This is the full version [GdKQ+23] of a USENIX 2024 paper with the same title [GdKQ+24].

SWOOSH: Efficient Lattice-Based Non-Interactive Key Exchange

Phillip Gajland[1,2], Bor de Kock[3], Miguel Quaresma[1], Giulio Malavolta[4,1], and Peter Schwabe[1,5]

[1]Max Planck Institute for Security and Privacy, Bochum, Germany
[2]Ruhr University Bochum, Bochum, Germany
[3]NTNU – Norwegian University of Science and Technology, Trondheim, Norway
[4]Bocconi University, Milan, Italy
[5]Radboud University, Nijmegen, The Netherlands
phillip.gajland@{mpi-sp.org,rub.de}, bor.dekock@ntnu.no,
{miguel.quaresma,giulio.malavolta}@mpi-sp.org, peter@cryptojedi.org

Abstract

The advent of quantum computers has sparked significant interest in post-quantum cryptographic schemes, as a replacement for currently used cryptographic primitives. In this context, lattice-based cryptography has emerged as the leading paradigm to build post-quantum cryptography. However, all existing viable replacements of the classical Diffie-Hellman key exchange require additional rounds of interactions, thus failing to achieve all the benefits of this protocol. Although earlier work has shown that lattice-based Non-Interactive Key Exchange (NIKE) is theoretically possible, it has been considered too inefficient for real-life applications.

In this work, we challenge this folklore belief and provide the first evidence against it. We construct an efficient lattice-based NIKE whose security is based on the standard module learning with errors (M-LWE) problem in the quantum random oracle model. Our scheme is obtained in two steps: (i) A passively-secure construction that achieves a strong notion of correctness, coupled with (ii) a generic compiler that turns any such scheme into an actively-secure one. To substantiate our efficiency claim, we provide an optimised implementation of our passively-secure construction in Rust and Jasmin. Our implementation demonstrates the scheme's applicability to real-world scenarios, yielding public keys of approximately 220 KBs. Moreover, the computation of shared keys takes fewer than 12 million cycles on an Intel Skylake CPU, offering a post-quantum security level exceeding 120 bits.

1 Introduction

A key exchange is a fundamental cryptographic primitive that allows two users to agree on a common secret key over an insecure channel, such as the Internet. When the protocol involves a single, asynchronous message from each party, it is known as a *Non-Interactive Key Exchange* (NIKE). The seminal work of Diffie and Hellman [DH76] introduced the well-known NIKE scheme that marked the birth of public-key cryptography; each party sends a single group element g^x (or g^y, respectively) and the shared key can be derived by computing $(g^y)^x = (g^x)^y$. From a theoretical stand-point NIKE implies the existence of public key encryption (PKE), key encapsulation mechanism (KEM), and even authenticated key-exchange (AKE) when combining the results of [BCGP08] with [FHKP13]. Moreover, in practice, the Diffie-Hellman key exchange lies at the heart of widely-used protocols such as Transport Layer Security (TLS) [Res18], the Signal protocol [MP16a, MP16b], or the Noise protocol framework [Per18].

The looming threat of quantum computers, combined with the discovery of efficient quantum algorithms for factoring integers and computing discrete logarithms [Sho94], has necessitated the exploration of alternative solutions based on new mathematical structures, departing from protocols based on the Diffie-Hellman key exchange. In particular, lattice-based cryptography [Reg05] has emerged as the leading paradigm for constructing *post-quantum* cryptographic schemes. Notably, three out of four algorithms selected for standardisation by NIST are lattice-based [SAB+22, LDK+22, PFH+22].

While efficient lattice-based key exchange protocols exist [BCD+16, ADPS16, SAB+22], they are all qualitatively different from the standard Diffie-Hellman-style key exchange, in the sense that they require *additional rounds of interaction*. For many applications, where interaction is already built-in, these protocols are perfectly fine substitutes for Diffie-Hellman (which is not post-quantum secure). However, in many scenarios of interest, the non-interactive nature of NIKE protocols is crucial (we discuss concrete examples in further detail in Section 1.1). Unfortunately, despite almost two decades of research on the subject, an efficient lattice-based NIKE remains elusive. Perhaps more worryingly, a recent work [GKRS20] has shown theoretical barriers on the efficiency of lattice-based NIKE, calling into question whether it is even possible to build an efficient scheme at all. Thus, the current state of affairs, leaves open the following question:

Is lattice-based *non-interactive* key exchange feasible in practice?

In our work we seek to answer this question in the affirmative, and show that lattice-based NIKE can be made efficient enough to be used in practice, whilst maintaining post-quantum security.

1.1 NIKE vs. KEMs and Applications

While the Diffie-Hellman (DH) key exchange happens to be non-interactive, most post-quantum approaches to key exchange are interactive key-encapsulation mechanisms (KEMs). In a NIKE, any user A can derive a shared key k_{AB} using their secret key sk_A and the public key pk_B of user B. At the same time, and without interaction with A, user B can compute the same shared key k_{AB} by combining their secret key sk_B with the public key pk_A of user A. On the other hand, in a KEM, this key-derivation becomes a two-stage, inherently asymmetric and interactive process. First, A invokes an encapsulation routine that accepts pk_B as input and produces as output the shared key k_{AB} and a ciphertext ct, which they send to B. User B then invokes the decapsulation routine that takes as input ct and secret key sk_B to produce the same shared secret k_{AB}. At this point, it is worth noting that KEMs should satisfy the standard security notions of IND-CCA security. The security models considered for NIKE are discussed in Section 4. Notably, we examine both a passive security notion and an active variant. In some protocols that employ DH, the non-interactive nature is not actually utilised, making the migration to post-quantum KEMs straightforward. Probably the most prominent example is TLS, which uses the DH key exchange with ephemeral keys on both sides for forward secrecy and has been updated to offer post-quantum security by using KEMs in multiple papers [BCNS15, PST20, BBCT22] and real-world deployments [Lan16, Lan18, KV19, WR19].

However, other protocols *do* make use of the non-interactive nature of DH and their migration to post-quantum primitives is thus much more involved. A common pattern in these protocols is the use of static DH keys for authentication. One example is OPTLS by Krawczyk and Wee [KW16], a proposal that eliminates the need for handshake signatures in TLS. The idea was picked up in the post-quantum setting in the KEMTLS proposal by Schwabe, Stebila, and Wiggers [SSW20]. Like OPTLS, also KEMTLS eliminates the need for handshake signatures, but unlike OPTLS uses static KEM keys for authentication. This adaptation comes at the cost of additional communication round-trips until full server authentication is achieved, which can be problematic in scenarios where protocols like HTTPS allow the server to transmit early payload data before completing the handshake. Similar issues with delayed authentication when moving from DH to KEMs were identified in the migration of WireGuard to the post-quantum setting in [HNS+21] and in the the recently proposed post-quantum version of the Noise protocol framework [ADH+22].

While these examples manage to migrate from DH to KEMs, they come at the additional cost of further communication round trips without requiring signature-based authentication or more complex cryptographic primitives. However, if communicating parties cannot be assumed to be online at the same time, this approach is doomed to fail. A prominent example of precisely this asynchronous communication setting is the Signal secure-messaging protocol and specifically the X3DH protocol [MP16b] that is invoked when a user A starts their communication with a (possibly offline) user B. The X3DH protocol uses a combination of ephemeral, static, and semi-static DH keys to achieve forward secrecy, mutual authentication, and offline deniability without the need for direct interaction between A and B. Several attempts have been made to migrate X3DH to the post-quantum setting [BFG+20, HKKP21, SSKF21, DG22, BFG+22], but they all either assume the existence of a reasonably efficient post-quantum NIKE, or fail to achieve the same security and privacy as the pre-quantum version from a single simple asymmetric primitive.

1.2 Our Contributions

In this work, we demonstrate the real-world feasibility of lattice-based *non-interactive key exchange*. We propose a new scheme, that we call "SWOOSH", based on the hardness of the M-LWE problem. We show a proof of its security, both in the passive and active setting, and provide parameter sets for the former with over 120-bits of security against quantum adversaries (using the best known attacks that incorporate recent advances in lattice cryptanalysis). Our contributions can be summarised as follows.

1. We propose a new construction of NIKE based on the hardness of the M-LWE problem. Our construction is based on the standard template [DXL12, Lyu17], but with a new tweak that allows us to prove a strong notion of correctness (which, in turn, is necessary to achieve active security) in the quantum random oracle model (QROM). Somewhat interestingly, our use of the random oracle appears to be different from the Fiat-Shamir [FS87] and the Fujisaki-Okamoto [FO99, FO13] transformations, and may thus be of independent interest.

2. We propose a compiler that allows for the generic transformation of a passively secure NIKE into an

actively secure scheme using non-interactive zero-knowledge (NIZK) proofs, that satisfy a strong soundness property (simulation-sound online-extractability). While this approach is folklore, we provide, to the best of our knowledge, the first explicit treatment of this technique in the literature. Furthermore, the exact notion of passive security needed for the proof to go through, turns out to be surprisingly subtle to identify.

3. We provide a highly optimised implementation[1] of Passive-SWOOSH, written in Rust and Jasmin [ABB+17, ABB+20]. With carefully selected parameters, our scheme achieves more that 120 bits of security against *quantum* adversaries. Notably, our benchmarks reveal smaller public keys compared to the smallest parameter set of Classic McEliece [ABC+22], an interactive KEM selected for round 4 of the NIST-PQC competition. Furthermore, we demonstrate that Passive-SWOOSH outperforms CSIDH [CLM+18], the only currently known (and realistic) post-quantum NIKE by orders of magnitude in terms of speed. Together with existing NIZK proof libraries [LNP22, LNS20], our work establishes the first competitive construction of a lattice-based NIKE for practical use.

1.3 Related work

Post-quantum NIKE. While interactive KEMs appear to be much more efficient in a post-quantum world than NIKEs, there have been notable efforts towards constructing post-quantum NIKE schemes as well. Boneh and Zhandry [BZ14] showed a construction using indistinguishability obfuscation (iO) to construct a multiparty NIKE from pseudorandom generators. However, the practicality of this approach is hindered by the performance limitations of iO, rendering it mainly of theoretical interest. Much more practical was supersingular-isogeny Diffie-Hellman (SIDH) [JD11, CLN16]. However, in 2016, this construction was shown to be susceptible to active attacks [GPST16]. This could be solved by employing the Fujisaki-Okamoto transform [FO99] in the NIST PQC candidate SIKE [JAC+17], but this came at the expense of turning the NIKE into an interactive KEM. Another approach to restoring the active security of SIKE was presented in [AJL17]. This approach preserved the non-interactive nature of SIDH, but required many parallel protocol executions and thus massively increased computation time and message sizes. In 2022, all of these approaches based on SIDH were made obsolete by the numerous attacks against SIDH [CD23, MMP+23, Rob23].

[1] See https://github.com/MQuaresma/pswoosh.

In 2018, Castryck, Lange, Martindale, Panny, and Renes proposed CSIDH, a different approach for constructing an isogeny-based NIKE [CLM+18]. CSIDH is not affected by the attacks on SIDH, and is arguably the most plausible candidate for practical post-quantum NIKE thus far, although the post-quantum security of concrete parameters is subject of debate [BS20, BLMP19, Pei20]. Multiple works have considered the efficient and secure implementation of CSIDH, currently the fastest approach is a variant called CTIDH [BBC+21]. We provide a performance comparison of our proposal to CTIDH in Section 6.2. Notably, the work of [BC18] introduced a compiler for achieving passive to active security. However, the compiler assumes a base scheme with perfect correctness and thus it does not apply to lattice-based NIKE.

Lattice-based NIKE. The idea of lattice-based NIKE using the approach employed in Passive-SWOOSH is not new; in [Lyu17] Lyubashevsky calls it *"folklore (since at least 2010)"*. An attempt at selecting parameters was made in [dK18]. However, the proposed scheme did not formally consider passive security, nor active security. Moreover, the selected parameters resulted in a correctness error that would not even allow the transformation into an actively secure scheme through the use of NIZK proofs, a crucial aspect we achieve in SWOOSH.

In fact, prior to our work, lattice-based NIKE was widely considered impractical and this was even substantiated by theoretical evidence. The work of [GKRS20] discovered information-theoretic barriers in constructing lattice-based NIKE with non-interactive reconciliations. In particular, they showed that any natural candidate of lattice-based NIKE with polynomial modulus-to-noise ratio would necessarily incur an inverse-polynomial correctness error. However, we stress that our work does not contradict the theorem of [GKRS20]. As the authors of [GKRS20] observe, non-interactive reconciliation is possible, if we consider (M-)LWE instances with *super-polynomial* modulus-to-noise ratio. This is indeed the regime of parameters that we adopt in our work.

2 Technical Outline

We give a self-contained overview of our approach for constructing a fast lattice-based NIKE. The following is somewhat informal and glosses over many important details, as it is only intended for an intuitive understanding of our approach. The reader is referred to the respective technical sections for precise statements.

The Basic Blueprint. Before delving into the specifics of our approach, it is useful to recall the folklore construction of lattice-based key exchange between Alice and Bob. Let $\boldsymbol{A}$

be a random public $N \times N$ matrix over some ring $\mathcal{R}_q$ and χ a noise distribution. The protocol proceeds as follows; Alice samples $\vec{s}_1$ and $\vec{e}_1$ from χ^N, and computes her public key as $\vec{s}_1^\top \boldsymbol{A} + \vec{e}_1^\top$. Bob samples an independent $\vec{s}_2$ and $\vec{e}_2$ from χ^N, and computes his public key as $\boldsymbol{A}\vec{s}_2 + \vec{e}_2$. After asynchronously obtaining each other's public keys, Alice and Bob can compute an approximate shared key as

$$\vec{s}_1^\top (\boldsymbol{A}\vec{s}_2 + \vec{e}_2) \approx \left(\vec{s}_1^\top \boldsymbol{A} + \vec{e}_1^\top\right) \vec{s}_2.$$

A simple calculation shows that the shared keys computed by both parties are identical with the exception of the error terms $\vec{s}_1^\top \vec{e}_2$ and $\vec{e}_1^\top \vec{s}_2$ for Alice and Bob, respectively. To correct these errors, known schemes in the literature run interactive *reconciliation* protocols, which can be realised quite efficiently. However, if we insist on a NIKE protocol, no further interaction is allowed, and Alice and Bob must correct the errors locally. That is, we need to devise a *non-interactive* reconciliation function `Rec` such that

$$\texttt{Rec}\left(\vec{s}_1^\top (\boldsymbol{A}\vec{s}_2 + \vec{e}_2)\right) = \texttt{Rec}\left(\left(\vec{s}_1^\top \boldsymbol{A} + \vec{e}_1^\top\right) \vec{s}_2\right).$$

Note that, thus far, we have assumed that both Alice and Bob compute their keys according to the specification of the protocol, i.e., we implicitly only considered passive attacks. However, for the security of the final scheme, it will be necessary to handle parties that may behave arbitrarily. In what follows, we show how we tackle these two challenges separately, in a way that preserves the efficiency and security of the scheme.

Challenge I: Non-Interactive Reconciliation. A natural approach for correcting the errors introduced by the noise terms, is to derive the key by *rounding* the coefficients of the resulting ring element. In fact this is the approach that we adopt in this work, however there are still new ideas required to simultaneously achieve all of the following objectives: (i) security from the hardness of the standard module learning with errors (M-LWE) problem, (ii) reducing the correctness error to negligible, and (iii) maintaining the concrete efficiency of the construction. Here, we stress that a negligible correctness error is not just a matter of convenience, but that a non-negligible correctness error translates to an attack against the scheme: Loosely speaking, this is because the attacker can observe whenever the key agreement fails, therefore learning some information about the secret key of the honest party. Let us now focus on making the rounding approach work for non-interactive reconciliation. A simple calculation shows that the error terms cause a correctness error, only when the term $\vec{s}_1^\top \boldsymbol{A}\vec{s}_2$ falls into a *danger interval*

$$S^* = \left[\frac{q}{4} \pm \beta^2 dN\right] \cup \left[\frac{3q}{4} \pm \beta^2 dN\right],$$

where β is a bound on the norm of the noise distribution and d is the degree of $\mathcal{R}_q$. It is tempting to conclude that, if q is sufficiently large, then this event only happens with negligible probability. However, this analysis is imprecise as it does not take into account *adaptive* attacks, where the adversary chooses their secret key intentionally to make this event more likely. To prevent this, and obtain a provably secure scheme, we add a *random shift* $\boldsymbol{r}$ to the term $\vec{s}_1^\top \boldsymbol{A}\vec{s}_2$ to ensure that their sum $\vec{s}_1^\top \boldsymbol{A}\vec{s}_2 + \boldsymbol{r}$ is indeed uniformly distributed in $\mathcal{R}_q$. Note that such $\boldsymbol{r}$ does not need to be kept private, although it is important that it is sampled independently of the keys. Our idea is to sample $\boldsymbol{r}$ as the output of a hash function (modelled as a random oracle) on input the two public keys. This allows us to achieve two goals simultaneously:

- Both parties can recompute the shift $\boldsymbol{r}$ without the need of further interaction.
- We can show that $\vec{s}_1^\top \boldsymbol{A}\vec{s}_2 + \boldsymbol{r}$ is indeed uniformly sampled, even if the adversary has quantum access to the random oracle.

In summary, we are able to build a non-interactive reconciliation mechanism so that the scheme is provably secure (in the passive settings) against the standard M-LWE assumption, in the QROM. In fact, we are also able to show a strong notion of correctness, namely that the adversary cannot cause a reconciliation error, *even if it is allowed to choose both secret keys*. This strong notion of correctness will be useful when lifting the scheme to the active setting.

Challenge II: From Passive to Active Security. The above discussion concerns keys that are guaranteed to be well-formed (passive security). However, in real-world scenarios we have to deal with attackers that can behave arbitrarily. In the stronger notion of *active security* [CKS08, FHKP13] the adversary is given access to various oracles that allow them to register honest keys, register corrupt keys (ones to which they do not know the corresponding secret key), or reveal the shared key between an honest key and a corrupted one. Ultimately the adversary wins if he can distinguish between a random key and a shared key, that was derived from two honestly generated key pairs.

In order to prove the active security of our scheme we present a *compiler* that generically lifts our scheme to the active setting using non-interactive zero-knowledge (NIZK) proofs. Here it is crucial that our scheme satisfies the aforementioned strong notion of correctness, since the only thing that the NIZK guarantees is that the keys are in the support of the honest distributions, but otherwise they may be chosen arbitrarily. For technical reasons, we require a NIZK that satisfies the strong property of simulation-sound online-extractability. We refer the reader to Section 5 for more details.

Putting Everything Together. Overall, we obtain a passively secure construction in the QROM assuming the hardness of the Module-LWE (M-LWE) problem (for the active settings, we additionally require a NIZK proof). Compared to Ring-LWE (R-LWE), M-LWE gives us greater flexibility over the choice of parameters, when implementing our scheme. However, this introduces an additional complication: Unlike the case for R-LWE, where single polynomials are considered and their multiplication is commutative, in the case of M-LWE we work with matrices where the matrix multiplication is not generally commutative. For the general case of two parties without predefined roles in a protocol, there is no way to know ahead of time whether to left multiply or right multiply. This means that each public key is effectively duplicated by adding a left multiplied key and right multiplied key. However, we argue that in many cases, when parties have predefined roles in a protocol, such as a server or client, this issue can be resolved (the server could "go right" and the client "left" or vice versa). We defer a more detailed discussion of this to Section 5.3.

Our parameters are selected as to provide more than 120 bits of post-quantum security, taking into account recent advances in lattice cryptanalysis. We work over the ring $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d+1)$ with $d = 256$. Along with our public matrix $\boldsymbol{A} \in \mathcal{R}_q^{N\times N}$, where $N = 32$, this gives us a lattice dimension of 8192. In order to reduce the correctness error to reasonable levels, q had to be sufficiently large. We choose $q = 2^{214} - 255$, a prime that is simultanously NTT-friendly and close to a power-of-two making for more efficient field arithmetic. Furthermore, we use ternary noise sampled from a centred binomial distribution, for the sake of efficiency.

Finally, we provide an open-source implementation of Passive-SWOOSH in Rust and Jasmin, which employs numerous optimisations rendering competitive benchmarks. Due to the modular fashion of our implementation we note that it can easily be tailored to use different parameters or be incorporated with suitable NIZKs. We defer a more detailed discussion to Section 6.

3 Preliminaries

In this section we introduce our notation and review some quantum preliminaries along with the relevant lattice-based hardness assumptions.

3.1 Notation

We define some standard notation used throughout the paper.

Sets, Vectors, Polynomials and Norms. For integers a, b, where $a < b$, $[a,b]$ denotes the set $\{a, a+1, \ldots, b\}$. For any positive $\beta \in \mathbb{Z}$, we define the set $[\beta] := \{-\beta, \ldots, -1, 0, 1 \ldots, \beta\}$, and let $x \xleftarrow{\$} \mathcal{S}$ denote the uniform sampling of x from the set $\mathcal{S}$. Let $\mathbb{Z}_q$ denote the ring of integers modulo a prime q. We define $\mathcal{R} := \mathbb{Z}[X]/(X^d+1)$ to be the ring of integer polynomials modulo X^d+1, for d a power of 2, and $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d+1)$ the ring of integer polynomials modulo X^d+1 where each coefficient is reduced modulo q. Bold upper case letters $\boldsymbol{A}$ and bold lower case letters with arrows $\vec{\boldsymbol{a}}$ denote matrices and column vectors over $\mathcal{R}_q$, respectively; for row vectors we use the transpose $\vec{\boldsymbol{b}}^\top$. For a polynomial $\boldsymbol{f} \in \mathcal{R}_q$, let $\vec{f} \in \mathbb{Z}_q^d$ denote the coefficient vector of $\boldsymbol{f}$, and $f_i \in \mathbb{Z}_q$ the i^{th} coefficient. However, we denote the constant coefficient by $\tilde{f} := f_0 \in \mathbb{Z}_q$. For an element $f_i \in \mathbb{Z}_q$, we write $|f_i|$ to mean $|f_i \bmod q|$. Let the ℓ_∞ norm for $\boldsymbol{f} = \sum_{i=0}^{d-1} f_i X^i \in \mathcal{R}_q$ and $\vec{\boldsymbol{f}} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_k) \in \mathcal{R}_q^k$ be defined as

$$\|\boldsymbol{f}\|_\infty := \max_{0\le i\le d-1} |f_i| \quad \text{and} \quad \left\|\vec{\boldsymbol{f}}\right\|_\infty := \max_{1\le i\le k} \|\boldsymbol{f}_i\|_\infty,$$

respectively.

Probabilities, Algorithms and Games. The support of a discrete random variable X is defined as $\sup(X) := \{x \in \mathbb{R} : \Pr[X = x] > 0\}$. Algorithms are denoted by upper case letters in sans-serif font, such as A and B. Unless otherwise stated all algorithms are probabilistic and $(x_1, \ldots) \xleftarrow{\$} \mathsf{A}(y_1, \ldots)$ is used to denote that A returns $(x_1, \ldots)$ when run on input $(y_1, \ldots)$. When A has oracle access to B during its execution, this is denoted by A^{B}. For a probabilistic algorithm A, the notation $x \in \mathsf{A}(y)$ denotes that x is a possible output of A on input y. We use code-based security games [BR06], where $\Pr[\mathsf{G} \Rightarrow 1]$ denotes the probability that the final output of game G is 1. The notation $[\![B]\!]$, where B is a Boolean statement, refers to a bit that is 1 if the statement is true and 0 otherwise. The following lemma demonstrates the high probability of a randomly selected matrix being invertible. Due to space constraints, the proof is deferred to Appendix A.

Lemma 1 (Invertibility of Random Matrices)**.** For $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$ with $d = 256$ and $q = 2^{214} - 255$, if $\boldsymbol{A}$ is a random matrix sampled from $\mathcal{R}_q^{N\times N}$, then the probability of $\boldsymbol{A}$ being invertible, denoted by $\Pr[\boldsymbol{A} \in \mathsf{GL}_N(\mathcal{R}_q)]$, satisfies

$$\Pr[\boldsymbol{A} \in \mathsf{GL}_N(\mathcal{R}_q) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{N\times N}] \ge \left(1 - \frac{128}{q^2}\right)^N.$$

3.2 Quantum Preliminaries

We review some quantum preliminaries as stated in [DHK+22]. Additional preliminaries are deferred to Appendix A.

Quantum Random Oracle Model. In the random oracle model [BR93], all parties have access to a uniformly sampled random function H. Since quantum adversaries can evaluate hash functions in superposition, we model quantum adversaries to have quantum access to random oracles [BDF+11]. Specifically, we assume that all algorithms have access to the unitary implementing the mapping: $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus \mathsf{H}(x)\rangle$ where H is a uniformly sampled random function.

Query Depth and Query Parallelism. As in the work of [AHU19] we consider the query depth D of an adversary making a total of Q_{H} random oracle queries. This is important in practice because for highly parallel adversaries we have $D \ll Q_{\mathsf{H}}$. By setting $D := Q_{\mathsf{H}}$ we obtain the bounds for sequential adversaries. We will use the following technical lemma from [AHU19].

Lemma 2 (Search in unstructured functions [AHU19, Lem. 2])**.** *Let* H *be a random function drawn from a distribution such that* $\Pr[\mathsf{H}(x) = 1] \leq \lambda$ *for all x. Let* A *be an adversary with query depth D, making at most* Q_{H} *many queries to* H*. Then*

$$\Pr\left[\mathsf{H}(x) = 1 : b \xleftarrow{\$} \mathsf{A}^{\mathsf{H}}\right] \leq 4 \cdot (D+2) \cdot (Q_{\mathsf{H}}+1) \cdot \lambda.$$

3.3 Hardness Assumption

The security of our scheme relies on Module-Learning With Errors (M-LWE), a well-known computational lattice problem [Reg05, LS15].

Definition 1 ($\mathbf{M\text{-}LWE}_{q,n,m,\chi}$)**.** The decisional *Module-Learning With Errors* problem (in its Hermite normal form) with parameters $n, m > 0$ and an error distribution χ over $\mathcal{R}_q$ is defined via the game $\mathbf{M\text{-}LWE}^b_{q,n,m,\chi}$ depicted in Figure 1. Here, $\mathbf{M\text{-}LWE}^b_{q,n,m,\chi}$ is parameterised by a bit b. We define A's advantage in $\mathbf{M\text{-}LWE}^b_{q,n,m,\chi}$ as

$$\mathsf{Adv}^{\mathbf{M\text{-}LWE}}_{q,n,m,\chi}(\mathsf{A}) := \left| \begin{array}{l} \Pr[\mathbf{M\text{-}LWE}^{0,\mathsf{A}}_{q,n,m,\chi} \Rightarrow 1] \\ -\Pr[\mathbf{M\text{-}LWE}^{1,\mathsf{A}}_{q,n,m,\chi} \Rightarrow 1] \end{array} \right|,$$

and say that $\mathbf{M\text{-}LWE}_{q,n,m,\chi}$ is ε-hard for all adversaries A satisfying $\mathsf{Adv}^{\mathbf{M\text{-}LWE}}_{q,n,m,\chi}(\mathsf{A}) \leq \varepsilon$.

Theoretic treatments of LWE-based schemes typically consider the modulus to be polynomial in n and χ to be the discrete Gaussian on $D_{\mathbb{Z},\alpha\cdot q}$ over $\mathbb{Z}$ with mean 0 and standard deviation $\sigma = \alpha \cdot q/\sqrt{2\pi}$ for some $\alpha < 1$. For these choices the work of [Reg05, BLP+13] showed that if $\alpha q > 2\sqrt{n}$ then worst-case GapSVP-$\tilde{O}(n/\alpha)$ reduces to average-case LWE. As such, many early implementations sampled from a discrete Gaussian distribution, which turns out to be either fairly inefficient [BCNS15] or vulnerable to

Game $\mathbf{M\text{-}LWE}^b_{q,n,m,\chi}$

$b' \xleftarrow{\$} \mathsf{A}^{\mathrm{RoR}(b)}$

return $[\![b = b']\!]$

Oracle $\mathrm{RoR}(b)$ **/ Once**

$\boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{n\times m}$

if $b = 0$:

  $\vec{\boldsymbol{s}} \xleftarrow{\$} \chi^m$

  $\vec{\boldsymbol{e}} \xleftarrow{\$} \chi^n$

  return $(\boldsymbol{A}, \boldsymbol{A}\vec{\boldsymbol{s}} + \vec{\boldsymbol{e}})$

elseif $b = 1$:

  $\vec{\boldsymbol{u}} \xleftarrow{\$} \mathcal{R}_q^n$

  return $(\boldsymbol{A}, \vec{\boldsymbol{u}})$

Figure 1: Game defining $\mathbf{M\text{-}LWE}^b_{q,n,m,\chi}$ with adversary A.

timing attacks [BHLY16, PBY17, EFGT17]. Furthermore, the performance of the best known attacks against LWE-based encryption schemes does not depend on the exact distribution of noise, but rather on the standard deviation (and potentially the entropy). This motivates the use of noise distributions that we can easily, efficiently, and securely sample from. One example is the centred binomial distribution used by CRYSTALS-Kyber [SAB+22] and in [ADPS16].

4 Definitions

In this section we present a formal definition of a non-interactive key exchange along with its security notions. A precise definition of non-interactive zero-knowledge proofs can be found in Appendix B.1.

4.1 Non-Interactive Key Exchange

Following the work of [CKS08, FHKP13], we formally define a non-interactive key exchange (NIKE). Through the use of IDs, the security model proposed in [CKS08] abstracts away all considerations concerning certification and public key infrastructure.

Definition 2 (Non-Interactive Key Exchange)**.** A non-interactive key exchange NIKE is defined as a tuple $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ of the following PPT algorithms. Furthermore, we define an identity space $\mathcal{IDS}$ and a shared key space $\mathcal{SKS}$.

$par \xleftarrow{\$} \mathsf{Stp}(1^\lambda)$**:** Given the security parameter 1^λ (encoded in unary), the probabilistic setup algorithm returns a set of system parameters par.

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}(\mathtt{ID})$**:** Given an identity $\mathtt{ID} \in \mathcal{IDS}$, the probabilistic key generation algorithm Gen returns a secret/public key pair (sk, pk).

$k \leftarrow \mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2)$**:** Given an identity $\mathtt{ID}_1 \in \mathcal{IDS}$ and its corresponding public key pk_1 along with another identity $\mathtt{ID}_2 \in \mathcal{IDS}$ and its corresponding secret key sk_2, the deterministic shared key establishment algorithm SdK returns a shared key $k \in \mathcal{SKS}$, or a failure symbol $\perp$. We assume that SdK always returns $\perp$ if $\mathtt{ID}_1 = \mathtt{ID}_2$.

Correctness. Informally, *honest correctness* states that shared keys derived by two honest parties should be the same with overwhelming probability. Although our subsequent definition of correctness implies honest correctness, we state both definitions here for completeness.

Definition 3 (Honest Correctness)**.** A non-interactive key exchange $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ has *honest correctness error* δ (or is said to be δ*-correct*), if for all $par \in \mathsf{Stp}(1^\lambda)$ and $\mathtt{ID}_1, \mathtt{ID}_2 \in \mathcal{IDS}$ it holds that,

$$\Pr\left[\mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2) \neq \mathsf{SdK}(\mathtt{ID}_2, pk_2, \mathtt{ID}_1, sk_1) \;\middle|\; \begin{matrix}(sk_1, pk_1) \xleftarrow{\$} \mathsf{Gen}(\mathtt{ID}_1)\\ (sk_2, pk_2) \xleftarrow{\$} \mathsf{Gen}(\mathtt{ID}_2)\end{matrix}\right] \leq \delta(\lambda),$$

where the probability is taken over the random choices of Stp and Gen.

In this work we define a stronger notion, *semi-malicious correctness* that captures the property that two maliciously chosen key pairs (that are in the support of the key generation algorithm) will not cause the key exchange to fail. Since this property clearly implies honest correctness, throughout the rest of this work we only focus on semi-malicious correctness. We formalise *semi-malicious correctness* for NIKE relative to a random oracle H via the game $\textbf{SM-COR}_{\mathsf{NIKE}}$ depicted in Figure 2 and define the advantage of an adversary A in $\textbf{SM-COR}_{\mathsf{NIKE}}$ as

$$\mathsf{Adv}^{\textbf{SM-COR}}_{\mathsf{NIKE},par}(\mathsf{A}) := \Pr[\textbf{SM-COR}^{\mathsf{A}}_{\mathsf{NIKE}} \Rightarrow 1].$$

Definition 4 (Semi-malicious Correctness)**.** Let $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ be a non-interactive key exchange. In the quantum random oracle model, we say that NIKE is $\delta(Q_{\mathsf{H}})$-**SM-COR** if for all $\mathtt{ID}_1, \mathtt{ID}_2 \in \mathcal{IDS}$ and for all (possibly unbounded) adversaries A of depth at most D, making at most Q_{H} queries (possibly in superposition) to the random oracle H, we have $\mathsf{Adv}^{\textbf{SM-COR}}_{\mathsf{NIKE},par}(\mathsf{A}) \leq \delta(Q_{\mathsf{H}}, D)$. [2]

Passive Security. Following the conventions of [FHKP13], we formalise the notion of key indistinguishability with *passive security*, or honest key registration (HKR), for a non-interactive key exchange NIKE, with respect to system parameters $par \in \mathsf{Stp}(1^\lambda)$ via the game $\textbf{HKR-CKS-l}_{\mathsf{NIKE},par}$ depicted in Figure 6. In $\textbf{HKR-CKS-l}_{\mathsf{NIKE},par}$, the adversary A may make two queries to the $\mathtt{RegHonUsr}$ oracle, where A provides and identity and the public and secret keys are derived honestly. A may then make one query to the $\mathtt{TestQue}$ oracle, where A has to distinguish the shared key from a random key. We define the advantage of adversary A in $\textbf{HKR-CKS-l}_{\mathsf{NIKE},par}$ as
$\mathsf{Adv}^{\textbf{HKR-CKS-l}}_{\mathsf{NIKE},par}(\mathsf{A}) := \left|\Pr\left[\textbf{HKR-CKS-l}^{\mathsf{A}}_{\mathsf{NIKE},par} \Rightarrow 1\right] - \frac{1}{2}\right|$.

[2] Note that in the standard model our correctness definition can be considered a special case where the number of random oracle queries is zero and hence $\delta(Q_{\mathsf{H}}, D)$ is a constant.

Game $\textbf{SM-COR}_{\mathsf{NIKE}}$

$par \leftarrow \mathsf{Stp}(1^\lambda)$
$\left.\begin{matrix}\mathsf{supp}(\mathsf{Gen}(\mathtt{ID}_1)) \ni (sk_1, pk_1)\\ \mathsf{supp}(\mathsf{Gen}(\mathtt{ID}_2)) \ni (sk_2, pk_2)\end{matrix}\right\} \xleftarrow{\$} \mathsf{A}^{|\mathsf{H}\rangle}(par)$
return $[\![\mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2) \neq \mathsf{SdK}(\mathtt{ID}_2, pk_2, \mathtt{ID}_1, sk_1)]\!]$

Figure 2: Correctness game $\textbf{SM-COR}_{\mathsf{NIKE}}$ for a non-interactive key exchange NIKE defined relative to a random oracle H with adversary A.

Definition 5 (Passive Security)**.** Let $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ be a non-interactive key exchange. We say that NIKE is $(\varepsilon, Q_{\mathsf{H}})$-**HKR-CKS-l** relative to $par \in \mathsf{Stp}(1^\lambda)$ if for all PPT adversaries A, making at most Q_{H} queries (possibly in superposition) to the random oracle H, two queries to the $\mathtt{RegHonUsr}$ oracle and one query to the $\mathtt{TestQue}$ oracle, we have $\mathsf{Adv}^{\textbf{HKR-CKS-l}}_{\mathsf{NIKE},par}(\mathsf{A}) \leq \varepsilon(Q_{\mathsf{H}})$.

Active Security. We formalise the notion of key indistinguishability with *active security* for a non-interactive key exchange NIKE, with respect to system parameters $par \in \mathsf{Stp}(1^\lambda)$ via the game $\textbf{CKS}_{\mathsf{NIKE},par}$ depicted in Figure 7. Observe that this **CKS** (sometimes called DKR-CKS, short for "Dishonest Key Registration") notion was first defined in [CKS08] and is polynomially equivalent to **CKS-l** (where the adversary is only allowed to make two $\mathtt{RegHonUsr}$ queries and one $\mathtt{TestQue}$ query) and *m-CKS-heavy* in the work of [FHKP13]. Unsurprisingly our definition of active security implies the former notion of passive security. The game starts by selecting a bit b uniformly at random after which the adversary A is given access to four oracles. A's queries may be made adaptively and are arbitrary in number. The $\mathtt{RegHonUsr}$ and $\mathtt{RegCorUsr}$ oracles let A register honest and corrupted user public keys, respectively. A may make multiple queries to $\mathtt{RegCorUsr}$, in which case only the most recent $(corrupt, \mathtt{ID}, \perp, pk)$ entry is kept. The $\mathtt{RevCorQue}$ oracle provides A with a shared key between a pair of registered identities, subject only to the restriction that at least one of the two identities was registered as honest. Depending on the bit b, the $\mathtt{TestQue}$ oracle returns either a random key or a shared key between two identities registered as honest. Finally, the adversary outputs a guess bit b' and wins the game if and only if $b = b'$. We define the advantage of adversary A in $\mathbf{CKS}_{\mathsf{NIKE},par}$ as $\mathsf{Adv}^{\mathbf{CKS}}_{\mathsf{NIKE},par}(\mathsf{A}) := \left|\Pr\left[\mathbf{CKS}^{\mathsf{A}}_{\mathsf{NIKE},par} \Rightarrow 1\right] - \frac{1}{2}\right|$.

Definition 6 (Active Security [CKS08])**.** Let $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ be a non-interactive key exchange. We say that NIKE is $(\varepsilon, Q_{\mathsf{H}}, Q_{\mathtt{RHU}}, Q_{\mathtt{RCU}}, Q_{\mathtt{RCQ}}, Q_{\mathtt{TQ}})$**-CKS** secure relative to $par \in \mathsf{Stp}(1^\lambda)$ if for all PPT adversaries A making at most; Q_{H} queries (possibly in superposition) to the random oracle H, $Q_{\mathtt{RHU}}$ queries to $\mathtt{RegHonUsr}$, $Q_{\mathtt{RCU}}$ queries to $\mathtt{RegCorUsr}$, $Q_{\mathtt{RCQ}}$ queries to $\mathtt{RevCorQue}$, and $Q_{\mathtt{TQ}}$ queries to $\mathtt{TestQue}$, we have $\mathsf{Adv}^{\mathbf{CKS}}_{\mathsf{NIKE},par}(\mathsf{A}) \leq \varepsilon$.

Single- and Multi-User Security. The following Theorem from [FHKP13] shows that **CKS-l** and **CKS** are polynomially equivalent and will become useful for our proofs in Section 5.2. The other direction of the Theorem is trivial.

Theorem 3 (CKS-light $\Rightarrow$ CKS [FHKP13, Thm. 1])**.** For any adversary A against NIKE in the *CKS* model, there exists an adversary B that breaks NIKE in the *CKS-light* model such that

$$\mathsf{Adv}^{\mathbf{CKS}}_{\mathsf{NIKE},par}(\mathsf{A}) \leq \frac{Q^2_{\mathtt{RHU}} \cdot Q_{\mathtt{TQ}}}{2} \cdot \mathsf{Adv}^{\mathbf{CKS\text{-}l}}_{\mathsf{NIKE},par}(\mathsf{B}),$$

Furthermore, the following observation will also be useful for our proofs in Section 5.2. Note that a variant of the $\mathbf{CKS}_{\mathsf{NIKE},par}$ game in which the adversary A is only given access to the $\mathtt{RegHonUsr}$ and $\mathtt{TestQue}$ oracles (as well as the random oracle H) can be thought of as a multi-user version of the $\mathbf{HKR\text{-}CKS\text{-}l}_{\mathsf{NIKE},par}$ game. Naturally we call this game $\mathbf{HKR\text{-}CKS}_{\mathsf{NIKE},par}$, as in the work of [FHKP13], and define the advantage of adversary A analogous to the previous definitions. As noted in [FHKP13], Theorem 3 carries over to the HKR setting.

5 Construction

We present our NIKE construction in two steps by introducing a scheme that only satisfies passive security followed by a generic transformation that turns it into a scheme with active security.

5.1 Passive Setting

In this section we present our construction of a non-interactive key exchange with semi malicious correctness that satisfies key indistinguishability for honestly registered public keys (passive security) in the random oracle model. The scheme is depicted in Figure 3. The Gen algorithm generates two independent left and right secret keys sk_L and sk_R along with their respective public keys pk_L and pk_R. Subsequently, the SdK algorithm computes the shared key using $\mathtt{ID}_1$'s pk_L and $\mathtt{ID}_2$'s sk_R when $\mathtt{ID}_1 \leq \mathtt{ID}_2$ and vice versa. By Lemma 1 a random matrix over $\mathcal{R}_q$ will be invertible with overwhelming probability.

Correctness. In order to achieve better bounds in our proof of security, we show that our scheme satisfies both honest correctness as well as the stronger notion of semi-malicious correctness of Definition 3 and Definition 4, respectively. Although Theorem 5 implies Lemma 4, we will use the latter and state its proof in Appendix C for sake of completeness.

Lemma 4 (Honest Correctness)**.** For all (possibly unbounded) adversaries A the non-interactive key exchange $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ construction depicted in Figure 3 has *honest correctness error*

$$\delta \leq \frac{4\beta^2 d^2 N}{q}$$

as per Definition 3.

We show that the scheme satisfies semi-malicious correctness in the quantum random oracle model.

Theorem 5 (**SM-COR** of NIKE)**.** For all (possibly unbounded) adversaries A of depth D making at most Q_{H} queries (possibly in superposition) to the random oracle H, the non-interactive key exchange $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ construction depicted in Figure 3 has *semi-malicious correctness error*

$$\delta(Q_{\mathsf{H}}, D) \leq 16 \cdot (D+2) \cdot (Q_{\mathsf{H}}+1) \cdot \frac{\beta^2 d^2 N}{q}$$

as per Definition 4, where β is a bound on the maximum absolute value of the support of χ.

Proof. We are going to prove that the adversary cannot cause an error in the key derivation, i.e., a mismatch between the derived keys, even if he is allowed to choose both secret keys from the support of the key generation algorithm. This trivially implies semi-malicious correctness. Let (sk_1, pk_1) and (sk_2, pk_2) be the pairs returned by the adversary. Without loss of generality we can consider $sk_1 = sk_L$ and $pk_2 = pk_R$, i.e. only "one side" of the key. A key mismatch occurs whenever

$$\begin{aligned}
\mathtt{Rec}\left(pk_L^\top sk_R + \boldsymbol{r}\right) &\neq \mathtt{Rec}\left(sk_L^\top pk_R + \boldsymbol{r}\right) \\
\mathtt{Rec}\left(\left(\vec{\boldsymbol{s}}_L^\top \boldsymbol{A} + \vec{\boldsymbol{e}}_L^\top\right)\vec{\boldsymbol{s}}_R + \boldsymbol{r}\right) &\neq \mathtt{Rec}\left(\vec{\boldsymbol{s}}_L^\top\left(\boldsymbol{A}\vec{\boldsymbol{s}}_R + \vec{\boldsymbol{e}}_R\right) + \boldsymbol{r}\right) \\
\mathtt{Rec}\big(\underbrace{\vec{\boldsymbol{s}}_L^\top \boldsymbol{A}\vec{\boldsymbol{s}}_R + \boldsymbol{r}}_{\boldsymbol{k}^\star \in \mathcal{R}_q} + \vec{\boldsymbol{e}}_L^\top \vec{\boldsymbol{s}}_R\big) &\neq \mathtt{Rec}\big(\underbrace{\vec{\boldsymbol{s}}_L^\top \boldsymbol{A}\vec{\boldsymbol{s}}_R + \boldsymbol{r}}_{\boldsymbol{k}^\star \in \mathcal{R}_q} + \vec{\boldsymbol{s}}_L^\top \vec{\boldsymbol{e}}_R\big),
\end{aligned}$$

where $\boldsymbol{r}$ is the output of the random oracle on both public keys and $\vec{\boldsymbol{e}}_L$ and $\vec{\boldsymbol{e}}_R$ are sampled from the noise distribution

$\mathsf{Stp}(1^\lambda)$

$\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d+1)$
$\boldsymbol{A} \xleftarrow{\$} \mathsf{GL}_N(\mathcal{R}_q)$
$par := (q, d, \mathcal{R}_q, N, \boldsymbol{A})$
return par

$\mathsf{Gen}(\mathtt{ID})$

$\vec{\boldsymbol{s}}_L, \vec{\boldsymbol{s}}_R \leftarrow \mathtt{Cbd}(\cdot)$ // Samples $\vec{\boldsymbol{s}} \in \mathcal{R}_q^N$ from χ^N
$\vec{\boldsymbol{e}}_L, \vec{\boldsymbol{e}}_R \leftarrow \mathtt{Cbd}(\cdot)$ // Samples $\vec{\boldsymbol{e}} \in \mathcal{R}_q^N$ from χ^N
$sk_L := \vec{\boldsymbol{s}}_L^\top \in \mathcal{R}_q^{1\times N}$
$sk_R := \vec{\boldsymbol{s}}_R \in \mathcal{R}_q^N$
$pk_L := \vec{\boldsymbol{s}}_L^\top \boldsymbol{A} + \vec{\boldsymbol{e}}_L^\top \in \mathcal{R}_q^{1\times N}$
$pk_R := \boldsymbol{A}\vec{\boldsymbol{s}}_R + \vec{\boldsymbol{e}}_R \in \mathcal{R}_q^N$
return $\left(sk_{\mathtt{ID}} := (sk_L, sk_R), pk_{\mathtt{ID}} := (pk_L, pk_R)\right)$

$\mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2)$

if $\mathtt{ID}_1 \leq \mathtt{ID}_2$:
  $\boldsymbol{r} := \mathsf{H}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, pk_2) \in \mathcal{R}_q$
  parse $pk_1 \to (pk_L, \perp) =: \vec{\boldsymbol{u}}_L^\top \in \mathcal{R}_q^{1\times N}$
  parse $sk_2 \to (\perp, sk_R) =: \vec{\boldsymbol{s}}_R \in \mathcal{R}_q^N$
  $\boldsymbol{k}' := \vec{\boldsymbol{u}}_L^\top \vec{\boldsymbol{s}}_R + \boldsymbol{r} \in \mathcal{R}_q$
else :
  $\boldsymbol{r} := \mathsf{H}(\mathtt{ID}_2, pk_2, \mathtt{ID}_1, pk_1) \in \mathcal{R}_q$
  parse $pk_1 \to (\perp, pk_R) =: \vec{\boldsymbol{u}}_R \in \mathcal{R}_q^N$
  parse $sk_2 \to (sk_L, \perp) =: \vec{\boldsymbol{s}}_L^\top \in \mathcal{R}_q^{1\times N}$
  $\boldsymbol{k}' := \vec{\boldsymbol{s}}_L^\top \vec{\boldsymbol{u}}_R + \boldsymbol{r} \in \mathcal{R}_q$
$k := \mathtt{Rec}(\boldsymbol{k}') \in \{0,1\}^d$
return k

$\mathtt{Rec}(\boldsymbol{k})$

for $i \in \{0, \dots, d-1\}$:
  $\mathrm{k_i} := \mathtt{Rnd}(k_i) \in \{0,1\}$
return $k \in \{0,1\}^d$

$\mathtt{Rnd}(k_i)$

if $\frac{q}{4} \leq k_i \leq \frac{3q}{4}$:
  return 1
else :
  return 0

$\mathtt{Cbd}(\cdot)$

for $i \in \{1, \dots, N\}$:
  for $j \in \{0, \dots, d-1\}$:
    $a, b \xleftarrow{\$} \{0,1\}$
    $f_j := a - b$
  $\boldsymbol{f}_i := \sum_{j=0}^{d-1} f_j X^j$
return $\vec{\boldsymbol{f}} := (\boldsymbol{f}_1, \dots, \boldsymbol{f}_N)$

Figure 3: Construction of passively secure non-interactive key exchange $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ with functions $\mathtt{Rec} : \mathcal{R}_q \to \{0,1\}^d$, $\mathtt{Rnd} : \mathbb{Z}_q \to \{0,1\}$ and $\mathtt{Cbd} : \emptyset \to \mathcal{R}_q^N$, and random oracle $\mathsf{H} : \mathcal{IDS} \times \left(\mathcal{R}_q^{1\times N} \times \mathcal{R}_q^N\right) \times \mathcal{IDS} \times \left(\mathcal{R}_q^{1\times N} \times \mathcal{R}_q^N\right) \to \mathcal{R}_q$. Here $\mathsf{GL}_N(\mathcal{R}_q)$ denotes the set of invertible $N \times N$ matrices over $\mathcal{R}_q$.

χ^N. By definition of the $\mathtt{Rec}$ function, this means that the term $\vec{\boldsymbol{e}}_L^\top \vec{\boldsymbol{s}}_R$ (or, equivalently, the term $\vec{\boldsymbol{s}}_L^\top \vec{\boldsymbol{e}}_R$) is causing a rounding error on one out of the d coefficients of $\boldsymbol{k}^\star$. We now bound the size of the largest coefficient of $\vec{\boldsymbol{e}}_L^\top \vec{\boldsymbol{s}}_R$ as

$$\left\| \vec{\boldsymbol{e}}_L^\top \vec{\boldsymbol{s}}_R \right\|_\infty = \left\| \sum_{i=1}^N \boldsymbol{e}_{L,i} \boldsymbol{s}_{R,i} \right\|_\infty \leq \sum_{i=1}^N \|\boldsymbol{e}_{L,i} \boldsymbol{s}_{R,i}\|_\infty \leq \beta^2 dN,$$

where the first inequality follows from the triangle inequality. The norm of $\vec{\boldsymbol{s}}_L^\top \vec{\boldsymbol{e}}_R$ can be bounded similarly. It follows that, in order for a key derivation error to occur, then at least one coefficient of $\boldsymbol{k}^\star$ must be in the following interval

$$S^\star = \left[\frac{q}{4} \pm \beta^2 dN\right] \cup \left[\frac{3q}{4} \pm \beta^2 dN\right].$$

If $\boldsymbol{k}^\star$ followed a uniform distribution, this would then happen with probability at most $\frac{d \cdot |S^\star|}{|\mathbb{Z}_q|} = \frac{4\beta^2 dN}{q}$. Next we define a function F that, on input two public keys and two identities samples a uniform $\boldsymbol{r}$, it returns 1 if a key mismatch occurs, i.e.,

$$\mathtt{Rec}\left(pk_L^\top sk_R + \boldsymbol{r}\right) \neq \mathtt{Rec}\left(sk_L^\top pk_R + \boldsymbol{r}\right)$$

and 0 otherwise. The function checks this by (inefficiently) recovering the secret keys and comparing the results of the $\mathtt{Rec}$ functions (see equation above). Due to the high probability of invertibility for $\boldsymbol{A}$, as stated in Lemma 1, the secret key is uniquely determined by the public key. Hence, this (inefficient) function is well defined on all inputs. Furthermore, note that the element

$$\boldsymbol{k}^\star = sk_L^\top \boldsymbol{A} sk_R + \boldsymbol{r}$$

is uniformly distributed in $\mathcal{R}_q$, since $\boldsymbol{r} \xleftarrow{\$} \mathcal{R}_q$. It follows that for any given input x:

$$\Pr[F(x) = 1] \leq \frac{4\beta^2 d^2 N}{q}.$$

Finally, observe that by definition a key mismatch happens if and only if the function F output 1 and consequently the adversary is able to find such accepting input. By Lemma 2, this happens with probability at most $16 \cdot (D+2) \cdot (Q_{\mathsf{H}}+1) \cdot \beta^2 d^2 N/q$ for an adversary of depth D, making at most Q_{H} quantum queries to the random oracle. ∎

On the Need for Random Oracles. An astute reader may wonder whether the usage of the random oracle is needed at all to prove the above notion of correctness, since there does not appear to be an immediate attack even if we omit the random oracle completely from the scheme. It is plausible to conjecture that semi-malicious correctness holds even without the random oracle. Informally, semi-malicious correctness boils down to showing that, for a given public key $pk \in \mathcal{R}_q^N$, it is hard to find an $\boldsymbol{s} \in \mathcal{R}_q^N$ such that no coefficient of the product $\boldsymbol{s}^\top pk$ lies in the interval $S^\star$. Thus, the bound in these settings would require one to estimate the hardness of this version of the (inhomogenous) 1-dimensional short integer solution (SIS) problem. By relying on the random oracle heuristic, we are able to bypass this problem and obtain a construction in the QROM that is: (i) unconditionally correct in *any ring* and (ii) whose security is based on the well-established M-LWE problem. We leave the precise study of the hardness of this 1-dimensional variant of the SIS problem as ground for future work.

Passive Security. Assuming the hardness of M-LWE, Definition 1, we show that the scheme satisfies *passive security*, Definition 5, in the QROM.

Theorem 6 (Passive Security)**.** For any PPT adversary A against NIKE := (Stp, Gen, SdK), depicted in Figure 5, making an arbitrary number of queries (possibly in superposition) to H, there exists PPT adversaries $\mathsf{B}_1, \mathsf{B}_2$ such that

$$\mathsf{Adv}^{\textbf{HKR-CKS-l}}_{\mathsf{NIKE},par}(\mathsf{A}) \leq \mathsf{Adv}^{\textbf{M-LWE}}_{q,N,N,\chi}(\mathsf{B}_1) + \mathsf{Adv}^{\textbf{M-LWE}}_{q,N,N+1,\chi}(\mathsf{B}_2) + \frac{4\beta d}{q}.$$

Proof of Theorem 6. Let A be an adversary against NIKE in the **HKR-CKS-l** game. Consider the sequence of games in Figure 4 that takes into account both independent left and right halves of the keys.

Games G_0, G_1, G_2, G_3

01 $b \xleftarrow{\$} \{0,1\}$
02 $(sk_1, pk_1) \xleftarrow{\$} \mathsf{Gen}(\mathtt{ID}_1)$
03 $pk_1 = (pk_{1_L}, pk_{1_R}) \xleftarrow{\$} \mathcal{R}_q^{1\times N} \times pk_{1_R}$ // G_1
04 $(sk_2, pk_2) \xleftarrow{\$} \mathsf{Gen}(\mathtt{ID}_2)$
05 $pk_2 = (pk_{2_L}, pk_{2_R}) \xleftarrow{\$} pk_{2_L} \times \mathcal{R}_q^N$ // G_3
06 **if** $\mathtt{ID}_1 \leq \mathtt{ID}_2$:
07 $\quad \boldsymbol{r} := \mathsf{H}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, pk_2) \in \mathcal{R}_q$
08 $\quad$ **parse** $pk_1 \to (pk_L, \perp) =: \vec{\boldsymbol{u}}_L^\top \in \mathcal{R}_q^{1\times N}$
09 $\quad$ **parse** $sk_2 \to (\perp, sk_R) =: \vec{\boldsymbol{s}}_R \in \mathcal{R}_q^N$
10 $\quad \boldsymbol{k}' := \vec{\boldsymbol{u}}_L^\top \vec{\boldsymbol{s}}_R + \boldsymbol{r} \in \mathcal{R}_q$
11 **else** :
12 $\quad \boldsymbol{r} := \mathsf{H}(\mathtt{ID}_2, pk_2, \mathtt{ID}_1, pk_1) \in \mathcal{R}_q$
13 $\quad$ **parse** $pk_1 \to (\perp, pk_R) =: \vec{\boldsymbol{u}}_R \in \mathcal{R}_q^N$
14 $\quad$ **parse** $sk_2 \to (sk_L, \perp) =: \vec{\boldsymbol{s}}_R^\top \in \mathcal{R}_q^{1\times N}$
15 $\quad \boldsymbol{k}' := \vec{\boldsymbol{s}}_R^\top \vec{\boldsymbol{u}}_R + \boldsymbol{r} \in \mathcal{R}_q$
16 $k_0 := \mathtt{Rec}(\boldsymbol{k}') \in \{0,1\}^d$ // G_0
17 $\boldsymbol{e} \xleftarrow{\$} \chi$ // G_2
18 $k_0 := \mathtt{Rec}(\boldsymbol{k}' + \boldsymbol{e}) \in \{0,1\}^d$ // G_2
19 $\boldsymbol{u} \xleftarrow{\$} \mathcal{R}_q$ // G_3
20 $k_0 := \mathtt{Rec}(\boldsymbol{u}) \in \{0,1\}^d$ // G_3
21 $k_1 \xleftarrow{\$} \mathcal{SKS}$
22 $b' \leftarrow \mathsf{A}^{|\mathsf{H}\rangle}(pk_1, pk_2, k_b)$
23 **return** $[\![b = b']\!]$

Figure 4: Games G_0, G_1, G_2, G_3 for the proof of **HKR-CKS-l** of NIKE in Figure 3.

Game G_0 This is the original $\textbf{HKR-CKS-l}_{\mathsf{NIKE},par}$ game so by definition $\Pr\left[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1\right] = \Pr\left[\textbf{HKR-CKS-l}^{\mathsf{A}}_{\mathsf{NIKE},par} \Rightarrow 1\right]$.

Game G_1 Without loss of generality we consider the half of pk_1 that is actually used in the key derivation, say pk_{1_L}. In this game pk_{1_L} is replaced with a uniform key on Line 03. It follows immediately from Definition 1 that

$$\left|\Pr\left[\mathsf{G}_0^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \mathsf{Adv}^{\textbf{M-LWE}}_{q,N,N,\chi}(\mathsf{B}_1).$$

Game G_2 In this hybrid we modify the way we compute the shared key. Consider $\boldsymbol{k}'$ as computed in the SdK algorithm, we define the shared key as $\mathtt{Rec}(\boldsymbol{k}' + \boldsymbol{e})$ where $\boldsymbol{e} \xleftarrow{\$} \chi$ is a freshly sampled ring element from the noise distribution. Note that the adversary can only detect a change in this hybrid if

$$\mathtt{Rec}(\boldsymbol{k}' + \boldsymbol{e}) \neq \mathtt{Rec}(\boldsymbol{k}').$$

Since $\boldsymbol{k}'$ is uniformly distributed in $\mathcal{R}_q$, the probability that any coefficient is rounded to a different term is at most $4\beta d/q$, which is also an upper bound on the distinguishing advantage of the adversary. Thus we get

$$\left|\Pr\left[\mathsf{G}_1^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \frac{4\beta d}{q}.$$

Game G_3 In this game the half of pk_2 that is used in the key derivation, pk_{2_R}, is replaced with a uniform key on Line 05. Furthermore $\boldsymbol{k}' + \boldsymbol{e}$ is replaced with a uniform ring element $\boldsymbol{u}$ on Line 20. By an invocation of the module-LWE assumption we have that

$$\left|\Pr\left[\mathsf{G}_2^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \mathsf{Adv}^{\textbf{M-LWE}}_{q,N,N+1,\chi}(\mathsf{B}_2).$$

Observe that k_0 and k_1 are identically distributed and the adversary can only guess b'. Hence, $\Pr\left[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1\right] = \frac{1}{2}$. Collecting all probabilities yields the bound stated in Theorem 6.

■

Reconciliation Function. The reconciliation function Rec depicted in Figure 3 maps a ring element $\boldsymbol{k} \in \mathcal{R}_q$ to a bit string of length d. In the final step of the proof of Theorem 6 the input to Rec is a uniformly sampled ring element $\boldsymbol{u} \in \mathcal{R}_q$. Each element in $\mathbb{Z}_q$ is rounded to a bit with almost equal probability. More specifically for all $u_i \in \mathbb{Z}_q$, $\Pr[\mathtt{Rnd}(u_i) = 0] = \frac{q-1}{2q}$ and $\Pr[\mathtt{Rnd}(u_i) = 1] = 1 - \frac{q-1}{2q}$, which are negligibly close to $1/2$, for an appropriate choice of q. Although the reconciliation function maps the key to a smaller domain, trying to guess $k := \mathtt{Rec}(\boldsymbol{u})$, even when given both public keys, would, in expectation, require guessing half of all possible keys, which amounts to 2^{d-1}. This becomes practically infeasible for large values of d. For more details on our specific choice of q and d, we refer to Section 5.4.

5.2 Active Setting

Here we show how a non-interactive key exchange with passive security can be generically transformed to one with active security. The transformation, depicted in Figure 5, requires a simulation-sound NIZK with a straight line extractor. Due to space constraints, the proof is deferred to Appendix C.1.

Theorem 7 (HKR-CKS-l and SM-COR of $\mathsf{NIKE}' \overset{\text{QROM}}{\underset{\mathsf{ZKPoK}}{\Rightarrow}}$ CKS of NIKE). Let $\mathsf{H}:\{0,1\}^* \to \mathcal{R}_q$ be a random oracle and $\mathsf{NIKE}' := (\mathsf{Stp}', \mathsf{Gen}', \mathsf{SdK}')$ a passively secure non-interactive key exchange with semi-malicious correctness defined relative to $par' \in \mathsf{Stp}'(1^\lambda)$. Further, let $\mathsf{ZKPoK} := (\mathsf{ZK.Prv}, \mathsf{ZK.Ver})$ be a simulation-sound online extractable zero-knowledge proof of knowledge for the NP relation $R = (pk_{\mathtt{ID}}, sk_{\mathtt{ID}})$. Then, for any **CKS** adversary A against $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$, depicted in Figure 5, there exist PPT adversaries $\mathsf{B}_1, \mathsf{B}_2, \mathsf{B}_3, \mathsf{C}$ such that

$$\mathsf{Adv}^{\mathbf{CKS}}_{\mathsf{NIKE},par}(\mathsf{A}) \leq \frac{Q^2_{\mathtt{RHU}} \cdot Q_{\mathtt{TQ}}}{2} \cdot \varepsilon$$

and

$$\begin{aligned}\varepsilon := Q_{\mathtt{RCU}} \cdot \mathsf{Adv}^{\mathbf{SSND}}_{\mathsf{ZKPoK}}(\mathsf{B}_1) + 2 \cdot \mathsf{Adv}^{\mathbf{SM\text{-}COR}}_{\mathsf{NIKE}',par'}(\mathsf{B}_2) \\ + 2 \cdot \mathsf{Adv}^{\mathbf{ZK}}_{\mathsf{ZKPoK}}(\mathsf{B}_3) + \mathsf{Adv}^{\mathbf{HKR\text{-}CKS\text{-}l}}_{\mathsf{NIKE}',par'}(\mathsf{C}),\end{aligned}$$

where $Q_{\mathtt{RHU}}$, $Q_{\mathtt{RCU}}$, $Q_{\mathtt{RCQ}}$, and $Q_{\mathtt{TQ}}$ denote the number of queries made by A to `RegHonUsr`, `RegCorUsr`, `RevCorQue`, and `TestQue`, respectively.

5.3 Practical considerations

Halving the Key Size. Observe that the "left" and "right" components pk_L and pk_R of the public key of the NIKE as specified in Figure 3 are necessary because we work in the non-commutative M-LWE setting. Note that all theorems, lemmas and proofs consider both the left and right components of keys. An easy way to halve the size of the public key would be to set $N = 1$, i.e., to work in the R-LWE setting; this also eliminates the need for the case distinction in SdK. We argue that for essentially all relevant applications of a NIKE, we can halve the public-key size even *without* moving to the R-LWE setting. All that is required is that protocol participants (and their associated NIKE keys) have different *roles*, typically called initiator and responder or client and server, and that these roles are clear from protocol context. This is certainly the case for the application examples sketched in Section 1.1: The OPTLS handshake, like the TLS handshake, clearly distinguishes the roles of client and server, so does the handshake in (post-quantum) WireGuard. Also in X3DH the critical static-semistatic key exchange has clear roles that can be used to distinguish between the "left" and "right" participant instead of transmitting both components of the key and using comparison of IDs. Note that this setting of a NIKE using keys with different roles is very similar to the ℓ_A and ℓ_B keys of SIDH [JD11, Sec. 3.2], when it was still considered as a replacement for DH, i.e., before it was shown to not be actively secure in [GPST16] and completely broken in [CD22].

Based on these considerations, we stick to the M-LWE setting for the construction of SWOOSH; in our performance evaluation in Section 6 we report the size of only one public-key component.

Security of the NIZK. We highlight that our proof of active security, Theorem 7, requires the strong property of simulation-sound online-extractability. Although constructions satisfying such a strong notion exist [Unr15], they tend to be less efficient than alternatives satisfying weaker notions of security. For instance, a proof of knowledge of an M-LWE secret satisfying simulation soundness, but without *online*-extractability, using state-of-the-art techniques [LNP22] and appropriate parameters is around 89 KB in size. Since the [LNP22] framework is relatively new, there are currently no implementations available to determine the exact running time of the prover and verifier. However, previous versions of lattice-based zero-knowledge proofs [LNS20], which serve as the basis for [LNP22], have shown implementations with prover and verifier running times on the order of milliseconds when proving similar relations. We remark that using NIZKs with slightly weaker security, in favour of a more efficient scheme, is a well-established heuristic and was already used in many works prior to ours, including [CDG+17, DKLs18, DKLs19, GG18, LN18, MMS+19, TCLM21]. However, this limitation is inherent in our approach, and investigating security under an alternative notion remains an interesting (and challenging) question. Tangentially, we also mention that for some applications, the performance of the NIZK does not affect the efficiency of the shared-key computation, since it can be verified once and for all for a given public key: In any scenario where the public keys are distributed by a PKI, the NIZK proof can be simply verified by the PKI upon the registration of the key, and then immediately discarded. The users would then trust the PKI to have verified the NIZK on their behalf. Note that this does not introduce any extra trust assumption, since the PKI is already trusted to provide the correct public key. In these scenarios, the efficiency of the NIZK only marginally impacts the overall system performance, and thus justifies ignoring the costs of the NIZK for shared-key computation.

5.4 Parameter selection

Selecting parameters for the scheme influences several aspects, most notably the correctness error and the hardness

$\underline{\mathsf{Stp}(1^\lambda)}$	$\underline{\mathsf{Gen}(\mathtt{ID})}$	$\underline{\mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2)}$
$par \xleftarrow{\$} \mathsf{Stp}'(1^\lambda)$	$(sk'_{\mathtt{ID}}, pk'_{\mathtt{ID}}) \xleftarrow{\$} \mathsf{Gen}'(\mathtt{ID})$	**parse** $pk_1 \to (pk'_1, \pi)$
return par	$\pi \xleftarrow{\$} \mathsf{ZK.Prv}(pk'_{\mathtt{ID}}, sk'_{\mathtt{ID}})$	**if** $\mathsf{ZK.Ver}(pk'_1, \pi) = 0$: **return** $\perp$
	$sk_{\mathtt{ID}} := sk'_{\mathtt{ID}}$	$k' := \mathsf{SdK}'(\mathtt{ID}_1, pk'_1, \mathtt{ID}_2, sk_2)$
	$pk_{\mathtt{ID}} := (pk'_{\mathtt{ID}}, \pi)$	**return** k'
	return $(sk_{\mathtt{ID}}, pk_{\mathtt{ID}})$	

Figure 5: Compiler for transforming a passively secure non-interactive key exchange $\mathsf{NIKE}' := (\mathsf{Stp}', \mathsf{Gen}', \mathsf{SdK}')$ with semi-malicious correctness into an actively secure non-interactive key exchange $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$.

of M-LWE. In order to evaluate the security of our scheme we use the *Lattice-Estimator* tool [APS15, Pla18, ACD+18a], to estimate the memory and CPU operations required to perform various lattice attacks, including dual attacks, uSVP, the Coded-BKW attack, and solving using Gröbner bases with the Arora-GB attack. The estimator has been used to estimate the concrete security for all LWE and NTRU based candidates of the NIST competition [ACD+18b], and is regularly updated to include the latest developments in lattice cryptanalysis [3]. However, we also take into account practical considerations for the implementation when selecting our parameters, such as the use of ternary secrets and noise sampled from a centred binomial distribution. For our scheme with parameters $n = 8192, q = 2^{214} - 255$ and $\mathcal{X}$ a ternary distribution, we estimate the hardness of the M-LWE problem underlying SWOOSH at 120 bits [4].

The other way to attack SWOOSH is, for an active attacker, to try to produce failures. We consider a quantum attacker with a bounded query depth of $D = 2^{64}$ (i.e., what NIST considers to be "the approximate number of gates that current classical computing architectures can perform serially in a decade" [NIS16, Sec. 4.A]) and a bound on the number of queries of 2^{120} (i.e., matching the hardness of the underlying lattice problem). Applying Theorem 5 yields a success probability (correctness error), after this amount of computation,

$$16 \cdot \left(2^{64} + 2\right) \cdot (2^{120} + 1) \cdot \frac{256^2 \cdot 32}{2^{214}} < \frac{1}{2^4} = \delta(Q_{\mathsf{H}}),$$

i.e., considerably smaller than 1/2. Note that this analysis is conservative as it ignores the circuit depth for the Grover oracle that an attacker would need to implement.

6 Implementation & Performance Evaluation

In order to demonstrate the efficiency of SWOOSH in terms of performance, we implement the core part of the scheme,

[3] An up-to-date list of implemented works can be found https://lattice-estimator.readthedocs.io/en/latest/references.html.

[4] These numbers can be reproduced with the estimator — the version used in this work is at commit 96875622c6b0e6f98a91ddeecaaa17b66dbc5a87.

Parameter	Description	Value
β	upper bound on $\|\vec{s}\|_\infty = \|\vec{e}\|_\infty$	1
q	prime modulus	$2^{214} - 255$
d	dim of $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d + 1)$	256
l	# factors $X^d + 1$ splits into mod q	128
N	height of the $\boldsymbol{A}$ matrix	32
n	lattice dimension	8192
χ	secret / noise distribution	$p(-1) = 25\%$ $p(0) = 50\%$ $p(1) = 25\%$

Table 1: Parameter selection for non-interactive key exchange NIKE.

Passive-SWOOSH, present benchmarks of this implementation, and compare to other KEMs and (pre- and post-quantum) NIKEs. We caution the reader that all implementation details and numbers we present in this section are for Passive-SWOOSH only. To obtain a full picture of the performance of SWOOSH, the implementation will need to be augmented with a future implementation of the NIZK proof from [LNP22]. As outlined in Section 5.3, the performance impact of adding the NIZK proof in terms of both size and computational effort depends on the concrete application scenario and may be negligible if key-generation performance is not critical and if NIZK proof verification can be outsourced to the PKI.

6.1 Implementation

As a NIKE, SWOOSH is composed of two major functions, the key generation procedure and the shared-key computation, the performance of which dictates the efficiency of SWOOSH.

In the case of the key generation, the matrix $\boldsymbol{A}$ is fixed and assumed to be in the NTT domain, so performance is dictated by the sampling of the secret and error vectors, as well as the computation of the public key which involves two NTT transformations, and a matrix multiplication followed by a polynomial addition. On Intel Skylake the cumulative execution time of the forward NTT accounts for $\sim 10\%$ of total key-generation time. This calculation is based on a single transformation taking 217 430 cycles, with

Scheme (variant)	Security	Assumption	Non-interactive	Post-quantum	Size (bytes)		Cycles	
					ct	*pk*	Gen	Encaps + Decaps **or** SdK
CRYSTALS-Kyber (`Kyber-768` [SAB+22])	IND-CCA2	M-LWE	✗	✓	1 088	1 184	200 302	(251 384 + 287 724) 539 108
Classic McEliece (`mceliece348864` [ABC+22])	IND-CCA2	Binary Goppa Codes	✗	✓	96	261 120	46 715 060	(31 000 + 112 178) 143 178
ECDH (`X25519` [Ber06])	HKR-CKS* [5]	CDH	✓	✗	—	32	28 187	87 942
CTIDH (`CTIDH-1024` [BBC+21])	HKR-CKS* [6]	CSIDH	✓	✓	—	128	469 520 000	511 190 000
This work (Passive-SWOOSH)	HKR-CKS	M-LWE	✓	✓	—	221 184	146 920 890	10 612 666

Table 2: Comparison of select post-quantum KEMs and NIKEs.

a total of 64 transformations executed (32 for each secret and error vector). In contrast, the inverse NTT and noise-generation processes require 262 992 and 89 776 cycles, respectively. As for the shared-key computation, its performance is mainly dictated by the random offset computation, which requires the use of cSHAKE [KjCP16] and the polynomial base multiplication required to calculate $\boldsymbol{k}'$ (see Figure 3). Similar to other schemes, the shared-key derivation also performs rounding of the shared key, however its execution time is negligible. At a high level, the architecture of our implementation is divided into two distinct parts: low-level field arithmetic over $\mathbb{F}_q$ that is implemented using the Jasmin language [ABB+17, ABB+20], and polynomial arithmetic in $\mathcal{R}_q$ as well as the scheme itself, both of which are implemented in Rust.

The structure largely mimics the abstract specification in Figure 3. The main difference is that, like other lattice-based schemes [ADPS16, SAB+22], we encode and transmit public keys in NTT domain. This massively reduces the number of cycles required for shared-key computation. In addition, as discussed in Section 5.3, we assume that the role of each party is well defined and thus only compute one half of the key. Finally, we implement the noise sampling in a slightly different way than one might expect; we will discuss this later in this section.

Zooming in on the low-level field arithmetic, the operations on integers modulo $2^{214} - 255$ require multiple-precision integers since native 64-bit registers are not large enough to store a single field element. This arithmetic is implemented through libjbn [7], a Jasmin library that exposes (modular) big-integer arithmetic.

Polynomial Arithmetic. On top of this layer, operations in polynomial rings are implemented using Rust, in addition to other functions such as reconciliation, matrix and noise generation. Similar to other lattice-based schemes, one of the more critical (and easier) operations to optimise (from a performance perspective) is polynomial multiplication. The naive algorithm for multiplying two polynomials in $\mathcal{R}_q$, sometimes called Schoolbook multiplication, involves multiplying all pairs of coefficients, calculating their sum and reducing modulo $X^d + 1$. However, the complexity of this approach is quadratic in the number of coefficients and thus quite costly.

The *Number Theoretic Transform* (NTT) provides a more efficient approach for polynomial multiplication with quasi-logarithmic time complexity $O(d \log(d))$ instead of $O(d^2)$. For a detailed discussion on the NTT refer to [Sei18].

As is the case for other implementations [SAB+22, ADPS16], we use an in-place NTT which requires bit-reversal operations in the forward and inverse transforms but uses less memory. Another optimisation is to make the NTT a part of our scheme, which means that the matrix $\boldsymbol{A}$ is sampled in the NTT domain, and the secret and public keys are stored in the NTT domain. This results in the NTT only being used three times, once for the shared-key derivation and twice in the key generation to transform the secret and error vectors, which are sampled in the normal domain, to the NTT domain before computing the public key. A common trick to speed-up the NTT transformation when using Montgomery reduction [Mon85], as is the case for libjbn, is to use pre-computed constants in Montgomery form $\zeta \cdot R \pmod q$.

Noise Sampling and Matrix Generation. Both the matrix generation and noise-sampling procedures use a seed, either set as a system parameter for $\boldsymbol{A}$ or as a secret input to a PRG in the case of $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{e}}$, to produce a stream bytes from which the distributions are sampled. In the case of matrix generation this is achieved via rejection sampling on the stream of bytes produced by an extendable output function (XOF). The noise-sampling procedure, used for generating the secret key and the error vector, samples these vectors from a centred binomial distribution using the output of a PRF with a random seed. As with other schemes where multiplication is optimised using the NTT, the choice of (symmetric) primitive that underlies these functions tends to be a deciding factor for the performance. We chose cSHAKE [KjCP16] based on Keccak [Dwo15] as the underlying primitive for the XOF and AES256-CTR for the PRF used in noise sampling.

Similar to the NewHope scheme [ADPS16], for efficiency reasons the secret and error vectors are sampled from a centred binomial distribution rather than a discrete Gaussian distribution. Using ternary noise means that each coefficient can be generated from only 2 bits and thus, the generation of a polynomial in $\mathcal{R}_q$ only requires

[5] See Section 6.2.

[6] See Section 6.2.

[7] See `https://github.com/formosa-crypto/libjbn`.

$(32 \cdot 256 \cdot 2)/8 = 2048$ (pseudo-random) bytes. Intuitively, our CBD definition in Figure 3 when a and b are sourced from a PRG, maps 00_b and 11_b to $0 \mod q$ with 50% probability, 10_b to $1 \mod q$ and 01_b to $-1 \mod q$ with 25% probability each. Our implementation differs from the specification by applying signed reduction modulo 3 to each two bit block and converting it to a congruent value in $\mathbb{F}_q$, as opposed to using big-integer field arithmetic to map bits a and b to an element in $\mathbb{F}_q$. Although this approach produces a different mapping (11_b to $-1 \mod q$, 00_b and 10_b to $0 \mod q$ and 01_b to $1 \mod q$), the distribution of the outputs is identical. Due to the size of our field elements, this approach results in a considerable speed up in the noise sampling. The random offset used in our scheme is generated by performing rejection sampling on the output of cSHAKE-256 [KjCP16].

6.2 Performance and Security Evaluation

In this section we evaluate the performance of our scheme and compare it to other NIKEs and KEMs. Additionally, we present a comparison of key sizes and the properties of each scheme such as post-quantum security, and whether they are non-interactive, and the security notions they satisfy.

Security Models. To set a common ground for comparison, we note that the SdK algorithm in a NIKE can be viewed as simultaneously running both Encaps and Decaps of a KEM. KEMs should satisfy the standard security notions of IND-CCA security, with schemes such as CRYSTALS-Kyber [SAB+22] satisfying indistinguishability under *adaptive* chosen ciphertext attacks (IND-CCA2 security). On the other hand, we implemented the passively-secure version of our NIKE. We argue that this comparison is still meaningful because, in the context of a Public Key Infrastructure (PKI), NIZK verification can be performed once and for all by the PKI. Therefore, the runtime of our scheme is primarily influenced by the operations of the passively-secure NIKE.

We also note that the *"unhashed"* version of ECDH (`X25519` [Ber06]), for which we report benchmarks, is insecure in the (DKR-)CKS model. In an unhashed NIKE where shared keys aren't linked to specific identities, an adversary, A, can carry out a two-step attack: (i) A registers the public key g^a belonging to an honest party, Alice, as if it were their own key ($\texttt{RegCorUsr}(\texttt{ID} = \mathsf{A}, pk = g^a)$). (ii) Next, A requests the shared key between themselves and another honest party, say Bob, who possesses a public key g^b ($\texttt{RevCorQue}(\texttt{ID} = \mathit{Bob}, \texttt{ID} = \mathsf{A}) \rightarrow g^{ab}$). This immediately gives the shared key g^{ab} between Alice and Bob. Due to the homomorphic properties of (EC)DH, this attack remains effective (requiring only a minor modification) even if A is prohibited from registering keys of existing users. On the other hand, a hashed version of (EC)DH can be shown to be DKR-CKS secure in the ROM under the *Strong Computational Diffie-Hellman* assumption.

Although there is no proof of CTIDH (`CTIDH-1024` [BBC+21]) in the CKS model, the unhashed variant (excluding identities) of CTIDH faces similar security issues for the same reason. Recent work [DHK+22] proved that achieving actively secure NIKE from CSIDH inherently depends on *"very strong variants of the Group Action Strong CDH"* problem.

Nevertheless, it is important to note that Passive-SWOOSH is indeed strictly weaker than unhashed DH, as a key-recovery attack becomes possible if the adversary is allowed to register corrupted keys. Furthermore, from a practical point of view, enhancing the security of ECDH and CTIDH does not come with a significant increase in costs, as the computation time for hashing is still minimal, compared to the dominating modular-arithmetic cycle counts.

Benchmarks. The benchmark results for Passive-SWOOSH were obtained on an Intel Core i7-6500U (Skylake) running on a single core with Hyper-threading and TurboBoost disabled. The Rust compiler version used for the benchmarks was 1.62.1[8] and the Jasmin compiler version was 2022.09.0. We report the median cycle counts of 10000 runs. In Table 2 we list the results and compare to the cycle counts of Kyber-768 (on the same hardware), mceliece348864 as reported in [BL23], CTIDH-1024 as reported in [BBC+21, Sec. 8], and of lib25519 [NB22], on Intel Skylake CPUs.

As expected, the pre-quantum X25519 [Ber06] scheme is orders of magnitude faster than Passive-SWOOSH for key generation. However, in many applications of NIKEs, keys are re-used many times and what is more critical is the performance of shared-key computation. Here the gap to pre-quantum X25519 is considerably smaller and Passive-SWOOSH outperforms the only real post-quantum competitor CTIDH by a factor of 48. Compared to post-quantum KEMs, such as Kyber-768 and mceliece348864, the performance gap increases to several orders of magnitude, similarly to X25519. However, these schemes require further rounds of interaction, which not only presents additional overhead, but also means they cannot be used as drop-in replacements for a NIKE.

Additionally, as shown in Table 2, CTIDH, Kyber, and X25519 have a public key size several orders of magnitude smaller than Passive-SWOOSH. In this aspect, only Classic McEliece has a public key size comparable to that of Passive-SWOOSH, even when taking into account the expected size of the proof of knowledge (see Section 5.3).

[8] The following build configuration options/values were used: `opt-level=3` and `target-cpu="native"`.

7 Conclusions

In this work, we constructed a NIKE based on the M-LWE problem, with a proof in the QROM. Our scheme is based on the standard blueprint, but with an additional twist to guarantee provable security for arbitrary rings. Our optimised implementation shows that our scheme offers reasonable computational performance and key sizes that should be acceptable for most applications. We view our work as the first evidence *contradicting* the folklore belief that lattice-based NIKE is too inefficient to be used in practice. As future work, we plan to explore applications of our scheme to more complex protocols and to formally verify the correctness of (parts of) our implementation.

8 Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG, German research Foundation) as part of the Excellence Strategy of the German Federal and State Governments – EXC 2092 CASA - 390781972; the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM research hub under grant number 16KISK038; and the European Commission through the ERC Starting Grant 805031 (EPOQUE). We also thank the anonymous reviewers for their helpful feedback.

References

[ABB+17] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Arthur Blot, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Hugo Pacheco, Benedikt Schmidt, and Pierre-Yves Strub. Jasmin: High-assurance and high-speed cryptography. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1807–1823, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3134078. (Cited on pages 3 and 13.)

[ABB+20] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Adrien Koutsos, Vincent Laporte, Tiago Oliveira, and Pierre-Yves Strub. The last mile: High-assurance and high-speed cryptographic implementations. In *2020 IEEE Symposium on Security and Privacy*, pages 965–982, San Francisco, CA, USA, May 18–21, 2020. IEEE Computer Society Press. doi:10.1109/SP40000.2020.00028. (Cited on pages 3 and 13.)

[ABC+22] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions. (Cited on pages 3 and 13.)

[ACD+18a] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! In Dario Catalano and Roberto De Prisco, editors, *SCN 18: 11th International Conference on Security in Communication Networks*, volume 11035 of *Lecture Notes in Computer Science*, pages 351–367, Amalfi, Italy, September 5–7, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-98113-0_19. (Cited on page 12.)

[ACD+18b] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! Cryptology ePrint Archive, Report 2018/331, 2018. https://eprint.iacr.org/2018/331. (Cited on page 12.)

[ADH+22] Yawning Angel, Benjamin Dowling, Andreas Hülsing, Peter Schwabe, and Florian Weber. Post quantum noise. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022: 29th Conference on Computer and Communications Security*, pages 97–109, Los Angeles, CA, USA, November 7–11, 2022. ACM Press. doi:10.1145/3548606.3560577. (Cited on page 2.)

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association. (Cited on pages 1, 6, and 13.)

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-26951-7_10. (Cited on page 6.)

[AJL17] Reza Azarderakhsh, David Jao, and Christopher Leonardi. Post-quantum static-static key agreement using multiple protocol instances. In Carlisle Adams and Jan Camenisch, editors, *SAC 2017: 24th Annual International Workshop on Selected Areas in Cryptography*, volume 10719 of *Lecture Notes in Computer Science*, pages 45–63, Ottawa, ON, Canada, August 16–18, 2017. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-72565-9_3. (Cited on page 3.)

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. https://eprint.iacr.org/2015/046. (Cited on page 12.)

[BBC+21] Gustavo Banegas, Daniel J. Bernstein, Fabio Campos, Tung Chou, Tanja Lange, Michael Meyer, Benjamin Smith, and Jana Sotáková. CTIDH: faster constant-time CSIDH. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(4):351–387, 2021. https://tches.iacr.org/index.php/TCHES/article/view/9069. doi:10.46586/tches.v2021.i4.351-387. (Cited on pages 3, 13, and 14.)

[BBCT22] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, and Nicola Tuveri. OpenSSLNTRU: Faster post-quantum TLS key exchange. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022: 31st USENIX Security Symposium*, pages 845–862, Boston, MA, USA, August 10–12, 2022. USENIX Association. (Cited on page 2.)

[BC18] Olivier Blazy and Céline Chevalier. Non-interactive key exchange from identity-based encryption. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3230833.3230864. (Cited on page 3.)

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1006–1018, Vienna, Austria, October 24–28, 2016. ACM Press. doi:10.1145/2976749.2978425. (Cited on page 1.)

[BCGP08] Colin Boyd, Yvonne Cliff, Juan González Nieto, and Kenneth G. Paterson. Efficient one-round key exchange in the standard model. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, *ACISP 08: 13th Australasian Conference on Information Security and Privacy*, volume 5107 of *Lecture Notes in Computer Science*, pages 69–83, Wollongong, Australia, July 7–9, 2008. Springer, Heidelberg, Germany. (Cited on page 1.)

[BCNS15] Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570, San Jose, CA, USA, May 17–21, 2015. IEEE Computer Society Press. doi:10.1109/SP.2015.40. (Cited on pages 2 and 6.)

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-25385-0_3. (Cited on page 6.)

[Ber06] Daniel J. Bernstein. Curve25519: New Diffie-Hellman speed records. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228, New York, NY, USA, April 24–26, 2006. Springer, Heidelberg, Germany. doi:10.1007/11745853_14. (Cited on pages 13 and 14.)

[BFG+20] Jacqueline Brendel, Marc Fischlin, Felix Günther, Christian Janson, and Douglas Stebila. Towards post-quantum security for Signal's X3DH handshake. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *SAC 2020: 27th Annual International Workshop on Selected Areas in Cryptography*, volume 12804 of *Lecture Notes in Computer Science*, pages 404–430, Halifax, NS, Canada (Virtual Event), October 21-23, 2020. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-81652-0_16. (Cited on page 2.)

[BFG+22] Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila. Post-quantum asynchronous deniable key exchange and the signal handshake. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022: 25th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 13178 of *Lecture Notes in Computer Science*, pages 3–34, Virtual Event, March 8–11, 2022. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-97131-1_1. (Cited on page 2.)

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th Annual ACM Symposium on Theory of Computing*, pages 103–112, Chicago, IL, USA, May 2–4, 1988. ACM Press. doi:10.1145/62212.62222. (Cited on page 25.)

[BHLY16] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems – CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 323–345, Santa Barbara, CA, USA, August 17–19, 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-53140-2_16. (Cited on page 6.)

[BL23] Daniel J. Bernstein and Tanja Lange. eBACS: ECRYPT benchmarking of cryptographic systems, 2023. https://bench.cr.yp.to/results-kem.html. (Cited on page 14.)

[BLMP19] Daniel J. Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny. Quantum circuits for the CSIDH: Optimizing quantum evaluation of isogenies. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 409–441, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-17656-3_15. (Cited on page 3.)

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press. doi:10.1145/2488608.2488680. (Cited on page 6.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. doi:10.1145/168588.168596. (Cited on pages 6 and 25.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. doi:10.1007/11761679_25. (Cited on page 5.)

[BS20] Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 493–522, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-45724-2_17. (Cited on page 3.)

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 480–499, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-44371-2_27. (Cited on page 3.)

[CD22] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Report 2022/975, 2022. https://eprint.iacr.org/2022/975. (Cited on page 11.)

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-30589-4_15. (Cited on page 3.)

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1825–1842, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3133997. (Cited on page 11.)

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 127–145, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-78967-3_8. (Cited on pages 4, 6, 7, and 8.)

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-03332-3_15. (Cited on page 3.)

[CLN16] Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 572–601, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-53018-4_21. (Cited on page 3.)

[DG22] Samuel Dobson and Steven D. Galbraith. Post-quantum signal key agreement from sidh. In Jung Hee Cheon and Thomas Johansson, editors, *Post-Quantum Cryptography*, pages 422–450, Cham, 2022. Springer International Publishing. (Cited on page 2.)

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. doi:10.1109/TIT.1976.1055638. (Cited on page 1.)

[DHK+22] Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Group action key encapsulation and non-interactive key exchange in the QROM. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 36–66, Taipei, Taiwan, December 5–9, 2022. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-22966-4_2. (Cited on pages 5 and 14.)

[dK18] Bor de Kock. A non-interactive key exchange based on ring-learning with errors. Master's thesis, Master's thesis, Eindhoven University of Technology, 2018. (Cited on page 3.)

[DKLs18] Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Secure two-party threshold ECDSA from ECDSA assumptions. In *2018 IEEE Symposium on Security and Privacy*, pages 980–997, San Francisco, CA, USA, May 21–23, 2018. IEEE Computer Society Press. doi:10.1109/SP.2018.00036. (Cited on page 11.)

[DKLs19] Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy*, pages 1051–1066, San Francisco, CA, USA, May 19–23, 2019. IEEE Computer Society Press. doi:10.1109/SP.2019.00024. (Cited on page 11.)

[Dwo15] Morris J. Dworkin. SHA-3 standard: Permutation-based hash and extendable-output functions. Technical report, National Institute of Standards and Technology, July 2015. doi:10.6028/nist.fips.202. (Cited on page 13.)

[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. https://eprint.iacr.org/2012/688. (Cited on page 2.)

[EFGT17] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1857–1874, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3134028. (Cited on page 6.)

[FHKP13] Eduarda S. V. Freire, Dennis Hofheinz, Eike Kiltz, and Kenneth G. Paterson. Non-interactive key exchange. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 254–271, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-36362-7_17. (Cited on pages 1, 4, 6, 7, and 8.)

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany. doi:10.1007/3-540-48405-1_34. (Cited on pages 2 and 3.)

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. doi:10.1007/s00145-011-9114-1. (Cited on page 2.)

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany. doi:10.1007/3-540-47721-7_12. (Cited on pages 2 and 25.)

[GdKQ+23] Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. Swoosh: Efficient lattice-based non-interactive key exchange. Cryptology ePrint Archive, Report 2023/271, 2023. https://eprint.iacr.org/2023/271. (Cited on page 1.)

[GdKQ+24] Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. Swoosh: Efficient lattice-based non-interactive key exchange. In *33rd USENIX Security Symposium (USENIX Security 24)*, Philadelphia, PA, August 2024. USENIX Association. (Cited on page 1.)

[GG18] Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 1179–1194, Toronto, ON, Canada, October 15–19, 2018. ACM Press. doi:10.1145/3243734.3243859. (Cited on page 11.)

[GKRS20] Siyao Guo, Pritish Kamath, Alon Rosen, and Katerina Sotiraki. Limits on the efficiency of (ring) LWE based non-interactive key exchange. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 374–395, Edinburgh, UK, May 4–7, 2020. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-45374-9_13. (Cited on pages 1 and 3.)

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press. doi:10.1145/22145.22178. (Cited on pages 25 and 26.)

[GPST16] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-53887-6_3. (Cited on pages 3 and 11.)

[HKKP21] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski, and Thomas Prest. An efficient and generic construction for signal's handshake (X3DH): Post-quantum, state leakage secure, and deniable. In Juan Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12711 of *Lecture Notes in Computer Science*, pages 410–440, Virtual Event, May 10–13, 2021. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-75248-4_15. (Cited on page 2.)

[HNS+21] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Philip R. Zimmermann. Post-quantum WireGuard. In *2021 IEEE Symposium on Security and Privacy*, pages 304–321, San Francisco, CA, USA, May 24–27, 2021. IEEE Computer Society Press. doi:10.1109/SP40001.2021.00030. (Cited on page 2.)

[JAC+17] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. SIKE. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions. (Cited on page 3.)

[JD11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34, Tapei, Taiwan, November 29 – December 2 2011. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-25405-5_2. (Cited on pages 3 and 11.)

[KjCP16] John Kelsey, Shu jen Change, and Ray Perlner. SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash. Technical report, National Institute of Standards and Technology, December 2016. doi:10.6028/nist.sp.800-185. (Cited on pages 13 and 14.)

[KV19] Kris Kwiatkowski and Luke Valenta. The TLS post-quantum experiment. Post on the Cloudflare blog, 2019. https://blog.cloudflare.com/the-tls-post-quantum-experiment/. (Cited on page 2.)

[KW16] Hugo Krawczyk and Hoeteck Wee. The OPTLS protocol and TLS 1.3. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 81–96, Saarbruecken, Germany, March 2016. IEEE. doi:10.1109/eurosp.2016.18. (Cited on page 2.)

[Lan16] Adam Langley. CECPQ1 results. Blog post, 2016. https://www.imperialviolet.org/2016/11/28/cecpq1.html. (Cited on page 2.)

[Lan18] Adam Langley. CECPQ2. Blog post, 2018. https://www.imperialviolet.org/2018/12/12/cecpq2.html. (Cited on page 2.)

[LDK+22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on page 1.)

[LN18] Yehuda Lindell and Ariel Nof. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 1837–1854, Toronto, ON, Canada, October 15–19, 2018. ACM Press. doi:10.1145/3243734.3243788. (Cited on page 11.)

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 71–101, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-15979-4_3. (Cited on pages 3, 11, and 12.)

[LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020: 27th Conference on Computer and Communications Security*, pages 1051–1070, Virtual Event, USA, November 9–13, 2020. ACM Press. doi:10.1145/3372297.3417894. (Cited on pages 3 and 11.)

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015. (Cited on page 6.)

[Lyu17] Vadim Lyubashevsky. Converting newhope/lwe key exchange to a diffe-hellman-like algorithm. Crypto Stack Exchange, 2017. [Online:] https://crypto.stackexchange.com/questions/48146/converting-newhope-lwe-key-exchange-to-a-diffe-hellman-like-algorithm. URL: https://crypto.stackexchange.com/questions/48146/converting-newhope-lwe-key-exchange-to-a-diffe-hellman-like-algorithm. (Cited on pages 2 and 3.)

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-30589-4_16. (Cited on page 3.)

[MMS+19] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. Anonymous multi-hop locks for blockchain scalability and interoperability. In *ISOC Network and Distributed System Security Symposium – NDSS 2019*, San Diego, CA, USA, February 24–27, 2019. The Internet Society. (Cited on page 11.)

[Mon85] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, 1985. (Cited on page 13.)

[MP16a] Moxie Marlinspike and Trevor Perrin. The double ratchet algorithm, 2016. URL: https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf. (Cited on page 1.)

[MP16b] Moxie Marlinspike and Trevor Perrin. The X3DH key agreement protocol (revision 1). Part of the Signal Protocol Documentation, 2016. https://signal.org/docs/specifications/x3dh/x3dh.pdf. (Cited on pages 1 and 2.)

[NB22] Kaushik Nath and Daniel J. Bernstein. lib25519, 2022. https://lib25519.cr.yp.to/. (Cited on page 14.)

[NIS16] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf. (Cited on page 12.)

[PBY17] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. To BLISS-B or not to be: Attacking strongSwan's implementation of post-quantum signatures. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1843–1855, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3134023. (Cited on page 6.)

[Pei20] Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 463–492, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-45724-2_16. (Cited on page 3.)

[Per18] Trevor Perrin. Noise protocol framework, 2018. https://noiseprotocol.org/noise.pdf (Revision 34 vom 2018-07-11). (Cited on page 1.)

[PFH+22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on page 1.)

[Pla18] Rachel Player. *Parameter selection in lattice-based cryptography*. PhD thesis, Royal Holloway, University of London, 2018. (Cited on page 12.)

[PST20] Christian Paquin, Douglas Stebila, and Goutam Tamvada. Benchmarking post-quantum cryptography in TLS. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 72–91, Paris, France, April 15–17, 2020. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-44223-1_5. (Cited on page 2.)

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press. doi:10.1145/1060590.1060603. (Cited on pages 1 and 6.)

[Res18] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018. URL: https://rfc-editor.org/rfc/rfc8446.txt, doi:10.17487/RFC8446. (Cited on page 1.)

[Rob23] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503, Lyon, France, April 23–27, 2023. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-30589-4_17. (Cited on page 3.)

[SAB+22] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. (Cited on pages 1, 6, 13, and 14.)

[Sei18] Gregor Seiler. Faster AVX2 optimized NTT multiplication for ring-LWE lattice cryptography. Cryptology ePrint Archive, Report 2018/039, 2018. https://eprint.iacr.org/2018/039. (Cited on page 13.)

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press. doi:10.1109/SFCS.1994.365700. (Cited on page 1.)

[SSKF21] Sara Stadler, Vitor Sakaguti, Harjot Kaur, and Anna Lena Fehlhaber. Hybrid signal protocol for post-quantum email encryption. Cryptology ePrint Archive, Report 2021/875, 2021. https://eprint.iacr.org/2021/875. (Cited on page 2.)

[SSW20] Peter Schwabe, Douglas Stebila, and Thom Wiggers. Post-quantum TLS without handshake signatures. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020: 27th Conference on Computer and Communications Security*, pages 1461–1480, Virtual Event, USA, November 9–13, 2020. ACM Press. doi:10.1145/3372297.3423350. (Cited on page 2.)

[TCLM21] Sri Aravinda Krishnan Thyagarajan, Guilhem Castagnos, Fabien Laguillaumie, and Giulio Malavolta. Efficient CCA timed commitments in class groups. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021: 28th Conference on Computer and Communications Security*, pages 2663–2684, Virtual Event, Republic of Korea, November 15–19, 2021. ACM Press. doi:10.1145/3460120.3484773. (Cited on page 11.)

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 755–784, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-46803-6_25. (Cited on pages 11, 25, and 26.)

[VV22] Viktoriya V. Vysotskaya and Lev I. Vysotsky. Invertible matrices over some quotient rings: identification, generation, and analysis. *Discrete Mathematics and Applications*, 32(4):263–278, 2022. URL: https://doi.org/10.1515/dma-2022-0022 [cited 2023-06-06], doi:doi:10.1515/dma-2022-0022. (Cited on page 24.)

[WR19] Bas Westerbaan and Cefan Daniel Rubin. Defending against future threats: Cloudflare goes post-quantum. Post on the Cloudflare blog, 2019. https://blog.cloudflare.com/post-quantum-for-all/. (Cited on page 2.)

A Proofs for Section 3 (Preliminaries)

Lemma 1 (Invertibility of Random Matrices)**.** For $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$ with $d = 256$ and $q = 2^{214} - 255$, if $\boldsymbol{A}$ is a random matrix sampled from $\mathcal{R}_q^{N\times N}$, then the probability of $\boldsymbol{A}$ being invertible, denoted by $\Pr[\boldsymbol{A} \in \mathsf{GL}_N(\mathcal{R}_q)]$, satisfies

$$\Pr[\boldsymbol{A} \in \mathsf{GL}_N(\mathcal{R}_q) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{N\times N}] \geq \left(1 - \frac{128}{q^2}\right)^N.$$

Proof. Let $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d+1)$ and let $\eta(f(X))$ denote the set of irreducible proper divisors of the polynomial $f(X) = X^d + 1 \in \mathbb{Z}_q$. Then

$$\begin{aligned}
&\Pr[\boldsymbol{A} \in \mathsf{GL}_N(\mathcal{R}_q) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{N\times N}] \\
&\quad= \frac{\mathrm{ord}(\mathsf{GL}_N(\mathcal{R}_q))}{q^{N^2\cdot \deg(f)}} \\
&\quad\geq \prod_{k=0}^{N-1}\left(1 - q^{(k-N)\cdot\deg(f)} \sum_{\alpha\in\eta(f)} q^{(N-k)\cdot(\deg(f)-\deg(\alpha))}\right) \\
&\quad= \prod_{k=0}^{N-1}\left(1 - q^{(k-N)\cdot d}\, 128 q^{(N-k)\cdot(d-2)}\right) \\
&\quad= \prod_{k=0}^{N-1}\left(1 - 128 q^{2(k-N)}\right) \\
&\quad= \prod_{k=0}^{N-1}\left(\frac{q^{2N} - 128q^{2k}}{q^{2N}}\right) \\
&\quad= \left(\frac{1}{q^{2N^2}}\left(\prod_{k=0}^{N-1}\left(q^{2N} - 128q^{2k}\right)\right)\right) \\
&\quad\geq \frac{1}{q^{2N^2}}\left(q^{2N} - 128q^{2(N-1)}\right)^N \\
&\quad= \left(1 - \frac{128}{q^2}\right)^N,
\end{aligned}$$

where the first inequality follows from Lemma 8. ■

Lemma 8 (Linear Independence of Vectors [VV22, Lem. 1])**.** Let $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d+1)$ and let $\eta(f(X))$ denote the set of irreducible proper divisors of the polynomial $f(X) = X^d + 1 \in \mathbb{Z}_q$. For linearly independent vectors $\vec{\boldsymbol{u}}_1,\dots,\vec{\boldsymbol{u}}_k$ where $\vec{\boldsymbol{u}}_i \in \mathcal{R}_q^N$ and $0 \leq k \leq N-1$, the number of vectors $\vec{\boldsymbol{v}} \in \mathcal{R}_q^N$ such that $\{\vec{\boldsymbol{u}}_1,\dots,\vec{\boldsymbol{u}}_k,\vec{\boldsymbol{v}}\}$ is linearly independent is at most

$$\sum_{\alpha\in\eta(f)} q^{k\deg(f)} q^{(N-k)\deg(\alpha)}.$$

Proof sketch. We estimate the maximum number of vectors $\vec{\boldsymbol{v}} \in \mathcal{R}_q^N$ for which there exists a nonzero tuple of coefficients $\alpha_1,\dots,\alpha_k,\alpha \in \mathcal{R}_q^{k+1}$ such that $\alpha_1\vec{\boldsymbol{u}}_1 + \cdots + \alpha_k\vec{\boldsymbol{u}}_k = \alpha\vec{\boldsymbol{v}}$. Consider the set $V_\alpha := \{\vec{\boldsymbol{v}} \in \mathcal{R}_q^N : \alpha\vec{\boldsymbol{v}} \in \mathcal{L}(\vec{\boldsymbol{u}}_1,\dots,\vec{\boldsymbol{u}}_k)\}$ for $\alpha \in \mathcal{R}_q$, where $\mathcal{L}(\vec{\boldsymbol{u}}_1,\dots,\vec{\boldsymbol{u}}_k)$ denotes the linear hull of vectors $\vec{\boldsymbol{u}}_1,\dots,\vec{\boldsymbol{u}}_k$. Note that $V_\alpha = V_{\alpha\beta}$ for any invertible $\beta \in \mathcal{R}_q$. Therefore the set of all different values of α may be split into equivalence relation $\alpha' \sim \alpha'' \iff (\exists$ an invertible $\beta : \alpha' = \beta\alpha'')$. Finally, one can show that

$$\begin{aligned}
|V_\alpha| &\leq \left|\mathbb{Z}_q[X]/(f/\alpha)\right|^k \cdot (q^{\deg\alpha})^N \\
&= q^{k\deg(f/\alpha)} q^{N\deg(\alpha)} \\
&= q^{k\deg f} q^{(N-k)\deg\alpha}.
\end{aligned}$$

■

Qubits, n-qubit States and Measurement. A qubit $|x\rangle := \alpha_0|0\rangle + \alpha_1|1\rangle$ is a unit vector in some Hilbert space $\mathcal{H}$. When $\alpha_0 \neq 1$ and $\alpha_1 \neq 1$, we say that $|x\rangle$ is in *superposition*. An n-bit quantum register $|x\rangle := \sum_{i=0}^{2^n-1}\alpha_i|i\rangle$ is a unit vector in $\mathcal{H}^{\otimes n} \cong \mathbb{C}^{2^n}$, that is $\sum_{i=0}^{2^n-1}|\alpha_i|^2 = 1$ for $\alpha_i \in \mathbb{C}$. We call the set $\{|0\rangle,|1\rangle,\dots,|2^n-1\rangle\}$ the *computational basis* and say that $|x\rangle$ is *entangled* when $|x\rangle$ cannot be written as the tensor product of single qubits. Unless otherwise stated, measurements are done in the computational basis. After measuring a quantum register $|x\rangle = \sum_{i=0}^{2^n-1}\alpha_i|i\rangle$ in the computational basis, the state *collapses* and $|x\rangle = \pm|i\rangle$ with probability $|\alpha_i|^2$.

Quantum Algorithms. A quantum algorithm A is a sequence of unitary operations U_i, where unitary operations are defined to map unit vectors to unit vectors, whilst preserving the normalisation constraint of quantum registers. A quantum oracle algorithm A^{O} is defined analogously, and can additionally query the oracle O before (or after) executing a unitary U_i. As quantum computations need to be reversible, we model an oracle $\mathsf{O} : X \to Y$ by a unitary U_{O} that maps $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus \mathsf{O}(x)\rangle$. For an oracle O, we write $|\mathsf{O}\rangle$ to denote that an algorithm has quantum-access to U_{O}.

B Proofs for Section 4 (Definitions)

Game HKR-CKS-I$_{\mathsf{NIKE},par}$

```
01  b ←$ {0,1}
02  b' ← A^{|H⟩,RegHonUsr(·),TestQue(·,·)}
03  return ⟦b = b'⟧
```

Figure 6: Game defining **HKR-CKS-I**$_{\mathsf{NIKE},par}$ for a non-interactive key exchange NIKE relative to a random oracle H with adversary A. `RegHonUsr` and `TestQue` are defined in Figure 7.

Game $\mathbf{CKS}_{\mathsf{NIKE},par}$ / $\mathbf{CKS\text{-}l}_{\mathsf{NIKE},par}$

$b \xleftarrow{\$} \{0,1\}$
$\mathcal{D} := \perp$
$\mathcal{K} := \perp$
$b' \leftarrow \mathsf{A}^{|\mathsf{H}\rangle,\ \mathtt{RegHonUsr}(\cdot),\ \mathtt{RegCorUsr}(\cdot,\cdot),\ \mathtt{RevCorQue}(\cdot,\cdot),\ \mathtt{TestQue}(\cdot,\cdot)}$
return $[\![b = b']\!]$

Oracle $\mathtt{RegHonUsr}(\mathtt{ID} \in \mathcal{IDS})$ **/** Only twice in **CKS-l**

if $(corrupt, \mathtt{ID}, \perp, \cdot) \in \mathcal{D}$:
  return $\perp$
$(sk, pk) \xleftarrow{\$} \mathsf{Gen}(\mathtt{ID})$
$\mathcal{D} \cup \{(honest, \mathtt{ID}, sk, pk)\}$
return pk

Oracle $\mathtt{RegCorUsr}(\mathtt{ID} \in \mathcal{IDS}, pk)$ **/** $Q_{\mathtt{RCU}}$ queries

if $(corrupt, \mathtt{ID}, \perp, \cdot) \in \mathcal{D}$:
  $(corrupt, \mathtt{ID}, \perp, \cdot) := (corrupt, \mathtt{ID}, \perp, pk)$
else :
  $\mathcal{D} \cup \{(corrupt, \mathtt{ID}, \perp, pk)\}$

Oracle $\mathtt{RevCorQue}(\mathtt{ID}_1 \in \mathcal{IDS}, \mathtt{ID}_2 \in \mathcal{IDS})$ **/** $Q_{\mathtt{RCQ}}$ queries

if $(honest, \mathtt{ID}_1, \cdot, \cdot) \in \mathcal{D} \wedge (corrupt, \mathtt{ID}_2, \cdot, \cdot) \in \mathcal{D}$:
  return $\mathsf{SdK}(\mathtt{ID}_2, pk_2, \mathtt{ID}_1, sk_1)$
elseif $(corrupt, \mathtt{ID}_1, \cdot, \cdot) \in \mathcal{D} \wedge (honest, \mathtt{ID}_2, \cdot, \cdot) \in \mathcal{D}$:
  return $\mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2)$
return $\perp$

Oracle $\mathtt{TestQue}(\mathtt{ID}_1 \in \mathcal{IDS}, \mathtt{ID}_2 \in \mathcal{IDS})$ **/** Only 1 in **CKS-l**

if $\mathtt{ID}_1 = \mathtt{ID}_2$:
  return $\perp$
if $(honest, \mathtt{ID}_1, \cdot, \cdot) \in \mathcal{D} \wedge (honest, \mathtt{ID}_2, \cdot, \cdot) \in \mathcal{D}$:
  if $b = 0$:
    $k := \mathsf{SdK}(\mathtt{ID}_1, pk_1, \mathtt{ID}_2, sk_2)$
  if $b = 1$:
    if $(\mathtt{ID}_1, \mathtt{ID}_2, k) \in \mathcal{K} \vee (\mathtt{ID}_2, \mathtt{ID}_1, k) \in \mathcal{K}$:
      return k
    $k \xleftarrow{\$} \mathcal{SKS}$
    $\mathcal{K} \cup \{(\mathtt{ID}_1, \mathtt{ID}_2, k)\}$
  return k
return $\perp$

Figure 7: Game defining $\mathbf{CKS}_{\mathsf{NIKE},par}$ (and $\mathbf{CKS\text{-}l}_{\mathsf{NIKE},par}$) for a non-interactive key exchange NIKE with adversary A.

B.1 Non-Interactive Zero-Knowledge Proofs

Zero-Knowledge proofs [GMR85] allow a verifier to convince a prover of the validity of a statement without revealing anything beyond that. In the random oracle model [BR93] zero-knowledge proofs can be made non-interactive [BFM88] by applying the Fiat-Shamir transformation [FS87].

Definition 7 (Zero-Knowledge Proof of Knowledge)**.** A zero-knowledge proof of knowledge ZKPoK for an NP language $\mathcal{L}$ [9] is defined as a tuple $\mathsf{ZKPoK} := (\mathsf{ZK.Prv}, \mathsf{ZK.Ver})$ of the following oracle algorithms.

$\pi \xleftarrow{\$} \mathsf{ZK.Prv}^{\mathsf{H}}(x, w)$**:** Given a statement x and a witness w, the probabilistic prover algorithm ZK.Prv returns a proof π.

$1/0 \leftarrow \mathsf{ZK.Ver}^{\mathsf{H}}(x, \pi)$**:** Given a statement x and a proof π, the deterministic verifier algorithm returns either 1 for accept or 0 for reject.

Similar to the work of [Unr15] we assume a distribution RODist on functions, modelling the distribution of our random oracle. That is, given a random oracle $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$, RODist would be the uniform distribution on $\{0,1\}^* \to \{0,1\}^n$.

ZKPoK Security Notions. Besides *completeness*, which captures that valid proofs are accepted by the verifier, a zero-knowledge proof of knowledge should fulfil two additional properties; *soundness* ensures a cheating prover cannot convince the verifier of a false proof, and *zero-knowledge* conveys that the verifier learns nothing from its interaction with the prover beyond the fact that he knows a valid witness to the proof. We make this more precise with the following definitions and note that we require the strong notion of *simulation soundness with a straight-line extractor* [Unr15], sometimes referred to as "online extractability" in the literature.

Definition 8 (Completeness)**.** *Completeness* for a zero-knowledge proof of knowledge ZKPoK of an NP language $\mathcal{L}$ is defined via the game $\mathbf{CMPLT}_{\mathsf{ZKPoK}}$ depicted in Figure 8. For an adversary A, we define A's advantage in $\mathbf{CMPLT}_{\mathsf{ZKPoK}}$ as

$$\mathsf{Adv}^{\mathbf{CMPLT}}_{\mathsf{ZKPoK}}(\mathsf{A}) := \Pr[\mathbf{CMPLT}^{\mathsf{A}}_{\mathsf{ZKPoK}} \Rightarrow 1],$$

and say that ZKPoK is $(\varepsilon, Q_{\mathsf{H}})$-**CMPLT** if for all quantum-polynomial-time adversaries A, making at most Q_{H} queries (possibly in superposition) to the random oracle H, we have $\mathsf{Adv}^{\mathbf{CMPLT}}_{\mathsf{ZKPoK}}(\mathsf{A}) \leq \varepsilon(Q_{\mathsf{H}})$.

Game $\mathbf{CMPLT}_{\mathsf{ZKPoK}}$

$\mathsf{H} \xleftarrow{\$} \mathsf{RODist}$
$(x, w) \xleftarrow{\$} \mathsf{A}^{|\mathsf{H}\rangle}$
$\pi \xleftarrow{\$} \mathsf{ZK.Prv}^{\mathsf{H}}(x, w)$
return $[\![\mathsf{ZK.Ver}^{\mathsf{H}}(x, \pi) = 0 \wedge R(x, w) = 1]\!]$

Figure 8: Game defining $\mathbf{CMPLT}_{\mathsf{ZKPoK}}$ for a zero-knowledge proof of knowledge ZKPoK with adversary A.

[9] The language $\mathcal{L}$ is defined as the set of all yes-instances of the relation R, i.e. $\mathcal{L} = \{x : \exists\, w \text{ s.t. } R(x, w) = 1\}$.

For the following notions we additionally require a simulator $\mathsf{ZK.Sim} := (\mathsf{ZK.Sim}_1, \mathsf{ZK.Sim}_2)$ that is split into two classical algorithms $\mathsf{ZK.Sim}_1$ and $\mathsf{ZK.Sim}_2$, where:

$\mathsf{H} \xleftarrow{\$} \mathsf{ZK.Sim}_1$**:** The probabilistic simulator algorithm $\mathsf{ZK.Sim}_1$ returns a circuit H which represents the initial simulated random oracle.

$\pi \xleftarrow{\$} \mathsf{ZK.Sim}_2(x)$**:** Given a statement x the stateful simulator algorithm $\mathsf{ZK.Sim}_2$ returns a proof π. Additionally, $\mathsf{ZK.Sim}_2$ is given access to the description of H and may replace it with a different description (i.e. it can program the random oracle).

Definition 9 (Zero-Knowledge [GMR85])**.** *Zero-knowledge* for a zero-knowledge proof of knowledge ZKPoK of an NP language $\mathcal{L}$ is defined via the game $\mathbf{ZK}^b_{\mathsf{ZKPoK}}$, depicted in Figure 9, where $\mathbf{ZK}^b_{\mathsf{ZKPoK}}$ is parametrised by a bit b. For an adversary A, we define A's advantage in $\mathbf{ZK}^b_{\mathsf{ZKPoK}}$ as

$$\mathsf{Adv}^{\mathbf{ZK}}_{\mathsf{ZKPoK}}(\mathsf{A}) := \left|\Pr[\mathbf{ZK}^{0,\mathsf{A}}_{\mathsf{ZKPoK}} \Rightarrow 1] - \Pr[\mathbf{ZK}^{1,\mathsf{A}}_{\mathsf{ZKPoK}} \Rightarrow 1]\right|,$$

and say that ZKPoK is (ϕ, Q_{H})-**ZK**, if there exists a PPT simulator $\mathsf{ZK.Sim} := (\mathsf{ZK.Sim}_1, \mathsf{ZK.Sim}_2)$, such that for all quantum-polynomial-time adversaries A, making at most Q_{H} queries (possibly in superposition) to the random oracle H, we have $\mathsf{Adv}^{\mathbf{ZK}}_{\mathsf{ZKPoK}}(\mathsf{A}) \le \phi(Q_{\mathsf{H}})$.

Game $\mathbf{ZK}^0_{\mathsf{ZKPoK}}$

$\mathsf{H} \xleftarrow{\$} \mathsf{RODist}$
$b' \xleftarrow{\$} \mathsf{A}^{|\mathsf{H}\rangle, \mathsf{ZK.Prv}(\cdot,\cdot)}$
return $[\![b' = 0]\!]$

Game $\mathbf{ZK}^1_{\mathsf{ZKPoK}}$

$\mathsf{H} \xleftarrow{\$} \mathsf{ZK.Sim}_1$
$b' \xleftarrow{\$} \mathsf{A}^{|\mathsf{H}\rangle, \mathsf{ZK.Sim}'_2(\cdot,\cdot)}$
return $[\![b' = 1]\!]$

Procedure $\mathsf{ZK.Sim}'_2(x, w)$

if $R(x,w) = 0$:
  return $\perp$
else :
  return $\mathsf{ZK.Sim}_2(x)$

Figure 9: Games defining $\mathbf{ZK}^b_{\mathsf{ZKPoK}}$ for a zero-knowledge proof of knowledge ZKPoK with adversary A and simulator $\mathsf{ZK.Sim} := (\mathsf{ZK.Sim}_1, \mathsf{ZK.Sim}_2)$. The purpose of $\mathsf{ZK.Sim}'_2(\cdot,\cdot)$ is merely to serve as an interface for the adversary who expects a prover taking two arguments x and w.

Definition 10 (Simulation-Sound Online-Extractability [Unr15])**.** *Simulation-sound online-extractability* [10] for a zero-knowledge proof of knowledge ZKPoK of an NP language $\mathcal{L}$ is defined via the game $\mathbf{SSND}_{\mathsf{ZKPoK}}$, depicted in Figure 10. For an adversary A, we define A's advantage in $\mathbf{SSND}_{\mathsf{ZKPoK}}$ as

$$\mathsf{Adv}^{\mathbf{SSND}}_{\mathsf{ZKPoK}}(\mathsf{A}) := \Pr[\mathbf{SSND}^{\mathsf{A}}_{\mathsf{ZKPoK}} \Rightarrow 1],$$

[10] Online-extractability is sometimes referred to as straight line extractability in the literature.

and say that ZKPoK is (ψ, Q_{H})-**SSND** relative to a simulator $\mathsf{ZK.Sim} := (\mathsf{ZK.Sim}_1, \mathsf{ZK.Sim}_2)$, if there exists a PPT extractor ZK.Ext such that for all quantum-polynomial-time adversaries A, making at most Q_{H} queries to the random oracle H, we have $\mathsf{Adv}^{\mathbf{SSND}}_{\mathsf{ZKPoK}}(\mathsf{A}) \le \psi(Q_{\mathsf{H}})$.

Game $\mathbf{SSND}_{\mathsf{ZKPoK}}$

$\mathsf{H} \xleftarrow{\$} \mathsf{ZK.Sim}_1$
$(x, \pi) \xleftarrow{\$} \mathsf{A}^{|\mathsf{H}\rangle, \mathsf{ZK.Sim}_2(\cdot)}$
$w \xleftarrow{\$} \mathsf{ZK.Ext}(\mathsf{H}, x, \pi)$
return $[\![\mathsf{ZK.Ver}^{\mathsf{H}}(x,\pi) = 1 \wedge R(x,w) = 0 \wedge (x,\pi) \notin \tilde{\boldsymbol{\pi}}]\!]$

Figure 10: Games defining $\mathbf{SSND}_{\mathsf{ZKPoK}}$ for a zero-knowledge proof of knowledge ZKPoK with adversary A, simulator $\mathsf{ZK.Sim} := (\mathsf{ZK.Sim}_1, \mathsf{ZK.Sim}_2)$ and extractor ZK.Ext. Here, $\tilde{\boldsymbol{\pi}}$ denotes the set of all proofs returned by $\mathsf{ZK.Sim}_2(\cdot)$ (together with the corresponding statements).

C Proofs for Section 5 (Construction)

Lemma 4 (Honest Correctness)**.** For all (possibly unbounded) adversaries A the non-interactive key exchange $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$ construction depicted in Figure 3 has *honest correctness error*

$$\delta \le \frac{4\beta^2 d^2 N}{q}$$

as per Definition 3.

Proof. The proof strategy is similar to the proof of Theorem 5, except that we can bound the probability of any coefficient of $\boldsymbol{k}^\star$ being in the interval

$$S^\star = \left[\frac{q}{4} \pm \beta^2 dN\right] \cup \left[\frac{3q}{4} \pm \beta^2 dN\right]$$

by $\frac{4\beta^2 d^2 N}{q}$, with a union bound over all coefficients. ■

Combining Theorem 6 with the passive version of Theorem 3 immediately implies the following Corollary.

Corollary 1. *For any* PPT *adversary* A *against* $\mathsf{NIKE} := (\mathsf{Stp}, \mathsf{Gen}, \mathsf{SdK})$*, depicted in Figure 3, making an arbitrary number of queries (possibly in superposition) to* H*,* Q_{RHU} *(classical) queries to the* `RegHonUsr` *oracle and one (classical) query to the* `TestQue` *oracle, there exist* PPT *adversaries* B_1*,* B_2 *such that*

$$\mathsf{Adv}^{\mathbf{HKR\text{-}CKS}}_{\mathsf{NIKE},par}(\mathsf{A}) \le \frac{Q^2_{RHU} \cdot Q_{TQ}}{2} \cdot \Big(\mathsf{Adv}^{\mathbf{M\text{-}LWE}}_{q,N,N,\chi}(\mathsf{B}_1) + \mathsf{Adv}^{\mathbf{M\text{-}LWE}}_{q,N,N+1,\chi}(\mathsf{B}_2) + \frac{4\beta d}{q}\Big).$$

```
Games G0, G1, G2, G3, G4, G5
Proof. 01 b ←$ {0,1}
02 𝒟 := ⊥
03 𝒦 := ⊥
04 b′ ← A^{|H⟩, RegHonUsr(·), RegCorUsr(·,·), RevCorQue(·,·), TestQue(·,·)}
05 return [[b = b′]]

Oracle RegHonUsr(ID ∈ IDS)                                   // Twice
06 if (corrupt, ID, ⊥, ·) ∈ 𝒟 :
07    return ⊥
08 (sk′_ID, pk′_ID) ←$ Gen(ID)
09 π ←$ ZK.Prv(pk′_ID, sk′_ID)
10 π ←$ ZK.Sim2(pk′_ID)                                      // G5
11 sk_ID := sk′_ID
12 pk_ID := (pk′_ID, π)
13 𝒟 ∪ {(honest, ID, sk_ID, pk_ID)}
14 return pk_ID

Oracle RegCorUsr(ID ∈ IDS, pk)                               // Q_RCU queries
15 parse pk → (pk′_ID, π)
16 ~sk′_ID ←$ ZK.Ext(H, pk′_ID, π)                           // G1
17 if (corrupt, ID, ·, ·) ∈ 𝒟 :
18    (corrupt, ID, ⊥, ·) := (corrupt, ID, ⊥, pk)
19    (corrupt, ID, ·, ·) := (corrupt, ID, ~sk′_ID, pk)      // G1
20 else :
21    𝒟 ∪ {(corrupt, ID, ⊥, pk)}
22    𝒟 ∪ {(corrupt, ID, ~sk′_ID, pk)}                       // G1

Oracle RevCorQue(ID1 ∈ IDS, ID2 ∈ IDS)                       // Q_RCQ queries
23 if (honest, ID1, ·, ·) ∈ 𝒟 ∧ (corrupt, ID2, ·, ·) ∈ 𝒟 :
24    if SdK(ID2, pk2, ID1, sk1) ≠ SdK(ID1, pk1, ID2, sk2) :  // G2-G3
25       abort                                               // G2-G3
26    return SdK(ID2, pk2, ID1, sk1)
27    return SdK(ID1, pk1, ID2, ~sk′2)                       // G3
28 elseif (corrupt, ID1, ·, ·) ∈ 𝒟 ∧ (honest, ID2, ·, ·) ∈ 𝒟 :
29    if SdK(ID1, pk1, ID2, sk2) ≠ SdK(ID2, pk2, ID1, sk1) :  // G2-G3
30       abort                                               // G2-G3
31    return SdK(ID1, pk1, ID2, sk2)
32    return SdK(ID2, pk2, ID1, ~sk′1)                       // G3
33 return ⊥

Oracle TestQue(ID1 ∈ IDS, ID2 ∈ IDS)                         // Once
34 if ID1 = ID2 :
35    return ⊥
36 if (honest, ID1, ·, ·) ∈ 𝒟 ∧ (honest, ID2, ·, ·) ∈ 𝒟 :
37    if b = 0 :
38       k := SdK(ID1, pk1, ID2, sk2)
39    if b = 1 :
40       if (ID1, ID2, k) ∈ 𝒦 ∨ (ID2, ID1, k) ∈ 𝒦 :
41          return k
42       k ←$ SKS
43       𝒦 ∪ {(ID1, ID2, k)}
44    return k
45 return ⊥
```

Figure 11: Games G_1-G_5, for the proof of **CKS-l** of NIKE in Figure 5.

C.1 Proof of Theorem 7

Note that it suffices to prove **CKS-l** and then apply Corollary 1 to obtain the result for **CKS**. Let A be an adversary against NIKE in the **CKS-l** game. Consider the sequence of games in Figure 11, where $Q_{\mathtt{TQ}} = 1$ and $Q_{\mathtt{RHU}} = 2$ denote the number of queries to `TestQue` and `RegHonUsr`, respectively.

Game G_0 This is the original $\textbf{CKS-l}_{\mathsf{NIKE},par}$ game, so by definition

$$\Pr\left[G_0^{A} \Rightarrow 1\right] = \Pr\left[\textbf{CKS-l}^{A}_{\mathsf{NIKE},par} \Rightarrow 1\right].$$

Game G_1 In this game we modify the `RegCorUsr` oracle so that the secret key $\widetilde{sk'_{\mathtt{ID}}}$ is extracted from the proof π of a public key pk on Line 16, and stored with the identity ID. This requires the strong notion of simulation-sound online-extractability, from Definition 10. If the extraction fails, then by default $\widetilde{sk'_{\mathtt{ID}}} = \perp$. I.e. the secret key is not stored, as in the original game. Therefore,

$$\left|\Pr\left[G_0^{A} \Rightarrow 1\right] - \Pr\left[G_1^{A} \Rightarrow 1\right]\right| \le Q_{\mathtt{RCU}} \cdot \mathsf{Adv}^{\textbf{SSND}}_{\mathsf{ZKPoK}}(B_1).$$

Game G_2 In this game we introduce a new condition for aborting: If at any point in the simulation the adversary asks a query the `RevCorQue` oracles on two public keys that cause a key mismatch, then abort the simulation. Note that this condition is efficiently testable, as the game knows all the secret keys. It is important to note that the strong notion of **SM-COR** quantifies over all IDs, meaning we can bound the probability of this event happening with a reduction to the semi-malicious correctness property, Definition 4, of NIKE′ by

$$\left|\Pr\left[G_1^{A} \Rightarrow 1\right] - \Pr\left[G_2^{A} \Rightarrow 1\right]\right| \le \mathsf{Adv}^{\textbf{SM-COR}}_{\mathsf{NIKE}',par'}(B_2).$$

Game G_3 In this game we modify the `RevCorQue` oracle on Line 27 and Line 32 to use the secret key that was extracted when the corresponding public key was registered as a corrupt key. Since this is just a conceptual change and the derived key is always the same for both secret keys, we get

$$\left|\Pr\left[G_2^{A} \Rightarrow 1\right] - \Pr\left[G_3^{A} \Rightarrow 1\right]\right| = 0.$$

Game G_4 In this game we undo the modification from G_2 and get

$$\left|\Pr\left[\mathsf{G}_3^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1\right]\right| \leq \mathsf{Adv}^{\textbf{SM-COR}}_{\mathsf{NIKE}',par'}(\mathsf{B}_2).$$

Game G_5 In this game we modify the `RegHonUsr` oracle and replace the zero-knowledge proof of knowledge on Line 10 with a simulated proof. Recall that we are considering **CKS-l**, meaning only 2 calls are made to `RegHonUsr`. By the zero-knowledge property, Definition 9, we get

$$\left|\Pr\left[\mathsf{G}_4^{\mathsf{A}} \Rightarrow 1\right] - \Pr\left[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1\right]\right| \leq 2 \cdot \mathsf{Adv}^{\textbf{ZK}}_{\mathsf{ZKPoK}}(\mathsf{B}_3).$$

Adversary $\mathsf{C}^{|\mathsf{H}\rangle,\mathtt{RegHonUsr}(\cdot),\mathtt{TestQue}(\cdot,\cdot)}$

$b' \leftarrow \mathsf{A}^{|\mathsf{H}\rangle,\ \mathtt{SimRegHonUsr},\ \mathtt{SimRegCorUsr},\ \mathtt{SimRevCorQue},\ \mathtt{SimTestQue}}$
return b'

Oracle $\mathtt{SimRegCorUsr}(\mathtt{ID} \in \mathcal{IDS}, pk)$ // Same as G_5

Oracle $\mathtt{SimRegHonUsr}(\mathtt{ID} \in \mathcal{IDS})$

$pk' \leftarrow \mathtt{RegHonUsr}(\mathtt{ID})$
$\pi \xleftarrow{\$} \mathsf{ZK.Sim}_2(pk')$
$pk := (pk', \pi)$
return pk

Oracle $\mathtt{SimRevCorQue}(\mathtt{ID}_1, \mathtt{ID}_2)$ // Same as G_5

Oracle $\mathtt{SimTestQue}(\mathtt{ID}_1, \mathtt{ID}_2)$

return $\mathtt{TestQue}(\mathtt{ID}_1, \mathtt{ID}_2)$

Figure 12: Adversary C for bounding G_5.

At this point we no longer need the secret keys to simulate and can show a reduction. Next we make the following Claim.

Claim. *There exists a* PPT *adversary* C *such that*

$$\Pr\left[\mathsf{G}_5^{\mathsf{A}} \Rightarrow 1\right] = \mathsf{Adv}^{\textbf{HKR-CKS-l}}_{\mathsf{NIKE}}(\mathsf{C}).$$

Proof. The adversary C, depicted in Figure 12, against **HKR-CKS-l** is constructed as follows. Any queries that A in G_5 makes to `SimRegHonUsr` and `SimTestQue`, C forwards to the `RegHonUsr` and `TestQue` oracles in the **HKR-CKS-l** game, except that C adds a simulated proof to the public key returned by the `RegHonUsr` oracle. On the other hand, the queries of A to the `RegCorUsr` and `RevCorQue` oracles are simulated by C as in G_5, since no secret key is required to compute their output. The same guess bit b' returned by A is also returned by C. It is easy to see that the reduction perfectly simulates the view of the adversary and therefore any advantage of A directly carries over to the C. ■

Collecting all the probabilities and applying Theorem 3 yields the bounds from the Theorem statement, concluding the proof. ■