}_q$ (instead of $k^* := \mathbf{pk}_L^* \mathbf{sk}_R^*$) and returning $\text{Round}_{(0^\rho)^{d'}}(k^*)$.

Lemma B.11 ($G_4 \rightsquigarrow G_5$).

$$|\Pr[G_4^{\mathcal{A}} \Rightarrow 1] - \Pr[G_5^{\mathcal{A}} \Rightarrow 1]| \leq 2^{-\lambda/2-1}$$

Proof. We assume the adversary makes an $\mathcal{O}_{\text{testL}}(\text{id}_R, \text{id}_L)$ or $\mathcal{O}_{\text{testR}}(\text{id}_L, \text{id}_R)$ query where id_L is the identity of the i^* -th challenge user. Otherwise, both games G_4 and G_5 abort and the lemma holds trivially in this case.

Let $\mathbf{sk}_R^*$ be the right challenge user's secret key. This secret key is not used in any $\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}$ query, because these queries are answered in G_4/G_5 using the left user's secret key unless the left user is id_L^* . We assume that the latter case does not occur (and that the adversary does not make an $\mathcal{O}_{\text{extrR}}(\text{id}_R^*)$ query), because then the adversary would lose the game anyway. Thus, the adversary's view can be simulated by using only $\mathbf{pk}_R^* := \mathbf{A} \mathbf{sk}_R^*$ and $\mathbf{pk}_L^* \mathbf{sk}_R^*$. The left-over hash Lemma thus shows that $\mathbf{pk}_L^* \mathbf{sk}_R^*$ is statistically close to uniformly random.

We present the details about the left-over hash Lemma in Section B.5. We can apply here Corollary 4.11 for the case $\mathcal{R} = \mathbb{Z}$ and Corollary B.23 for the general case that $\mathcal{R}$ is a cyclotomic ring to get the statement. $\square$

Lemma B.12

$$\Pr[G_5^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2} + \frac{d' \cdot 2^\rho}{q}$$

Proof. Since in G_5 the unrounded challenge shared key k^* is uniformly random in $\mathcal{R}_q$, by Lemma B.1 the view of the adversary has statistical distance at most $d' \cdot 2^\rho/q$ to a distribution that is statistically independent of the challenge bit b . Thus, the adversary can guess b with probability at most $1/2 + d' \cdot 2^\rho/q$. $\square$

Combining Lemmata B.7–B.12 yields Theorem B.6. $\square$

B.3 Polynomial modulus-to-noise ratio

The NIKE for polynomial modulus-to-noise ratio is shown in Figure 10. The only difference to the NIKE in Figure 9 are the noise added to the pre-rounding shared keys in `SharedKeyL`, the randomized rounding, and the use of a pseudorandom function to make `SharedKeyL` deterministic.

Theorem B.13 (Correctness). *The NIKE scheme presented in Figure 10 has correctness error*

$$\begin{aligned} \varepsilon_{\text{corr}}(\lambda) \leq & 2^\rho d' (2m \sqrt{md \sigma_{\max}(\Sigma)/\pi} + 2 \sqrt{(n+1)d \sigma_{\max}(\mathbf{T})/2\pi} + 1)/q \\ & + \sqrt{2^{md}} e^{-md/2} + \sqrt{n+1}^d e^{-nd/2} + 2^{-\lambda/2-1} + (gq^{-f})^n + \text{Adv}_{\mathcal{A}, \text{PRF}}^{\text{prf}}(\lambda). \end{aligned}$$

Setup (1^λ): $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$ return $\text{pp} := \mathbf{A}$ KeyGenL ($\text{pp} = \mathbf{A}, \text{id}_L$): $\mathbf{s}_L \xleftarrow{\$} \mathcal{R}_q^n$; $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$ $\mathbf{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ $k_{\text{PRF}} \leftarrow \text{GenPRF}(1^\lambda)$ $\text{sk}_L := (\mathbf{s}_L, k_{\text{PRF}})$ return $(\mathbf{pk}_L, \text{sk}_L)$ KeyGenR ($\text{pp} = \mathbf{A}, \text{id}_R$): $\text{sk}_R \xleftarrow{\$} \{0, 1\}^m$ $\mathbf{pk}_R := \mathbf{Ask}_R$ return $(\mathbf{pk}_R, \text{sk}_R)$	SharedKeyL ($\text{pp} = \mathbf{A}, \text{id}_R, \mathbf{pk}_R, \text{id}_L, \text{sk}_L$): parse $\text{sk}_L =: (\mathbf{s}_L, k_{\text{PRF}})$ $r \parallel \beta := \text{PRF}(k_{\text{PRF}}, \text{id}_R)$ $e' \leftarrow_r \mathcal{D}_{\mathcal{R}, \sqrt{\mathbf{T}}}$ return $\text{Round}_\beta(\mathbf{s}_L^\top \mathbf{pk}_R + e')$ SharedKeyR ($\text{pp} = \mathbf{A}, \text{id}_L, \mathbf{pk}_L, \text{id}_R, \text{sk}_R$): $\beta \xleftarrow{\$} (\{0, 1\}^\rho)^{d'}$ return $\text{Round}_\beta(\mathbf{pk}_L \text{sk}_R)$
---	---

Figure 10: The new LWE based NIKE NIKE_p we consider in this work in the variant for polynomial modulus-to-noise ratio.

Proof. For this analysis we assume that the randomness used in the **SharedKeyL** algorithm is truly random and not generated with a PRF. This changes the correctness error at most by $\text{Adv}_{\mathcal{A}, \text{PRF}}^{\text{prf}}(\lambda)$, where $\mathcal{A}$ is an adversary that tries to break the pseudorandomness of PRF as follows: The adversary generates a left and right public/secret key pair (except for the PRF key contained in the left secret key) for arbitrary identities and computes the shared key for these two key pairs in two different ways. Firstly, it uses the left secret key, the right public key, and queries the pseudorandomness game of PRF to retrieve the PRF outputs. Secondly, it computes the shared key using the right secret key and the left public key. The adversary outputs 1 iff the resulting shared keys are equal.

Let $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$, $\mathbf{s}_L \xleftarrow{\$} \mathcal{R}_q^n$; $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$, $\text{sk}_R \xleftarrow{\$} \{0, 1\}^m$, and $e' \leftarrow \mathcal{D}_{\mathcal{R}, \sqrt{\mathbf{T}}}$. When a user with left public key $\mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$ and a user with right public key $\mathbf{pk}_R := \mathbf{Ask}_R$ compute the shared key between them, the unrounded keys are

$$k_L := \mathbf{s}_L^\top \mathbf{Ask}_R + e' \quad \text{and} \quad k_R := (\mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top) \text{sk}_R = \mathbf{s}_L^\top \mathbf{Ask}_R + \mathbf{e}_L^\top \text{sk}_R$$

and thus the ∞ -norm of their difference is

$$\begin{aligned} \|k_L - k_R\|_\infty &= \|\mathbf{e}_L^\top \text{sk}_R + e'\|_\infty \leq m \cdot \|\mathbf{e}_L\|_\infty + |e'| \\ &\stackrel{(*)}{\leq} m \sqrt{md \sigma_{\max}(\Sigma)/\pi} + \sqrt{(n+1)d \sigma_{\max}(\mathbf{T})/2\pi} =: B_K. \end{aligned}$$

where the inequality marked with $(*)$ holds except with probability at most $\sqrt{2}^{md} e^{-md/2} + \sqrt{n+1}^d e^{-nd/2}$ (which is negligible in nd) by applying Lemma A.9 with $k = \sqrt{2}$ to bound $\|\mathbf{e}_L\|_2 \geq \|\mathbf{e}_L\|_\infty$ and applying the same lemma with $k = \sqrt{n+1}$ to bound $|e'|$.

Next, we argue that the term $\mathbf{s}_L^\top \mathbf{Ask}_R$ (and therefore also k_L) is close to uniformly random. First, note that the vector $\mathbf{Ask}_R$ is within statistical distance $2^{-\lambda/2-1}$ to a uniformly random vector (using the left-over hash Lemma, see Corollary 4.11 for the case $\mathcal{R} = \mathbb{Z}$ and Corollary B.23 for the general case that $\mathcal{R}$ is a cyclotomic ring). We also have $\Pr[\exists i \in [n] : \mathbf{s}_i \in \mathcal{R}_q^\times] = 1 - (gq^{-f})^n$ by Lemma A.7. The inner product of a fixed vector $\mathbf{s}$ which contains at least one unit and a uniformly random vector is a uniformly random scalar in $\mathbb{Z}_q$. Thus, we get the statistical distance of $\mathbf{s}_L^\top \mathbf{Ask}_R$ to a uniformly random scalar is at most $2^{-\lambda/2-1} + (gq^{-f})^n$.

Assume k_L is uniformly random in $\mathcal{R}_q$. Then, k_L will be in the B_K -danger zone with probability $|D_{B_K}|d'/q \leq 2^\rho d'(2B_K + 1)/q$ by Lemma B.4. If this does not occur, we have $\text{Round}_\beta(k_L) = \text{Round}_\beta(k_R)$ by Lemma B.3.

This shows that a correctness error can only occur with probability

$$\varepsilon_{\text{corr}}(\lambda) \leq |D_{B_K}|/q + \sqrt{2}^{md} e^{-md/2} + \sqrt{n+1}^d e^{-nd/2} \leq 2^\rho d' (2(m\sqrt{md\sigma_{\max}(\mathbf{\Sigma})/\pi} + \sqrt{(n+1)d\sigma_{\max}(\mathbf{T})/2\pi}) + 1)/q + \sqrt{2}^{md} e^{-md/2} + \sqrt{n+1}^d e^{-nd/2} + 2^{-\lambda/2-1} + (gq^{-f})^n + \text{Adv}_{\mathcal{A}, \text{PRF}}^{\text{prf}}(\lambda). \quad \square$$

The security proof combines the proof strategy from Theorem B.6 with the technique by [Lan23] to handle non-negligible correctness errors. For the security proof we have the following requirements on the parameters:

- q is a prime.
- The $(\mathcal{R}, n, m, \sqrt{\mathbf{\Sigma}})$ -LWE assumption holds.
- $\varepsilon > 0$ is negligible.
- $k_R \geq N_R - 1$ or $k_R \in \Omega((\mu_R + 1) \cdot \text{superlog}(\lambda))$ for a super-logarithmic function superlog and for

$$\mu_R := \frac{(N_R - 1)2^{\rho+1}d'(m\sqrt{md\sigma_{\max}(\mathbf{\Sigma})/\pi} + \sqrt{(n+1)d\sigma_{\max}(\mathbf{T})/2\pi})}{q} \leq N_R \varepsilon_{\text{corr}}(\lambda).$$

- The matrices $\sqrt{\mathbf{\Sigma}} \in \mathbb{R}^{nd \times nd}$, $\mathbf{T} \in \mathbb{R}^{d \times d}$, and $\sqrt{\mathbf{\Sigma}} \in \mathbb{R}^{nd \times nd}$ meet the requirements of Lemma A.12 for every hint matrix $\mathbf{Z} \in \mathcal{R}_q^{k_R \times n}$ with $\|\mathbf{Z}\|_\infty \leq B_s$. That is, there exist $s, t \in \mathbb{R}$ with
 - $s, t \geq 2\sqrt{2}$
 - $\mathbf{\Sigma} = (s^2 + 1)\mathbf{\Sigma}_0$, $\mathbf{T} = (t^2 + 1)\mathbf{T}_0$, and $\bar{\mathbf{\Sigma}} = s^2/4 \cdot \mathbf{\Sigma}_0$
 - $\sqrt{\mathbf{\Sigma}_0} \geq \eta_\varepsilon(\mathcal{R}^n)$ and $\sqrt{\mathbf{T}_0} \geq \eta_\varepsilon(\mathcal{R}_R^k)$
 - $t^2 \sigma_{\min}(\mathbf{T}_0) \geq \frac{(s^2+1)(s^2+2)}{s^2} \sigma_{\max}(\mathbf{\Sigma}_0) B_s^2 k_R$ (using Lemma A.8)
- $k_L \geq N_L - 1$ or $k_L \in \Omega((\mu_L + 1) \cdot \text{superlog}(\lambda))$ for a super-logarithmic function superlog and for

$$\mu_L := \frac{(N_L - 1)2^{\rho+1}d'(m\sqrt{md\sigma_{\max}(\mathbf{\Sigma})/\pi} + \sqrt{(n+1)d\sigma_{\max}(\mathbf{T})/2\pi})}{q} \leq N_L \varepsilon_{\text{corr}}(\lambda).$$

- If $\mathcal{R} = \mathbb{Z}$, we need $m \geq (n + 1 + k_L) \log(q) + \lambda$.
- If $\mathcal{R}$ is a cyclotomic ring other than $\mathbb{Z}$, we need $m \geq (n + 1 + k_L) \cdot d \log(q) + g \log(h + 1) + \lambda$, where $\langle q \rangle = \mathfrak{p}_1^h \cdots \mathfrak{p}_g^h$ with $\mathcal{N}(\mathfrak{p}_i) = q^f$ is the prime ideal factorization of $\langle q \rangle$ (see Section A.3).

Theorem B.14 (Security). *The NIKE NIKE_p in Figure 10 is adaptive HKR secure if all the requirements mentioned for the above parameters are satisfied. More precisely, for every PPT adversary $\mathcal{A}$ against the adaptive HKR security of NIKE there exist PPT adversaries $\mathcal{B}$ and $\mathcal{C}$ (with approximately the same run time) such that*

$$\begin{aligned} \text{Adv}_{\mathcal{A}, \text{NIKE}}^{\text{adaptive-HKR}}(\lambda) &\leq 2N_L \left(\text{Adv}_{\mathcal{R}, n, m, q, \sqrt{\mathbf{\Sigma}}}^{\text{lwe}}(\mathcal{B}) + \text{Adv}_{\mathcal{C}, \text{PRF}}^{\text{prf}}(\lambda) + N_R 2\varepsilon_{\text{corr}} \right. \\ &\quad + N_L \cdot \sqrt{2}^{md} e^{-md/2} + N_R \cdot \sqrt{n+1}^d e^{-nd/2} + 44\varepsilon + 2^{-\lambda/2-1} \\ &\quad + \begin{cases} 0 & \text{if } k \geq N_R - 1 \\ \exp(-\frac{k_R^2/\mu_R - 2k_R + 1/\mu_R}{1+k_R/\mu_R}) & \text{otherwise} \end{cases} \\ &\quad \left. + \begin{cases} 0 & \text{if } k \geq N_L - 1 \\ \exp(-\frac{k_L^2/\mu_L - 2k_L + 1/\mu_L}{1+k_L/\mu_L}) & \text{otherwise} \end{cases} \right) \end{aligned}$$

where N_L and N_R are the maximum number of left or right users the adversary registers, respectively.

Proof. The proof proceeds via a hybrid argument. The real adaptive HKR game is $\mathbf{G}_0$.

The first hybrid game $\mathbf{G}_1$ is identical to $\mathbf{G}_0$, with the following change: The game samples a random index $i^* \xleftarrow{\$} [N_L]$. If the left user registered in the i^* -th $\mathcal{O}_{\text{regHL}}$ query is the left user used in the $\mathcal{O}_{\text{testR}}$ query, the game proceeds as in $\mathbf{G}_0$. Otherwise, the game aborts¹⁵ and the adversary wins with probability $1/2$.

¹⁵We assume that the game aborts if $\mathcal{A}$ makes an $\mathcal{O}_{\text{testR}}$ query with a left user with a different index than i^* or the adversary makes an $\mathcal{O}_{\text{extrL}}$ query for the left user with index i^* .

Lemma B.15 ($G_0 \rightsquigarrow G_1$).

$$\left| \Pr[G_0^A \Rightarrow 1] - \frac{1}{2} \right| = N_L \left| \Pr[G_1^A \Rightarrow 1] - \frac{1}{2} \right|$$

Proof. The index i^* chosen in the game $\Pr[G_1^A \Rightarrow 1]$ is completely independent of the adversary's view. Thus, the probability that the adversary successfully distinguishes a real from a random shared key in the $\mathcal{O}_{\text{testR}}$ query is independent of i^* , and the probability that i^* is the registration index of the left user used for the $\mathcal{O}_{\text{testR}}$ query is $1/N_L$. $\square$

In the next hybrid game G_2 , the game does not sample a PRF key for the left user with index i^* and in all $\mathcal{O}_{\text{revL}}$ queries and the $\mathcal{O}_{\text{testL}}$ query (if the adversary makes such a query) with the i^* -th left user a random function (sampled on-the-fly) is evaluated instead of the PRF.

Lemma B.16 ($G_1 \rightsquigarrow G_2$). *For every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{C}$ with*

$$|\Pr[G_1^A \Rightarrow 1] - \Pr[G_2^A \Rightarrow 1]| \leq \text{Adv}_{\mathcal{C}, \text{PRF}}^{\text{prf}}(\lambda)$$

Proof. Since the game aborts if left user with index i^* is corrupted, its PRF key is never revealed. Thus, an adversary who is able to distinguish G_1 and G_2 can be reduced straightforwardly to an adversary breaking the PRF security game. $\square$

In game G_3 changes all $\mathcal{O}_{\text{revR}}$, $\mathcal{O}_{\text{revL}}$ and possibly the $\mathcal{O}_{\text{testL}}$ query as follows:

1. For $\mathcal{O}_{\text{revR}}$ and $\mathcal{O}_{\text{revL}}$ queries, the pre-rounding shared key is computed using the right user's secret key, if the left user is the i^* -th user. Otherwise, the left user's secret key will be used, but without adding e' to the shared key. If the adversary makes an $\mathcal{O}_{\text{testL}}$ query, the pre-rounding shared key is computed using the right user's secret key.
2. If the query was formulated such that the left user's secret key should be used and the left user secret key was also used in the previous step, we add $e' \leftarrow_r \mathcal{D}_{\mathcal{R}, \sqrt{\mathbf{T}}}$ (where $r||\beta := \text{PRF}(k_{\text{PRF}}, \text{id}_R)$ and k_{PRF} is the PRF key stored in the left user's secret key) to the pre-rounding shared key.
3. If the pre-rounding shared keys are within distance B_K from a rounding boundary (measured in $\|\cdot\|_\infty$), then we adjust them as follows.
 - If the query was formulated such that the left user's secret key should be used, but the right user's secret key was used in the previous step, we add $-\mathbf{e}_L^\top \mathbf{sk}_R + e'$ (where $e' \leftarrow_r \mathcal{D}_{\mathcal{R}, \sqrt{\mathbf{T}}}$, $r||\beta := \text{RF}(\text{id}_R)$ for a random function RF, id_R is the right user's identity, $\mathbf{e}_L$ is the left user's error term, and $\mathbf{sk}_R$ is the right user's secret key) to the pre-rounding shared key.
 - If the query was formulated such that the right user's secret key should be used, but the left user's secret key was used in the previous step, we add $\mathbf{e}_L^\top \mathbf{sk}_R$ (where $\mathbf{e}_L$ is the left user's error term and $\mathbf{sk}_R$ is the right user's secret key) to the pre-rounding shared key.
4. Finally, the rounding function is applied as in the previous hybrid.

Lemma B.17 ($G_2 \rightsquigarrow G_3$).

$$|\Pr[G_2^A \Rightarrow 1] - \Pr[G_3^A \Rightarrow 1]| \leq N_L \cdot \sqrt{2}^{md} e^{-md/2} + N_R \cdot \sqrt{n+1}^d e^{-nd/2}.$$

Proof. Firstly, we assume that all left users' error terms satisfy $\|\mathbf{e}_L\|_\infty \leq \sqrt{md\sigma_{\max}(\mathbf{\Sigma})/\pi}$. This happens for each user independently with probability at least $1 - \sqrt{2}^{md} e^{-md/2}$ by applying Lemma A.9 with $k = \sqrt{2}$. Secondly, we assume for each right user identity id_R that $|e'| \leq \sqrt{(n+1)d\sigma_{\max}(\mathbf{T})/2\pi}$ for $e' \leftarrow_r \mathcal{D}_{\mathcal{R}, \sqrt{\mathbf{T}}}$ and $r||\beta := \text{RF}(\text{id}_R)$. This happens for each user independently with probability at least $1 - \sqrt{n+1}^d e^{-nd/2}$ by applying Lemma A.9 with $k = \sqrt{n+1}$.

We argue that under these assumptions, the games G_2 and G_3 are identical. Firstly, we look at $\mathcal{O}_{\text{revR}}$ queries where the left user is not the i^* -th user. These queries are answered in G_2 using the right user's secret key and in G_3 using the left user's secret key.

If in G_3 the pre-rounding shared key obtained with the left user's secret key is not within distance B_K to a rounding boundary, we will retrieve the same shared key after rounding as when the right user's secret key would be used. Thus, these queries are identical.

If in G_3 the pre-rounding shared key obtained with the left user's secret key is within distance B_K , the game G_3 adds the term $\mathbf{e}_L^\top \mathbf{sk}_R$ to the pre-rounding shared key, which makes it identical to the pre-rounding shared key that would be used in G_2 . Thus, these queries are identical as well.

Next, we look at $\mathcal{O}_{\text{revL}}$ queries where the left user is the i^* -th user. These queries are answered in G_2 using the left user's secret key and in G_3 using the right user's secret key.

If in G_3 the pre-rounding shared key obtained with the right user's secret key is not within distance B_K to a rounding boundary, we will retrieve the same shared key after rounding as when the left user's secret key would be used. Importantly, we use here that the left user uses a random function instead of a PRF to sample e' , and thus we can apply the tail-bound restriction from above here. Thus, these queries are identical.

If in G_3 the pre-rounding shared key obtained with the right user's secret key is within distance B_K , the game G_3 adds the term $-\mathbf{e}_L^\top \mathbf{sk}_R + e'$ to the pre-rounding shared key, which makes it identical to the pre-rounding shared key that would be used in G_2 . Note here that e' is computed using a random function like in G_2 . Thus, these queries are identical as well.

Finally, we look at the test query. If the test query does not use the i^* -th left user, the game aborts in G_2 and G_3 , and thus the games behave identical in this game. Otherwise, if the adversary makes an $\mathcal{O}_{\text{testL}}$ query, we can argue as in the case for $\mathcal{O}_{\text{revL}}$ queries from above. If the adversary makes an $\mathcal{O}_{\text{testR}}$ query, this query is answered identically anyway.

Note that all other queries are identical in both games. $\square$

The next hybrid game G_4 generates the public key for the i^* -th left user as $\mathbf{pk}_L \xleftarrow{\$} \mathcal{R}_q^{1 \times m}$ (instead of $\mathbf{pk}_L := \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$). The game still samples $\mathbf{e}_L \leftarrow \mathcal{D}_{\mathcal{R}^m, \sqrt{\Sigma}}$ to compute the correction terms used to answer $\mathcal{O}_{\text{revL}}$ queries for the i^* -th left user, but this is sampled independently of the public key.

Lemma B.18 ($G_3 \rightsquigarrow G_4$). *For every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ with*

$$\begin{aligned} |\Pr[G_3^{\mathcal{A}} \Rightarrow 1] - \Pr[G_4^{\mathcal{A}} \Rightarrow 1]| &\leq \text{Adv}_{\mathcal{R}, n, m, q, \sqrt{\Sigma}}^{\text{lwe}}(\mathcal{B}) + 44\varepsilon \\ &\quad + \begin{cases} 0 & \text{if } k \geq N_R - 1 \\ \exp(-\frac{k_R^2/\mu_R - 2k_R + 1/\mu_R}{1 + k_R/\mu_R}) & \text{otherwise} \end{cases} \end{aligned}$$

Proof. Firstly, note that since the secret key of the i^* -th left user is never used, except for the terms of the form $-\mathbf{e}_L^\top \mathbf{sk}_R + e'$ (where $e' \leftarrow_r \mathcal{D}_{\mathcal{R}, \sqrt{\mathbf{T}}}$, $r||\beta := \text{RF}(\text{id}_R)$ for a random function RF, id_R is the right user's identity, $\mathbf{e}_L$ is the left challenge user's error term, and $\mathbf{sk}_R$ is any of the right user's secret keys) that are used as correction terms in some of the $\mathcal{O}_{\text{revL}}$ queries and possibly the $\mathcal{O}_{\text{testL}}$ query. On a high level, the reduction will use Lemma A.12 to simulate these terms without actually knowing $\mathbf{e}_L$. Then we can reduce an adversary distinguishing these games to the MLWE assumption.

The reduction receives the LWE challenge $(\mathbf{A}, \mathbf{b})$ and uses $\mathbf{A}$ as public parameters.

The reduction then pre-generates N_R right public/secret key pairs, that will be used to answer the $\mathcal{O}_{\text{regHR}}$ queries of the adversary. For each of these key pairs $(\mathbf{pk}_R, \mathbf{sk}_R)$, the reduction computes the pre-rounding shared key $\mathbf{pk}_L^\top \cdot \mathbf{sk}_R$. If this value is within distance B_K from a rounding boundary, the reduction adds $\mathbf{sk}_R^\top$ as a new row to the initially empty matrix $\mathbf{Z}$. Finally, $\mathbf{Z}$ is padded to an $\mathcal{R}_q^{k_R \times m}$ matrix by adding all-zero rows if $\mathbf{Z}$ has strictly less than k_R rows and aborting if $\mathbf{Z}$ has strictly more than k_R rows.

Now let $\mathcal{F}$ be the distribution from Lemma A.12 when using this lemma with $\sqrt{\Sigma}$ for the distribution of $\mathbf{e}_1$, $\sqrt{\mathbf{T}}$ for the distribution of $\mathbf{e}_2$ and $\sqrt{\Sigma}$ for the distribution of $\mathbf{e}$ and $\mathbf{Z}$ as the hint matrix. The reduction then samples $(\mathbf{f}_1, \mathbf{f}_2) \leftarrow \mathcal{F}$.

In the i^* -th $\mathcal{O}_{\text{regHL}}$ query, the reduction returns $\mathbf{b} + \mathbf{f}_1^\top$ as public key. For each $\mathcal{O}_{\text{regHR}}$ query the reduction returns one of the pre-generated right public keys. The remaining $\mathcal{O}_{\text{regHL}}$ queries as well as all $\mathcal{O}_{\text{extrL}}$ and $\mathcal{O}_{\text{extrR}}$ queries are answered honestly.

For $\mathcal{O}_{\text{revL}}$ queries (or the $\mathcal{O}_{\text{testL}}$ query, if applicable), where the left user is the i^* -th user and the pre-rounding shared key obtained with the right user's secret key is within distance B_K to a rounding boundary, the reduction uses $-(\mathbf{f}_2)_i$ instead of the term $-\mathbf{e}_L^\top \mathbf{sk}_R + e'$ to answer the query, where i is the row index of the right user's secret key in the matrix $\mathbf{Z}$ (note that such an index has to exist by construction of $\mathbf{Z}$). All other $\mathcal{O}_{\text{revL}}$, all $\mathcal{O}_{\text{revR}}$, and the $\mathcal{O}_{\text{testR}}$ (if applicable) query do not need the right challenge user's secret key and are answered honestly by the reduction.

Analysis. First, assuming that the reduction does not abort, note that when the reduction gets a real LWE sample $\mathbf{b} = \mathbf{s}_L^\top \mathbf{A} + \mathbf{e}_L^\top$, the term $-(\mathbf{f}_2)_i$ is within statistical distance 22ε to the term $-\mathbf{e}_L^\top \mathbf{sk}_R + e'$

in each $\mathcal{O}_{\text{revL}}$ query (or the $\mathcal{O}_{\text{testL}}$ query) where this term is used to correct the pre-rounding shared key by Lemma A.12. Thus, the game that the reduction simulates in this case has statistical distance 22ε to the game G_3 .

If $\mathbf{b} \xleftarrow{\$} \mathcal{R}_q^m$, then the distribution of the i^* -th left user's public key is also uniformly random and the reduction is simulating a game that has statistical distance 22ε to the game G_4 .

If $k_R \geq N - 1$, the reduction never aborts. Otherwise, we use that the probability that a right user has a pre-rounding shared key in the danger zone with the i^* -th left user is at most $p := 2^{\rho d'}(2B_K + 1)/q$ and this is independent for each right user. Let X be the random variable denoting the number right users that have a pre-rounding shared key in the danger zone with the i^* -th left user. Then the expected value of X is $\mu_R := N_R/p$. With the Chernoff bound for Bernoulli distributions, we get the following bound on the probability that X exceeds k_R :

$$\Pr[X \geq k_R] \leq \exp\left(-\frac{(k_R/\mu_R - 1)^2 \mu_R}{1 + k_R/\mu_R}\right) = \exp\left(-\frac{k_R^2/\mu_R - 2k_R + 1/\mu_R}{1 + k_R/\mu_R}\right) \quad (1)$$

which is negligible if $k_R \in \Omega((\mu_R + 1) \cdot \text{superlog}(\lambda))$ for a super-logarithmic function superlog . $\square$

In the final hybrid G_5 , the $\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ query is answered for $b = 0$ by sampling $k^* \xleftarrow{\$} \mathcal{R}_q$ (instead of $k^* := \text{pk}_L^* \text{sk}_R^*$) and returning $\text{Round}_\beta(k^*)$ (for $\beta \xleftarrow{\$} (\{0, 1\}^\rho)^{d'}$).

Lemma B.19 ($\mathsf{G}_4 \rightsquigarrow \mathsf{G}_5$).

$$|\Pr[\mathsf{G}_4^A \Rightarrow 1] - \Pr[\mathsf{G}_5^A \Rightarrow 1]| \leq 2^{-\lambda/2-1} + \begin{cases} 0 & \text{if } k \geq N_L - 1 \\ \exp\left(-\frac{k_L^2/\mu_L - 2k_L + 1/\mu_L}{1 + k_L/\mu_L}\right) & \text{otherwise} \end{cases}$$

Proof. We assume the adversary makes an $\mathcal{O}_{\text{testL}}(\text{id}_R, \text{id}_L)$ or $\mathcal{O}_{\text{testR}}(\text{id}_L, \text{id}_R)$ query where id_L is the identity of the i^* -th challenge user. Otherwise, both games G_4 and G_5 abort and the lemma holds trivially in this case.

Let sk_R^* be the right challenge user's secret key. This secret key sk_R^* is only used to compute the terms $\mathbf{e}_L^\top \text{sk}_R^*$ in $\mathcal{O}_{\text{revR}}$ queries with the right challenge user (and the $\mathcal{O}_{\text{testL}}$ query, if applicable), if the pre-rounding shared key computed using the left secret key (or the right secret key in the $\mathcal{O}_{\text{testL}}$ query) is within distance B_K from a rounding boundary. By applying the Chernoff bound analogous to the previous lemma, we get that this happens for at most k_L different queries except with probability $\exp\left(-\frac{k_L^2/\mu_L - 2k_L + 1/\mu_L}{1 + k_L/\mu_L}\right)$ with $\mu_L := N_L/p$. We assume from now on this occurs. Let $\mathbf{Z}$ be the matrix that contains all the left user's error vectors as rows, for which the correction term $\mathbf{e}_L^\top \text{sk}_R^*$ is needed. Then $\text{aux} := \mathbf{Z} \text{sk}_R^*$ can be described with $k_L d \log(q)$ bits.

We assume that the adversary does not make an $\mathcal{O}_{\text{extrR}}(\text{id}_R^*)$ query, because then the adversary would lose the game anyway. Thus, the adversary's view can be simulated by using only $\text{pk}_R^* := \mathbf{A} \text{sk}_R^*$, aux and $\text{pk}_L^* \text{sk}_R^*$. The left-over hash Lemma thus shows that $\text{pk}_L^* \text{sk}_R^*$ is statistically close to uniformly random.

We present the details about the left-over hash Lemma in Section B.5. We can apply here Corollary 4.11 for the case $\mathcal{R} = \mathbb{Z}$ and Corollary B.23 for the general case that $\mathcal{R}$ is a cyclotomic ring to get the statement. $\square$

Lemma B.20

$$\Pr[\mathsf{G}_5^A \Rightarrow 1] = \frac{1}{2}$$

Proof. In G_5 the view of the adversary is independent of the bit b it is trying to guess by Lemma B.2. $\square$

Combining Lemmata B.15–B.20 yields Theorem B.14. $\square$

B.4 Extension to multi-challenge security

Our security proof can be extended to the multi-challenge setting, where the adversary can make as many $\mathcal{O}_{\text{testL}}$ and $\mathcal{O}_{\text{testR}}$ queries as he wants, without increasing the security loss. We give here a sketch of the proof strategy. The same proof strategy can also be used for [HHK18] to extend their result to the multi-challenge setting.

The proof proceeds via $N_L + 1$ hybrids $G_0, \dots, G_{N_L}$. The hybrid G_i works like the multi-challenge variant of the adaptive HKR security game, except that all $\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ return always a random shared key (independent of the challenge bit) if the left user was registered in the j -th left registration query with $j \leq i$. G_0 is then the initial game and in G_{N_L} the view of the adversary is statistically independent of the challenge bit.

The transition from hybrid G_{i-1} to hybrid G_i is essentially the same as the reduction in the single-challenge case, except that the reduction does not guess an index i^* . Instead, it only flips a coin $c \xleftarrow{\$} \{0, 1\}$ which indicates if the i -th left user will be corrupted or can appear in a challenge query (we can assume without loss of generality that at least one of these events always occurs). This guess will be correct with probability $1/2$.

If $c = 0$, the reduction expects the i -th left user to be corrupted and it just plays the game honestly. Since in this case G_{i-1} and G_i are identical, this “reduction” (which actually reduces to nothing) successfully switches from G_{i-1} to G_i .

If $c = 1$, the reduction expects the i -th left challenge user to not be corrupted. In this case, the reduction runs through several intermediate hybrids as outlined above in the single-challenge case. More specifically, it embeds in the i -th left user’s public key an LWE challenge to switch it to uniformly random. Then we can argue that all shared keys between this left user and any of the right users are statistically close to uniform, which allows us to switch all $\mathcal{O}_{\text{testL}}$ and $\mathcal{O}_{\text{testR}}$ with this left user to uniformly random (independently of b) in one shot. Finally, we switch the i -th left user’s public key back to normal with an LWE challenge. Of course, in all of these steps we have to also pay attention that all $\mathcal{O}_{\text{revL}}/\mathcal{O}_{\text{revR}}/\mathcal{O}_{\text{testL}}/\mathcal{O}_{\text{testR}}$ queries are answered with the suitable secret key, which requires carefully switching which user’s secret key is used for these queries, as in the single-challenge case.

Each of these hybrid switches only incurs a constant security loss, thus we get a total security loss of $\mathcal{O}(N_L)$ even with in the multi-challenge setting.

B.5 Left-over hash Lemma.

The last step of our tighter proof requires us to argue that $(\mathbf{A}, \mathbf{A}\mathbf{u})$ for $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$ and $\mathbf{u} \leftarrow \{0, 1\}^m$ is statistically close to a uniformly random value from $\mathcal{R}_q^{n \times m} \times \mathcal{R}_q^n$, even given some linear leakage about $\mathbf{u}$. We present here a general argument for cyclotomic rings. If $\mathcal{R} = \mathbb{Z}$, the simpler variant in Section B.5 is sufficient.

Cyclotomic rings. In cyclotomic rings $\mathcal{R}$ we cannot argue with the traditional left-over hash lemma from above, because $h_{\mathbf{A}} : \mathcal{R}^m \rightarrow \mathcal{R}^n, \mathbf{u} \mapsto \mathbf{A}\mathbf{u}$ is in general not a universal hash function. Instead, we use the algebraic left-over hash lemma [LW20, Theorem 5.5]. We recall the version by [Jin+25].

Lemma B.21 (Algebraic leftover hash Lemma (ALOHL), [Jin+25, Lemma 5.1]). *Let $\mathcal{R}$ be a cyclotomic ring, $q, n, m \in \mathbb{N}_+$, $(\mathcal{U}, \text{aux})$ be correlated distributions where $\mathcal{U}$ is a distribution over $\mathcal{R}^m$. Then for $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$, $\mathbf{u} \leftarrow \mathcal{U}$, $\mathbf{x} \xleftarrow{\$} \mathcal{R}_q^n$ we have*

$$\text{SD}((\mathbf{A}, \mathbf{A}\mathbf{u}, \text{aux}), (\mathbf{A}, \mathbf{x}, \text{aux})) \leq \frac{1}{2} \sqrt{\sum_{\mathbf{q} \mid \langle q \rangle} \mathcal{N}(\mathbf{q})^n \cdot 2^{-H_2(\mathcal{U}_{\mathbf{q}} | \text{aux})} - 1}$$

where $\mathcal{U}_{\mathbf{q}}$ is the distribution that samples $\mathbf{u} \leftarrow \mathcal{U}$ and returns $\mathbf{u} \bmod \mathbf{q}$.

For the case where $\mathcal{R}$ is a cyclotomic ring and q is prime we get the following special case.

Corollary B.22 *Let $\mathcal{R}$ be the ℓ -th cyclotomic ring of degree $d := \phi(\ell)$ with $d \leq \text{poly}(\lambda)$ for a polynomial poly . Moreover, let q be a prime, $\langle q \rangle = \mathfrak{p}_1^h \cdots \mathfrak{p}_g^h$ with $\mathcal{N}(\mathfrak{p}_i) = q^f$ be the prime ideal factorization of $\langle q \rangle$ (see Section A.3), $n, m \in \mathbb{N}_+$, and $(\mathcal{U}, \text{aux})$ be correlated distributions where $\mathcal{U}$ is a distribution over $\mathcal{R}^m$. If for every ideal $\mathbf{q} \mid \langle q \rangle$, $\mathbf{q} \neq \langle 1 \rangle$ we have $H_{\infty}(\mathcal{U}_{\mathbf{q}} | \text{aux}) \geq n \cdot d \cdot \log q + g \log(h + 1) + \lambda$ then*

$$\text{SD}((\mathbf{A}, \mathbf{A}\mathbf{u}, \text{aux}), (\mathbf{A}, \mathbf{x}, \text{aux})) \leq 2^{-\lambda/2-1}$$

The proof of this corollary is inspired by [Jin+25, Theorem 5.3].

Proof. By Lemma A.4 we have $H_2(\mathcal{U}_q \mid \text{aux}) \geq H_\infty(\mathcal{U}_q \mid \text{aux})$. Moreover, $\mathcal{N}(\langle 1 \rangle) = 1$. Thus, we get

$$\begin{aligned}
\sum_{q \mid \langle q \rangle} \mathcal{N}(q)^n \cdot 2^{-H_2(\mathcal{U}_q \mid \text{aux})} - 1 &\leq \sum_{\substack{q \mid \langle q \rangle \\ q \neq \langle 1 \rangle}} \mathcal{N}(q)^n \cdot 2^{-(n \cdot d \cdot \log q + g \log(h+1) + \lambda)} \\
&= \sum_{\substack{0 \leq i_1, \dots, i_g \leq h \\ i_1 + \dots + i_g \geq 1}} \mathcal{N}(p_1^{i_1} \dots p_g^{i_g})^n \cdot (h+1)^{-g} \cdot q^{-n \cdot d} \cdot 2^{-\lambda} \\
&= (h+1)^{-g} \cdot 2^{-\lambda} \sum_{\substack{0 \leq i_1, \dots, i_g \leq h \\ i_1 + \dots + i_g \geq 1}} q^{f(i_1 + \dots + i_g)n - n \cdot d} \\
&\leq (h+1)^{-g} \cdot 2^{-\lambda} \sum_{\substack{0 \leq i_1, \dots, i_g \leq h \\ i_1 + \dots + i_g \geq 1}} 1 \leq 2^{-\lambda}.
\end{aligned}$$

Plugging this into Lemma B.21 shows the corollary. $\square$

Corollary B.23 *For any $q, k, n \in \mathbb{N}_+$ with q prime, any $\mathbf{X} \in \mathcal{R}_q^{k \times m}$ and $m \geq (n \cdot d + k) \cdot \log q + g \log(h+1) + \lambda$ we have for $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$ and $\mathbf{u} \xleftarrow{\$} \{0, 1\}^m$ and $\mathbf{x} \xleftarrow{\$} \mathcal{R}_q^n$*

$$\text{SD}((\mathbf{A}, \mathbf{A}\mathbf{u}, \mathbf{X}\mathbf{u}), (\mathbf{A}, \mathbf{x}, \mathbf{X}\mathbf{u})) \leq 2^{-\lambda/2-1}.$$

Proof. For every ideal $q \neq \langle 1 \rangle$ we have $1 \neq 0 \bmod q$ and thus $H_\infty(\mathcal{U}_q) \geq m$. By Lemma A.3, we get $H_\infty(\mathcal{U}_q \mid \mathbf{X}\mathbf{u}) \geq m - k \log q$. Plugging this into Corollary B.22 yields the statement. $\square$

C The HPS perspective.

NIKE from hash proof systems. Hesse, Hofheinz and Kohl [HHK18] show that we can generically transform a hash proof system to a NIKE with security loss $\mathcal{O}(N)$. A hash proof system, introduced in [CS02], is a proof system for an NP relation $\mathcal{R}$ where proofs for a statement x (certifying $x \in \mathcal{L} := \{x \mid \exists w : (x, w) \in \mathcal{R}\}$) can be generated using a hash public key (sometimes called the projection key) hpk via a procedure $\text{ProjHash}(\text{hpk}, x, w)$ that requires both the statement and the witness. Alternatively, one can generate a proof using the hash secret key hsk via $\text{Hash}(\text{hsk}, x)$ without the witness w . For correctness, we require $\text{ProjHash}(\text{hpk}, x, w) = \text{Hash}(\text{hsk}, x)$ for all $(x, w) \in \mathcal{R}$. The security property we need here is smoothness, which says that for any $x \notin \mathcal{L}$ the proof generated via $\text{Hash}(\text{hsk}, x)$ is uniformly random, even when given hpk . Moreover, we need that $\mathcal{L} \subseteq \mathcal{X}$ such that no PPT distinguisher can distinguish a sample x generated via $(x, w) \xleftarrow{\$} \mathcal{R}$ from a uniform sample $x \xleftarrow{\$} \mathcal{X} \setminus \mathcal{L}$, except with negligible probability (subset membership problem).

This can be turned into an asymmetric NIKE as follows: The left user samples $(x, w) \xleftarrow{\$} \mathcal{R}$ and uses this as its key pair (i.e., its public key is x and its secret key is w). The right user generates a key pair (hpk, hsk) and uses this as its key pair for the NIKE (i.e., hpk is its public key and hsk its secret key). The shared key for two users is simply a proof of the left user's statement x under the right user's hashing keys. The right user, who knows hsk but not w , can compute this proof via Hash and the left user, who knows w and x but not hsk , can compute this via ProjHash .

To prove security, the reduction only has to guess the left challenge user and switch its public key to $x \xleftarrow{\$} \mathcal{X} \setminus \mathcal{L}$. When this guess is wrong, the reduction aborts. Under the hardness of the subset membership problem, the adversary cannot detect this change. With $x \in \mathcal{X} \setminus \mathcal{L}$ the shared key between the left challenge user and the right challenge user is uniformly random from the adversary's point of view. In fact, the shared keys between the left challenge user and *every* right user of the system is uniformly random from the adversary's point of view.

In [HHK18], the authors instantiate this with a group-based hash proof system. The difficulty when instantiating this transformation based on post-quantum assumptions is the lack of suitable hash proof systems from post-quantum assumptions.

The first lattice-based hash proof system was introduced in [KV09]. However, their scheme (and all other lattice-based constructions such as [ZY17; Ben+18]) have the following two limitations:

1. **Approximate correctness:** Proofs are elements in $\mathbb{Z}_q$ and $|\text{ProjHash}(\text{hpk}, x, w) - \text{Hash}(\text{hsk}, x)|$ is small. To get proofs that are identical with high probability, we can round these values, but we cannot get perfect correctness (i.e. guarantee that $\text{ProjHash}(\text{hpk}, x, w) = \text{Hash}(\text{hsk}, x)$ always holds).
2. **Gap smoothness:** The smoothness property only holds for a subset of the statements in $\mathcal{X} \setminus \mathcal{L}$. More concretely, in these hash proof systems the language $\mathcal{L}$ consists of all vectors close (say, within distance B_1) to a lattice Λ . Smoothness, however, holds only for vectors far away (with distance at least $B_2 \gg B_1$) from the lattice Λ .

There are some code-based hash proof systems that get rid of the second limitation, but not of the first limitation [Bet+22; TV23].

In this work, we observe that the second limitation is unproblematic for the transformation to NIKE. Namely, as long as there is an (efficiently samplable) distribution such that (1) smoothness holds for samples of this distribution (with overwhelming probability), and (2) samples from this distribution are computationally indistinguishable from samples of $\mathcal{L}$. The reason is that in the security proof the reduction chooses the statements for which smoothness needs to hold. This is in contrast to other applications of hash proof systems, such as IND-CCA secure public-key encryption, where the adversary can choose the statement [CS02].

Secondly, we can also get away with approximate correctness, as long as the correctness error is negligible. The reason is that then in the security game, with overwhelming probability, in every pair of users both users obtains the same shared key. This allows the reduction to freely choose which user's secret key is used to obtain the shared keys. Of course, the resulting NIKE scheme will then also have a negligible correctness error.

Requiring a negligible correctness error is very undesirable for lattice-based scheme, because it requires a superpolynomial modulus-to-noise ratio, which makes the scheme less efficient. We show that for the lattice-based instantiation we can even tolerate a non-negligible correctness error when the number of users is a priori bounded. This result is based on techniques by Langrehr [Lan23] and leverages the specific structure of lattices (and thus cannot be applied to approximately correct hash proof systems in general).

A lattice-based HPS. With the observations from above, we note that the following basic lattice-based HPS (which is also the starting point for the more advanced lattice-based HPS constructions in [KV09; ZY17; Ben+18]) suffices to build a NIKE with security loss of only $\mathcal{O}(N)$. For simplicity, we describe the scheme here over $\mathbb{Z}_q$, but we show in the technical sections that we can use instead also a cyclotomic ring and thus get constructions based on module and ring LWE.

As global setup we have a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ with $m \approx n \log(q)$ which defines the lattice $\Lambda = \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{s} \in \mathbb{Z}_q^n : \mathbf{s}^\top \mathbf{A} = \mathbf{y}^\top \bmod q\}$. The language $\mathcal{L}$ consists of the vectors in $\mathbb{Z}_q^m$ that are close to the lattice Λ , which we sample as $\mathbf{y}^\top := \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{\Sigma}}$ (where $\mathcal{D}_{\mathbb{Z}^m, \sqrt{\Sigma}}$ is the discrete Gaussian distribution with covariance Σ). The vector $\mathbf{s}$ serves here as a witness for $\mathbf{y}$. Under the LWE assumption, $\mathbf{y}$ is computationally indistinguishable from a uniformly random vector.

A hash secret key hsk is simply a uniformly random binary¹⁶ vector $\text{hsk} \xleftarrow{\$} \{0, 1\}^m$ and the corresponding hash public key is $\text{hpk} := \mathbf{A} \cdot \text{hsk}$. Approximate correctness now follows from

$$(\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top) \cdot \text{hsk} = \mathbf{s}^\top \mathbf{A} \cdot \text{hsk} + \mathbf{e}^\top \cdot \text{hsk} \approx \mathbf{s}^\top \mathbf{A} \cdot \text{hsk} = \mathbf{s}^\top \cdot \text{hpk}.$$

The weak form of smoothness that we need here is $(\mathbf{y}, \mathbf{A}, \mathbf{A} \cdot \text{hsk}, \mathbf{y}^\top \cdot \text{hsk}) \approx_s (\mathbf{y}, \mathbf{A}, \mathbf{A} \cdot \text{hsk}, \pi)$ for uniformly random $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^m$ (and $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\text{hsk} \xleftarrow{\$} \{0, 1\}^m$, $\pi \xleftarrow{\$} \mathbb{Z}_q$). This follows directly from the left-over hash Lemma. In cyclotomic rings, we need here the algebraic left-over hash lemma [LW20, Theorem 5.5].

Plugging this hash proof system into the transformation of [HHK18] yields the core of the lattice-based NIKE. We then have to add the rounding function to deal with approximate correctness. Furthermore, for polynomial modulus-to-noise ratio, we have to make some changes to the security proof to incorporate the technique by Langrehr [Lan23].

¹⁶Any other short distribution would work here, too.