Technische Universität München
TUM School of Computation, Information and Technology

Formalization and Verification of Post-Quantum Cryptography

Katharina Beate Heidler

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Helmut Seidl

Prüfende der Dissertation:

1. Prof. Tobias Nipkow, Ph.D.
2. Prof. Dr. Gilles Barthe

Die Dissertation wurde am 06.02.2025 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 24.04.2025 angenommen.

Acknowledgements

First of all, I want to thank Tobias Nipkow for his steady guidance throughout this thesis, for the insightful discussions, for asking the right kind of questions on my projects and changing my perspectives to find the right answers. Thank you also to Manuel Eberl, my mentor, for introducing me to having fun with Isabelle, discussing work and private problems, always hearing out my troubles and giving me the confidence needed.

Special thanks to Manuel Barbosa, Peter Schwabe and Andreas Hülsing, for the inspiring discussions on Kyber and quantum security and giving me the opportunity to come to the Formosa retreat. A huge thank you to Dominique Unruh for never tiring to explain quantum computing concepts to me and for making our meetings fun and inspiring. Thanks to the ConVeY research training group for the financial support. Especially with the pandemic closing all kinds of connections between PhD students, the ConVeY retreats were a great way to connect with peers, get feedback and exchange ideas.

Finally, I want to thank all my family and friends who supported me throughout my thesis. The biggest thank you goes to my loving husband for always being by my side.

1 Abstract

With the advancements on quantum computers, the threat on classical cryptography becomes more and more imminent. To counter this quantum threat, post-quantum cryptography is being developed. Formalizations of the specifications, proofs and implementations help to guarantee security and correctness of these new cryptosystems. Sometimes, the formalization also reveals errors and gaps in the scheme or proofs that could potentially be used for concrete attacks.

This thesis highlights several aspects of post-quantum cryptography where formalization can be useful. These aspects include hardness assumptions, the verification of concrete crypto systems, their correctness as well as classical and quantum security proofs. For every aspect mentioned, a major contribution is described, demonstrating the possibilities of formalization in post-quantum cryptography with the theorem prover Isabelle. Since the formalization of post-quantum cryptography is a very new field, my formalizations often are the first in their respective areas.

The projects for this thesis are: 1) formalizing the Shortest and Closest Vector Problems for algebraic lattices and their NP-hardness reduction proofs, 2) formalizing the post-quantum public key encryption scheme Kyber, its correctness and classical security and 3) formalizing the One-way to Hiding Theorem for security proofs against quantum adversaries. For the first project, we point out several inaccuracies and problems in the proofs and give verified alternatives. In the second project, we show that the correctness error bound was invalid, giving counter-examples in small parameter sets. We verify the proofs for an alternative bound, showing that correctness is indeed fulfilled. The classical security proof was formalized without problems. The third project takes a step towards foundationally formalizing quantum security proofs. We verify and extend the theorem to possibly infinite dimensional Hilbert spaces, non-terminating adversaries and give an alternative proof omitting the notions of Bures-distance and fidelity.

2 Zusammenfassung

Durch die Fortschritte der Quantencomputer sieht sich die klassische Kryptographie immer mehr bedroht. Um dagegen vorzugehen, wird die Post-Quantum Kryptographie entwickelt. Die Formalisierung der Spezifikationen, Beweise und Implementierungen hilft die Sicherheit und Korrektheit dieser neuen Kryptosysteme zu garantieren. Manchmal zeigen die Formalisierungen auch Fehler und Lücken in den Schemata und Beweisen auf, die potenziell für konkrete Angriffe genutzt werden können.

Diese Dissertation zeigt verschiedene Aspekte der Post-Quantum Kryptographie auf, bei denen Formalisierungen hilfreich sind. Diese Aspekte beinhalten die Annahmen zu NP-schweren Problemen, die Verifikation von konkreten Kryptosystemen, ihrer Korrektheit und der klassischen Sicherheitsbeweisen sowie Sicherheitsbeweisen bezüglich Gegner mit Zugriff auf Quantencomputern. Für jeden der genannten Aspekte wird ein wichtiger Beitrag zur Wissenschaft beschrieben und die Möglichkeiten der Formalisierung von Post-Quantum Kryptographie mit dem Theorembeweiser Isabelle aufgezeigt. Meine Formalisierungen sind oft die ersten in den jeweiligen Bereichen, da die Formalisierung von Post-Quantum Kryptographie ein sehr neues Gebiet ist.

Die Projekte dieser Dissertation sind: 1) die Formalisierung des Kürzesten und Nähesten Vektor Problems in algebraischen Gittern und deren Reduktionsbeweise bezüglich ihrer NP-Schwierigkeit 2) die Formalisierung des Post-Quantum Public-Key Verschlüsselungssystems Kyber, dessen Korrektheit und der klassischen Sicherheitsbeweise und 3) die Formalisierung des One-way to Hiding Theorems, das in Sicherheitsbeweisen mit Quanten-Gegner verwendet wird. Für das erste Projekt weisen wir auf einige Ungenauigkeiten und Probleme in den Beweisen hin und beweisen verifizierte Alternativen. Im zweiten Projekt zeigen wir, dass die Schranke für den Korrektheitsfehler ungültig ist und geben Gegenbeispiele mit kleinen Parametern an. Außerdem verifizieren wir die Beweise für eine alternative Schranke und zeigen, dass die Korrektheit trotzdem noch erfüllt ist. Der klassische Sicherheitsbeweis wurde ohne Probleme formalisiert. Das dritte Projekt macht einen Schritt in die Richtung von Grund auf formalisierter Sicherheitsbeweise gegen Quantencomputer. Wir verifizieren und erweitern das Theorem zu möglicherweise unendlich-dimensionalen Hilberträumen, nicht terminierenden Gegner und zeigen einen alternativen Beweis, der die Konzepte der Bures-Distanz und Fidelität nicht benutzt.

Contents

Index of Publications

All listed publications are single first author publications.

Core Publications

1) Katharina Kreuzer and Tobias Nipkow. *Verification of NP-Hardness Reduction Functions for Exact Lattice Problems.* In Automated Deduction – CADE 29, page 365–381. Springer Nature Switzerland, 2023. DOI: `10.1007/978-3-031-38499-8_21`. *Core factor 1.*

3) Katharina Kreuzer. *Verification of Correctness and Security Properties for CRYSTALS-KYBER.* In 2024 IEEE 37th Computer Security Foundations Symposium (CSF), volume 2283 of LNCS, page 511–526. IEEE, July 2024. DOI: `10.1109/csf61375.2024.00016`. *Core factor 1.*

6) Katharina Heidler and Dominique Unruh. *Formalizing the One-way to Hiding Theorem.* In Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2025, page 243–256, New York, NY, USA, 2025. ACM. DOI: `10.1145/3703595.3705887`. *Core factor 1.*

Acronyms

AFP	Archive of Formal Proofs.
BHLE	Bounded Homogeneous Linear Equations problem.
CVP	Closest Vector Problem.
FO	Fujisaki-Okamoto transform.
IND-CPA	INDinduishability under Chosen Plaintext Attack.
KEM	Key Encapsulation Module.
LWE	Learning With Errors problem.
ML-KEM	Module-Lattice-Based Key-Encapsulation Mechanism.
mLWE	module Learning With Errors problem.
NIST	National Institute of Standards and Technologies (US).
NTT	Number Theoretic Transform.
O2H	One-way to Hiding Theorem.
PKE	Public Key Encryption.
PQC	Post-Quantum Cryptography.
PRF	Pseudo-Random Function Family.
QROM	Quantum Random Oracle Model.
ROM	Random Oracle Model.
RSA	Rivest-Shamir-Adleman crypto system.
SIS	Shortest Integer Solution problem.
SIVP	Shortest Independent Vector Problem.
SVP	Shortest Vector Problem.

3 Introduction

Already since ancient times, cryptography has enabled people to communicate secret information: Caesar noting troop movements, Queen Elizabeth I. deciphering documents which led to Mary Queen of Scots' execution, the Enigma machine used by the Germans in World War II, just to name some prominent examples. In all of the examples above, the cryptographers and cryptanalysts battled for the secret information. Especially in modern times, with computers at the hand and cryptography being used in everyday life, we want to make sure that the cryptography protecting our sensitive data and money stays safe to use and secure against attacks.

A modern approach to tighten the security of crypto schemes is the formalization in an automated tool. This has already been proposed by Halevi in 2005 [52] and many researchers have since worked on this idea. Using formalization, we can find errors in the systems, protocols and proofs. Sometimes these errors can even be exploited to discover new attacks. For example, Albrecht et al. [79, 6] found active attacks against the Jitsi video platform and the Matrix chat protocol during formalization. Both software are widely used by companies and individuals all around the world. Another example is the Ethereum blockchain which has started an effort to formally verify their smart contracts[1]. The cryptocurrency ether managed by Ethereum has the second largest market capitalization after bitcoin. Small bugs in the code of Ethereum can lead to huge monetary losses.

In the past decades, a new threat to cryptography has risen: with large-scale quantum computers all widely-used cryptosystems based on RSA or Diffie-Hellman will be broken by Shor's algorithm [104]. Recent advances on quantum computers point towards the possibility of these large-scale constructions. Already today, a possible attack on sensitive data is to store it now for decryption once quantum computers become powerful enough. This is called the harvest-now-decrypt-later attack[2].

To counter the quantum threat, a new cryptographic field is evolving, namely post-quantum cryptography (PQC). PQC describes cryptography that runs on classical computers but is secure against attacks from both classical and quantum machines. The users of PQC do not need any access to quantum computers. Therefore, these PQC systems can already be used now to defend against the harvest-now-decrypt-later attack.

However, developing new cryptographic principles may take time and a lot of trial-and-error. New systems need to be developed basing on different hard underlying problems, security proofs have to be checked, and new attacks are being found. Formalization can be a handy tool to find bugs and errors early on, especially when the formalization

[1] see `https://ethereum.org/en/developers/docs/smart-contracts/verifying/`

[2] see `https://en.wikipedia.org/wiki/Harvest_now,_decrypt_later`

and the implementation are developed simultaneously. Therefore, formalizing PQC is especially interesting and the research area is gaining momentum more and more.

To find good candidates for PQC, the National Institute of Standards and Technology (NIST) of the US is running a standardization process for PQC since 2017. In August 2024, after three rounds of evaluations, NIST finally published a standard for encryption: The Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) standard is based on the submitted crypto scheme Kyber [95].

The general idea behind Kyber is quite simple: solving a system of linear equations over $\mathbb{Z}_p$ (for p prime) with an additional error term is known to be hard on average (under certain constraints). This is called the Learning With Errors (LWE) problem. Regev proved that the LWE problem is hard to solve even with the help of quantum computers [97, 98]. This leads us to our first application of formalization: can we formally verify these hardness results? Many hardness proofs of problems similar to the LWE were discovered in the nineties and sometimes lack a rigorous formulation or formalization. Still, they are the most basic building blocks for PQC.

The algorithms for the Kyber public key encryption (PKE) schemes base on the LWE. A PKE consists of three algorithms: the key generation, encryption and decryption. For the key generation of Kyber, the public key constitutes a LWE instance. The encryption is an extended LWE instance with the message being part of the error term. However, since both key generation and encryption use (random) error terms, it may be the case that decryption fails if the errors get too large. Therefore, the Kyber PKE is only correct up to a correctness error δ. For Kyber to be useful in practice, we must show that the correctness error is indeed small enough to be negligible. Indeed, with careful formalization, I could show in this thesis that the original correctness error estimation was faulty. This may have an impact on the security parameters of Kyber. Fortunately, Barbosa et al. [10] could show a different, but larger bound that still suffices the security requirements for Kyber.

Another important aspect of new crypto systems is showing security properties. The common approach are game-based proofs where we formally model games against a (potentially malicious) adversary and show that the adversary cannot gain information. In the case of the Kyber PKE for example, an important security property is not being able to distinguish a ciphertext from a random instance, even when the plaintext was chosen by the adversary. This is called the indistinguishability under chosen plaintext attack (IND-CPA). The first version of Kyber contained a compression of the encryption that led to an error in the security proof. Such examples show that it is essential to formalize and thoroughly check security proofs as well. Most often, flaws in security proofs can directly be used to implement attacks.

Proving and verifying security properties becomes even harder when handling quantum attackers, that is an attacker with access to quantum computers. An interesting example is the modelling of pseudo-random functions (PRFs). Classically, we use the random oracle model to replace PRFs by lazily sampled truly random functions (called an oracle). Using the classical lazy sampling, we can argue that we cannot distinguish new queries to the oracle function from values that were already queried. This implies that we can easily reprogram the oracle for unqueried values. Quantumly, we cannot reason this way. The problem is that the adversary may query all values in superposition in just one query. It is not straightforward that we can "reprogram" any oracle value at all. The One-way to Hiding (O2H) Theorem considers the "reprogramming" of an oracle in the quantum setting. As such mathematical tools for handling quantum adversaries are quite recent and are being refined more and more, only very few are already formalized foundationally.

When I started this thesis, formalization of PQC was still in its early stages. During the realization of this thesis' projects, other researchers have developed formalizations for PQC as well. Most notably, the development of tools handling quantum attackers and PQC has increased. Since many of these tools are quite new, they often lack the foundational formalizations. With this thesis, I take a first step towards foundational verification of PQC in Isabelle, the theorem prover of my choice. We will justify choosing Isabelle (Section 4.3) and compare Isabelle to other theorem provers in our use-case. In the next section, we will develop the research questions posed in this thesis.

3.1 Research Questions

We outline some important research questions in this field. For example when analysing a crypto system such as Kyber, we can ask several questions. First of all, we will analyse the crypto system and its functionality: Is the proposed crypto system correct, i.e. does the decryption of the encryption always return the original message for a PKE? Next, we need to analyse its security: What security properties does the crypto system guarantee? Going to the foundations of the security proofs, we may ask: What underlying hardness assumptions do the security proofs require? Or considering the difference between classical and quantum adversaries: Do the security properties hold for both classical and quantum adversaries? Lastly, once we verified the specifications, we may ask: Is the crypto system correctly implemented or can we obtain a verified implementation e.g. by code generation?

Often, the answers to the questions above are given by (sometimes long and hard) cryptographic proofs. However, as we humans tend to make errors along the way, errors in cryptographic proofs may lead to attacks on or breaking of said systems. Even in the first Kyber submission, a small error in the security proof led to a significant security issue and a change in the system itself [28]. To minimize the risk of errors in proofs, we formalize the results in theorem provers. Especially when the theorem provers can also verify the implementation or generate verified code, this is of great interest for high security applications. Formalizing and verifying PQC in the interactive theorem prover Isabelle is the main topic of this thesis.

However, as the research questions asked at the beginning of this section are very general, formalizing all findings is out of reach for just one thesis. Instead, this thesis contributes steps to the formalization of the following, more concrete questions:

Research Questions

1. Can we formally verify a hardness property assumed in cryptography, to strengthen the foundation of the crypto systems?
2. Can we formally verify the correctness of PQC systems? For example, can we verify the correctness of Kyber?
3. Can we formally verify security properties against classical and quantum adversaries? Or can we develop formal foundations for tools used to verify security proofs?

The focus of this thesis lies on the formalization and verification of specifications and mathematical tools for post-quantum cryptography. Using these verified specifications, the next step for follow-up research is the generation of verified code or the verification of existing code with respect to the formalized specification.

3.2 Contributions and Outline

With this thesis, I take a first step in foundationally formalizing and verifying post-quantum cryptography in Isabelle. First of all, I give a short overview on PQC in Chapter 4. We discuss the standardization process by NIST, different approaches to PQC (Section 4.1) and lattice-based PQC (Section 4.2) in more detail. We also justify choosing Isabelle for our formalizations in comparison with other theorem provers and give an overview on the state of the art of formalizations in cryptography, especially in PQC (Section 4.3). My formalizations were all implemented in Isabelle.

Motivated by the research questions discussed in the previous section, my contributions take steps in three different directions: hardness reduction proofs, correctness of crypto systems, and security proofs against classical and quantum attackers.

In Chapter 5, we introduce basic hardness assumptions for lattice-based PQC. Our main focus lies on the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP) (Section 5.1), which were the first lattice problems used in cryptography. My contribution of formalizing the CVP and SVP hardness reductions contrive the first step towards a formalization of hardness reductions for lattice-based PQC (Section 5.2). During the formalization, I uncovered imprecisions and flaws in the literature and found alternative proofs filling all gaps. This formalization was – to my knowledge – the first formal verification of any lattice problem hardness reduction so far. At the end of the chapter, we also introduce a “newer” problem: the Learning With Errors (LWE) problem (Section 5.3) is the hardness assumption underlying the Kyber PKE.

In Chapter 6, we discuss an example of lattice-based PQC, namely the Kyber PKE. We define the algorithms of the Kyber PKE (Section 6.1), the notion of δ-correctness (Section 6.2) and the IND-CPA security notion (Section 6.3). My contribution is a formalization of the Kyber PKE, its δ-correctness and classical IND-CPA security proof (Section 6.4). A notable result of my formalization efforts is finding an essential flaw in the correctness error bound. The result was also acknowledged by authors of Kyber. Since the original error bound was faulty, I formalized an alternative correctness error bound showing that the correctness is still valid, but with a different bound. My formalization was – to my knowledge – also the first publicly available and published formalization of the Kyber PKE.

In Chapter 7, we address the security against quantum attackers. After introducing the basics of quantum computing (Section 7.1) and the quantum adversarial model (Section 7.2), we state the O2H Theorem (Section 7.3). Formalizing the proof of the O2H Theorem (Section 7.4) is my contribution in this chapter. This formalization is – to my knowledge – the first foundational verification of the O2H Theorem. We also give an alternative (and for the formalization simpler) proof to the existing literature and extend the result to infinite dimensions and possibly non-terminating adversaries.

In the end of the thesis, we will shortly summarize the findings in a conclusion (Chapter 8). We also give several ideas for future work following the three main topics: hardness assumptions, Kyber and quantum adversaries.

This thesis is based on the following core publications:

1) *Verification of NP-Hardness Reduction Functions for Exact Lattice Problems* [67] (joint work with Tobias Nipkow, published at CADE29, received the "Best Student Paper Award") — Appendix A

2) *Verification of Correctness and Security Properties for CRYSTALS-KYBER* [66] (published at CSF24) — Appendix B

3) *Formalizing the One-way to Hiding Theorem* [55] (joint work with Dominique Unruh, published at CPP25) — Appendix C

The following publications are not part of this thesis, but may be of interest to the reader:

- *Verification of the (1–δ)-Correctness Proof of CRYSTALS-KYBER with Number Theoretic Transform* [65] (preprint, presented at FAVPQC 2023)

- Isabelle formalization: *Hardness of Lattice Problems* [63] (formalization artefact of paper 1, published at AFP)

- Isabelle formalization: *CRYSTALS-Kyber* [61] (formalization artefact of paper 2, published at AFP)

- Isabelle formalization: *CRYSTALS-Kyber Security* [62] (formalization artefact of paper 2, published at AFP)

- Isabelle formalization: *O2H: A formalization of the one-way-to-hiding lemma in Isabelle* [54] (formalization artefact of paper 3, public repository, to be published at AFP)

4 Overview on Post-Quantum Cryptography

In the last decades, quantum computers have made extensive progress. In 2019, Google first claimed quantum supremacy with their Sycamore quantum computer [39]. Quantum supremacy means achieving a quantum computer that can perform some calculation faster than any classical computer. After that, in 2020, a group of the University of Science and Technology of China followed in building Jiuzhang, a quantum computer also achieving quantum supremacy but using different technology [16]. The main issue with Sycamore and Jiuzhang is the relatively high error rate on the qubits. In December 2024, Google Quantum AI announced their new quantum computer Willow [86]. Willow successfully uses quantum error correction and generates more reliable outcomes whilst increasing the number of qubits [1]. This is a big step towards useful, large-scale quantum computers.

This major breakthrough of achieving quantum supremacy and quantum error correction heated up the discussion on quantum computers and their implications. A major problem of reasonably large quantum computers is that they could factor large prime numbers and solve the discrete logarithm problem in polynomial (even cubic) time using Shor's algorithm [104]. This algorithm was already known since 1994, but gains more relevance with quantum computers coming into reach. A major application of Shor's algorithm on scalable quantum computers is breaking the RSA and Diffie-Hellman crypto systems. Since RSA and Diffie-Hellman are at the core of most widely used cryptosystems, this poses a huge threat on our current cryptography.

To counter this quantum threat, the National Institute of Standards and Technology (NIST) of the United States [93] has started a standardization process for post-quantum cryptography (PQC). PQC denotes cryptographic algorithms running on classical machines that is safe with respect to attacks from both classical and quantum computers. In this context, classical attackers (or classical adversaries) can perform any polynomial-time algorithm using classical computers. Quantum attackers (or quantum adversaries) can perform any polynomial-time algorithm using both classical and quantum computers.

The research on PQC can be divided into two major categories: key encapsulation modules (KEMs) for encryption and signature schemes. KEMs are crypto schemes that use public key encryption (PKE) schemes to securely transmit a symmetric key for further communication using symmetric cryptography. Signature schemes are used for digital signatures and verified authentication. Our focus for this thesis will lie on encryption, especially PKEs.

4.1 Approaches to Post-Quantum Cryptography

After many years and rounds of the standardization process, in July 2022, NIST announced that they have selected an algorithm for KEMs (as well as several signature schemes). However, the search for good alternatives is still ongoing. This is important since most PQC proposals are quite new and therefore did not endure the test of time as long as well-established cryptography. NIST evaluated the new crypto systems during several stages trying to find the most practical and secure schemes.

As PQC asks for novel approaches to cryptography, there was a variety of submissions with different underlying concepts. The main ideas to make schemes quantumly secure can roughly be ordered in five categories:

- **Lattice-based cryptography:** Lattices are the integer span of basis vectors in n-dimensional Euclidean space and form the foundation of this category. Using hard problems on lattices, messages or signatures are obscured. Since lattices yield a multitude of different hard problems as a basis for cryptosystems, the most submissions were of this category. Examples for crypto systems are Kyber [102], Dilithium [101], NTRU [100], and many others.

- **Multivariate cryptography:** The difficulty of solving systems of multivariate equations is exploited for cryptographic functions in this category. As many multivariate schemes yield relatively short signatures, these schemes are especially useful to build signatures. Examples for crypto systems include the Rainbow Signature Scheme [33].

- **Hash-based cryptography:** The hardness of these schemes stems from the hardness to invert hash functions. Again, most schemes in this category are signature schemes. However, hash-based cryptography can also be used for zero-knowledge proofs or other proof based protocols. Examples for crypto systems include the Merkle signature schemes [81] and SPHINCS$^+$ [103].

- **Code-based cryptography:** These schemes rely on error-correcting codes, an idea taken from communication and information theory. The idea is to send redundant information in the encryption, so when some information gets lost over a noisy communication channel, we can reconstruct these errors using error-correction. The Goppa code is an example of error-correcting codes used in cryptography. This category yields both signature and encryption schemes. Examples for crypto systems include McEliece [80].

- **Isogeny-based cryptography:** Isogeny graphs of elliptic curves over finite fields may yield desired properties for crypto systems. Unfortunately, a prominent example, called SIKE [38], was broken in 2022 as a new connection with a different mathematical field was established. There are other crypto systems based on isogenies that are still unbroken, for example C-SIDH [32].

In August 2024, NIST published a standard for encryption: The Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) standard is based on Kyber [95]. For signatures, NIST's standard Module-Lattice-Based Digital Signature Algorithm is based on Dilithium, a signature scheme similar to Kyber [94]. Both standards belong to lattice-based cryptography.

Since most submissions to the standardization process and the new standards themselves are lattice-based, we lay special interest on this category in this thesis. In the next section, we will explain the general ideas of lattice-based cryptography in more detail.

4.2 Lattice-based Post-Quantum Cryptography

Lattices are algebraic structures that yield interesting and hard problems. The most basic problems concerning lattices are the Shortest Vector Problem (SVP — find the shortest vector in a lattice) and the Closest Vector Problem (CVP — find a closest lattice point to a target). The NP-hardness for these two problems can be shown in the worst-case (see [82] for an overview on the complexity of lattice problems). However, knowing that the worst-case problem instance is hard is not enough for cryptography.

Ajtai [2] took the first step to making lattices useful for cryptography by considering the average-case hardness of the SVP and CVP. Average-case hardness means that any instance of these problems is hard with very high probability. More research followed, finding new interesting problems or showing their average-case hardness reductions. To name some: Dwork and Ajtai showed the average-case hardness of the Shortest Integer Solution (SIS — find a short integer vector that solves a system of linear equations) problem and built a crypto system on that [4], and Regev introduced the Learning With Errors (LWE) problem and its reduction proofs [97]. We will look more closely at lattice problems in Chapter 5.

First, let us define lattices. This concept of linear algebra underlies all ideas of lattice-based cryptography.

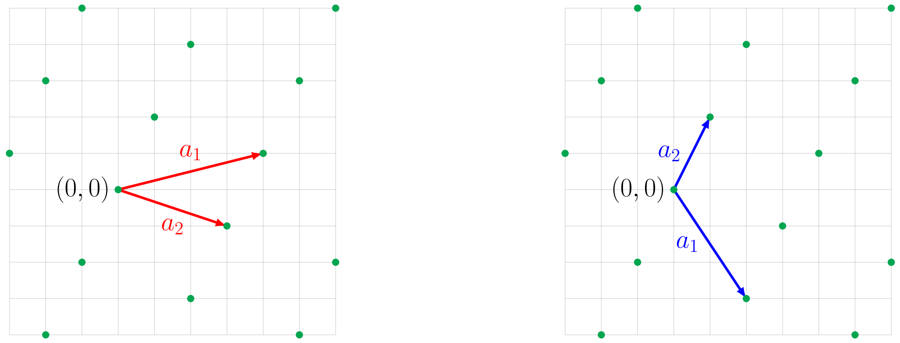
Definition 1 (Lattice)**.** Let $a_1, \dots, a_n \in \mathbb{R}^n$ be a set of linearly independent vectors. The integer span of $a_1, \dots, a_n$ forms a **lattice** $\mathcal{L}$, i.e.

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} c_i a_i \quad \text{where } c_i \in \mathbb{Z} \right\}$$

Lattices form a set of points, that form a group with respect to addition. The group structure is directly inherited from the integer span. The origin is the neutral element of the group and every addition of two lattice points is again in the lattice.

Note that different choices of basis vectors can result in the same lattice. We look at an example of a lattice in the Euclidean plane.

Example 4.2.1. Consider the lattice $\mathcal{L}$ depicted in Figure 4.1. The lattice on the left and right pictures is the same, even though it is generated by different basis vectors. In Figure 4.1a, the generating basis is depicted as red arrows, in the Figure 4.1b as blue arrows.



(a) Lattice $\mathcal{L}$ with red basis vectors **(b)** Lattice $\mathcal{L}$ with blue basis vectors

Figure 4.1: A lattice can have different basis vectors.

In Isabelle, we formalized lattices as a set of integer vectors [63]. The property `is_lattice` checks if a set of integer vectors is generated by an integer matrix with independent columns. We also implemented the function `gen_lattice` generating lattices from an arbitrary matrix. In the latter case, the matrix might not consist of independent columns, so we have to handle bases of the generated lattices more carefully.

Now that we have introduced lattices, we look at the cryptography we can do with lattice problems. We give a short overview on the most important lattice-based crypto systems.

One of the earliest cryptosystem based on lattices is the **Goldreich-Goldwasser-Halevi encryption**, developed in 1997 [49]. Even though it was broken shortly after [87], the idea of using the SVP on lattices started more research in this area.

Another cryptosystem that started in the very beginning of lattice-based crypto was **NTRU** developed by Hoffstein, Pipher and Silverman [56]. It is also based on an interpretation of the SVP on a truncated polynomial ring. There have been many developments of NTRU over the last decades. For example NTRU Prime and NTRUEncrypt were both candidates for the NIST standardization process, NTRUEncrypt even making it to the third round finalists [100, 5]. The NTRUSign algorithms use the NTRU ideas for signatures.

More recent lattice-based PQC include algorithms like FrodoKEM, NewHope, Saber and CRYSTALS-Kyber (which we will abbreviate as Kyber); all of them candidates for standardization of encryption schemes. For signing, well-known lattice-based algorithms include Falcon and CRYSTALS-Dilithium. Let us inspect these systems more closely.

Most of the newer systems also rely on “newer” lattice problems. We will focus on two such lattice problems: the Learning With Errors (LWE) and Learning With Rounding (LWR) problems. Both problems ask to find an integer solution to a perturbed system of linear equations. In the case of the LWE problem, the system is perturbed by adding small but random errors, whereas the LWR problem adds disturbances by rounding. Furthermore, there are extensions of both problems to rings and modules.

FrodoKEM [43] is a cryptosystem based on the LWE problem on algebraically unstructured lattices. It was discarded from the NIST standardization process after the third round due to low performance, but is still recommended by German and Dutch authorities.

Using a variation of LWE over a polynomial ring, **NewHope** [99, 8] made it to round two of NIST's standardization process. The system tries to take advantage of the extra structure of the polynomial ring; however, it was not clear if this additional structure did not lead to more possible attacks.

The **CRYSTALS-Kyber** [28, 102] suite (short Kyber) adds another layer: working with the LWE on modules, Kyber combines the advantages of polynomial rings and vectorization. The result is an optimized and refined cryptosystem that was selected for standardization by NIST after the third round [30, 95]. We will inspect Kyber in more detail in Chapter 6.

In contrast to the systems above, **Saber** [36, 68] uses the LWR problem on modules. Its general construction is very similar to that of Kyber. Saber was a NIST third round finalist and is now under consideration as an alternative.

Falcon is an acronym for Fast Fourier lattice-based compact signatures over NTRU. As its name suggests, it is a signature scheme based on a trapdoor construction using hash-and-sign techniques over special NTRU lattices. The SIS is the hardness assumption for this signature scheme. Falcon was chosen for standardization by NIST.[42]

The signature scheme **CRYSTALS-Dilithium** [101] also belongs to the CRYSTALS suite and is therefore very similar to the ideas of Kyber. Using the "Fiat-Shamir with Aborts" technique by Lyubashevsky, Dilithium is also based on the LWE problem. As it is very efficient and has very small signatures, it was also chosen for standardization after round three of NISTs process.

Finally, we still have to think about the **Q**uantum part in lattice-based P**Q**C: After having developed all these new crypto systems basing on new hard problems, how can we prove that they are secure also against quantum computers? We will discuss ideas on this question in Chapter 7.

4.3 Formalizations in Post-Quantum Cryptography

Formalizing cryptographic proofs has become more and more relevant to guarantee security and generate verified implementations. Still, it is quite a new field with several new specialized theorem provers emerging and well-established provers developing relevant theories.

Use cases for the application of theorem provers are wide: Firstly, one can check and formalize the system specifications and verify cryptographic proofs for protocols, KEMs, PKEs, signature schemes and more. Secondly, one can verify foundational knowledge for cryptography, such as hardness assumptions, background theory, adversarial models, etc. Thirdly, implementations can be verified against their specifications or even generated automatically from verified specifications.

In this thesis, we will mainly look at the first two points. Obtaining verified implementations was out of reach in this project. In this dissertation project, the formalizations were carried out in the interactive theorem prover **Isabelle/HOL** [90, 89, 85]. Why did I choose Isabelle? Firstly, Isabelle has a powerful term-rewriting engine and good automation (e.g. the simplifier and the auto command). The sledgehammer proof search also greatly helps in finding proofs. Secondly, the locale [17, 60] and type class [51] environments make it easy to group parameters and properties. They are also easy to instantiate in various contexts (e.g. different security levels of a crypto systems). Thirdly, there already exist extensive and foundational libraries essential for cryptographic proofs in the distribution and the archive of formal proofs (AFP)[40], e.g. a huge library on probability theory, analysis and algebra, as well as a more concrete library for cryptographic proofs called CryptHOL [73].

The **CryptHOL** library by Lochbihler [73, 74] offers a wide range of cryptographic definitions and tools for game-based security proofs. However, these security proofs only cover the case against classical adversaries. To bridge the gap towards quantum adversaries, the **qrhl-tool** by Unruh [106, 109] based on Isabelle is a novel tool for security proofs in the quantum setting. Unfortunately, the qrhl-tool is not yet foundational. This thesis also provides a step in making the qrhl-tool foundational by formalizing the One-way to Hiding Theorem (see Chapter 7).

Of course, there are also other theorem provers used for formalization of cryptography. In the following, we give a short overview of notable theorem provers dealing with (post-quantum) crypto.

Most prominent is the **EasyCrypt** [20, 48] prover which gives a wide toolset for defining crypto systems and proving security properties. Especially the connection with the compiler language **Jasmin** [70] yields a continuous verification chain from implementation to security proofs. However, a big drawback is that many foundational results on algebra, analysis and probability theory are assumed as black boxes. As EasyCrypt is a relatively new theorem prover, this is not surprising since the workload to build everything foundationally is just too high. In that respect, Isabelle is a good alternative since it provides much more background theory and all its results are indeed foundational.

Another tool for automatic reasoning on cryptography is **CryptoVerif** [23, 24]. Its focus lies on the security proof of protocols, but can also handle symmetric and asymmetric encryptions as well as message authentication codes, signatures and hash functions. It is developed at INRIA and is computationally sound. Recently, CryptoVerif was extended to include some reasoning about quantum adversaries as well [25]. Still, CryptoVerif does not allow the expressivity that we have in Isabelle and their automated provers tend to be harder to handle in hard cases.

The theorem prover **Lean** also has its own crypto libraries called cryptolib [77] and lean-crypto [84]. However, both libraries are just private developments and not yet in the official lean library. This makes cryptographic formalizations in lean hard to work with and maintain.

For the theorem prover **Coq**, the Foundational Cryptographic Framework by Pechter [96] also allows a foundational formalization of cryptographic proofs. Since I already knew Isabelle, but not Coq, I decided to work in Isabelle.

For the rest of the section, we will summarize the state-of-the-art concerning the three areas of research questions we considered in the introduction: the formalization of hardness assumptions, of post-quantum crypto systems (i.e. Kyber) and of the security against quantum adversaries.

Formalization of hardness assumptions. In Isabelle, there is an ongoing effort to show NP-hardness of several problems. Starting with a full formalization of the Cook-Levin Theorem [15], Karp's list of NP-hard problems [46] is being formalized [41]. Gäher and Kunze formalized the Cook-Levin Theorem in Coq [45].

The Karp problems are a basis for hardness reductions for cryptographic hardness assumptions in PQC. For example, the CVP is reduced to the Subset Sum problem, one of Karp's 21 NP-hard problems. When starting this thesis, there were no hardness reductions for hardness assumptions used in lattice-based PQC in Isabelle. Therefore, a formalization of these hardness problems for lattice-based PQC had to start from the very beginning.

Related to the hardness reductions for lattice problems are algorithms for finding a good basis for a lattice. For example, the Lenstra-Lenstra-Lovász algorithm [71] is a well-known technique for the reduction of a lattice basis. The Lenstra-Lenstra-Lovász algorithm was also formalized in Isabelle by Bottesch et al. [29].

Formalization of post-quantum crypto systems. The basis for formalizations of crypto systems in Isabelle is the CryptHOL library by Lochbihler et al. [73, 74, 21]. Using this library, a number of cryptographic schemes have been formalized in Isabelle: the ElGamal encryption system [76], the one-time pad [75], sigma protocols [31] and commitment schemes [31]. Other Isabelle formalizations of cryptography (independent of CryptHOL) include basic formalisms for RSA [72] and some classical cryptographic standards [112]. When starting this thesis, no formalizations of PQC algorithms were available in Isabelle.

During the course of this thesis, several PQC algorithms have been formalized in other theorem provers. For example, Barbosa et al. [10] have published a formalization of Kyber in EasyCrypt and Jasmin shortly after my paper [66] appeared. Other formalizations include a formalization of Saber in EasyCrypt by Hülsing et al. [59] and SPHINCS$^+$ by Barbosa et al. [19] also in EasyCrypt.

To my knowledge, CryptoVerif, Lean or Coq do not have any formalizations of post-quantum cryptography yet.

Formalization of security against quantum adversaries. Up to now and to my knowledge, there is no full and foundational proof of any security property of a PQC system against quantum adversaries. However, tools like qrhl-tool [106, 109] and EasyPQC [18] give possibilities to (partially) formalize some security results against quantum adversaries. Unfortunately, a lot of foundational groundwork still needs to be formalized for these tools to be applicable to concrete crypto schemes. Many mathematical results, like the O2H Theorem, are still black-box assumptions up to now. To my knowledge, there are no formalization of quantum security properties of concrete PQC schemes in qrhl-tool or EasyCrypt so far.

An important method for extending PKEs to KEMs is the Fujisaki-Okamoto (FO) transform [44]. The FO transform was machine-checked in the qrhl-tool by Unruh [107].

For example, Kyber uses the FO transform to generate its KEM from the Kyber PKE. However, a full formalization of the Kyber KEM for example still needs the formalization of the Kyber PKE and a connection to the qrhl-tool implementation of the FO transform. My thesis project of formalizing the Kyber PKE takes a step in this direction.

5 Hardness Assumptions

The mathematical basis for most crypto systems are hardness assumptions. A hardness assumption is a mathematical problem that is thought (or proven) to be hard to solve, but easy to verify. The common hardness assumptions in classical cryptography are prime factorization and the discrete logarithm problem.

For example, the well-known Rivest-Shamir-Adleman (RSA) crypto system is based on the prime factorization problem. Taking the product of two very large prime numbers, the prime factorization problem asks to find the prime factors knowing just the product. With classical computers, it would take too much time to calculate the prime factors if the factors are large enough. Therefore, it would also take too much time to break RSA, yielding a cryptographic security.

As large-scale quantum computers can solve prime factorization using Shor's algorithm [104], the RSA cryptosystem will be broken. Shor's algorithm can also break the discrete logarithm problem used as a hardness assumption in the Diffie-Hellman key exchange [104]. Therefore, we need other hardness assumptions to base our new PQC on. Lattice problems have been studied for decades and offer a variety of hard problems that can be used in cryptography. For example, the SVP and CVP have been the first lattice problems considered for new cryptosystems [49].

However, just having a worst-case hardness for hardness assumptions is not enough. For example, the k-colourability of graphs is NP-hard in the worst-case but is polynomial in expected time for the average-case [35]. Therefore, in cryptography, we often use the term average-case hardness, to state that a problem is not only hard in the worst-case, but on average.

But how can we show that a problem is hard? The go-to method is to show a hardness reduction, that is to reduce a known hard problem to the problem under consideration. More formally, we consider decision problems (problems that ask for a yes/no answer) given by a set of instances. For example, the decision problem "Is $n \in \mathbb{N}$ a prime number?" constitutes a decision problem with the set of instances being the natural numbers. We call instances with the outcome "yes" the YES-instances. Following the example, the set of prime numbers is the set of YES-instances. We denote a decision problem given the set of instances Γ and YES-instances Y by $Y \subseteq \Gamma$ (if the set of instances is clear from context, we may also just say "the problem Y"). Let us now define the problem reduction formally:

Definition 2 (Problem reduction)**.** Let $A \subseteq \Gamma$ and $B \subseteq \Delta$ be two decision problems. A **reduction from** A **to** B is a function $f : \Gamma \to \Delta$ that can be computed in polynomial time and that fulfils:

$$\forall a \in \Gamma : a \in A \Leftrightarrow f(a) \in B$$

If a problem A is NP-hard, a reduction from A to B proves NP-hardness for B as well.

In the formalization, we focus on the property $\forall a \in \Gamma : a \in A \Leftrightarrow f(a) \in B$ and omit a formal treatment of the polynomial time property. The reason is that formalizing time properties is not easy and the framework needed (NREST [53] in Isabelle) was only added to the AFP after the end of this project part.

5.1 Shortest and Closest Vector Problems

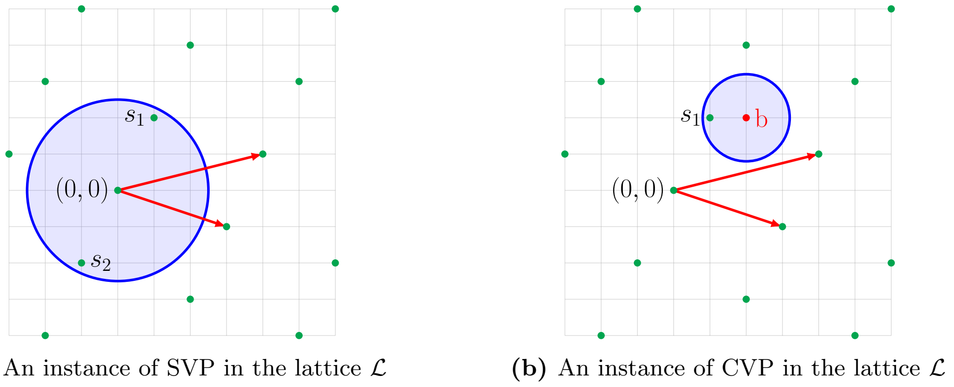
As discussed in the previous section, the SVP and CVP were the first lattice problems considered for cryptography. Let us now define them more formally. Even though the SVP and CVP can also be stated as search problems, we focus on the decision problem in this thesis. Recall the definition of lattices (Definition 1) from Section 4.2.

Definition 3 (Shortest Vector Problem)**.** Let $\mathcal{L}$ be a lattice in $\mathbb{Z}^n$ and k an estimate. The **Shortest Vector Problem** (SVP) in decision form states:
Decide whether there exists a vector $v \in \mathcal{L}$ with $v \neq 0$ and $\|v\| \leq k$.

Definition 4 (Closest Vector Problem)**.** Let $\mathcal{L}$ be a lattice in $\mathbb{Z}^n$, $b \in \mathbb{Z}^n$ a target vector and k an estimate. The **Closest Vector Problem** (CVP) in decision form states:
Decide whether there exists a vector $v \in \mathcal{L}$ such that $\|v - b\| \leq k$.

In the above definitions, $\|\cdot\|$ denotes a norm on $\mathbb{Z}^n$. The most common norms in the context of lattice problems are p-norms. The p-norm (for $p \geq 1$) is defined as $\|x\|_p := \sqrt[p]{\sum_{i=1}^n |x_i|^p}$ for $x \in \mathbb{Z}^n$. Examples include the Manhattan-norm for $p = 1$ and the Euclidean norm for $p = 2$. The infinity norm $\|\cdot\|_\infty$ is defined as the pointwise limit for $p \to \infty$. It can be proven that $\|x\|_\infty = \max_i |x_i|$ for $x = (x_1, \dots, x_n)^T \in \mathbb{Z}^n$.

Example 5.1.1. Let us look at an example of instances of the SVP and CVP. Consider the lattice $\mathcal{L}$ from Example 4.2.1. In Figure 5.1, the lattice $\mathcal{L}$ is depicted by green points with the origin marked by $(0, 0)$. The red vectors are basis vectors of the lattice. The blue circle represents the estimate k.



(a) An instance of SVP in the lattice $\mathcal{L}$ **(b)** An instance of CVP in the lattice $\mathcal{L}$

Figure 5.1: Exemplary instances of lattice problems.

For the SVP in Figure 5.1a, the blue circle is centred around the origin. The SVP asks whether there exists a lattice point (a green dot) that is different from the origin with norm less or equal k (in/on the blue circle). This instance is a YES-instance since s_1 and s_2 fulfil this property. Note that there are always two shortest vectors, namely s_1 and its negative $s_2 = -s_1$.

For the CVP in Figure 5.1b, the blue circle is centred on the target vector b (the red point). The CVP asks if the is a lattice point with distance to b less or equal to k (in/on the blue circle around b). This instance is a YES-instance since s_1 fulfils this property.

In Isabelle, the SVP and CVP are defined as the set of YES-instances, whereas the type defines the set of all instances. For example, the SVP is defined as the set of tuples $(\mathcal{L}, k)$ where $\mathcal{L}$ is a lattice and $\exists v \in \mathcal{L}. \|v\| \leq k$. Similarly, the CVP is defined as the set of tuples $(\mathcal{L}, b, k)$ where $\mathcal{L}$ is a lattice and $\exists v \in \mathcal{L}. \|v - b\| \leq k$.

The reductions to SVP and CVP use the well-known Partition and Subset Sum problems. Let us briefly define these problems formally.

Definition 5 (Subset Sum problem)**.** Let $a_1, \ldots, a_n$ and s be integers. The **Subset Sum problem** in decision form states:
Decide whether there exists a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = s$.

Definition 6 (Partition problem)**.** Let $a_1, \ldots, a_n, s$ be integers. The **Partition problem** in decision form states:
Decide whether there exists a partition of $\{1, \ldots, n\}$ into subset I and $J := \{1, \ldots, n\} \setminus I$ such that $\sum_{i \in I} a_i = \sum_{j \in J} a_j$.

As an intermediate step in the reduction to the SVP, we use the Bounded Homogeneous Linear Equations problem. We define it in the following. The scalar product over $\mathbb{Z}^n$ is denoted by $\langle \cdot, \cdot \rangle$.

Definition 7 (Bounded Homogeneous Linear Equations problem)**.** Let $b \in \mathbb{Z}^n$ be a vector and k a positive integer. The **Bounded Homogeneous Linear Equations** (BHLE) problem in decision form states:
Decide whether there exists an $x \in \mathbb{Z}^n \setminus \{0\}$ with $\|x\|_\infty \leq k$ such that $\langle b, x \rangle = 0$.

In the next section, we summarize one major contribution to this thesis: the formalization of hardness reductions of the SVP and CVP. The reduction chains are from Subset Sum to CVP (in any p-norm for $p \geq 1$) and from Partition to BHLE to SVP (only in infinity norm).

5.2 Paper 1: Formalizing Hardness Reductions for Lattice Problems

In the paper “Verification of NP-hardness Reduction Functions for Exact Lattice Problems” [67], we discuss the formal verification of the NP-hardness reductions for SVP and CVP in the infinity norm (for CVP also in any p-norm with $p \geq 1$). This paper is joint work with Tobias Nipkow and can be found in Appendix A. Due to a number of inaccuracies and problems in the original proofs uncovered by the formalization, we give examples where proofs fail and fill the gaps when necessary. Our paper [67] goes along the lines of the pen-and-paper proofs presented by Micciancio and Goldwasser [82] and Van Emde Boas [110].

First of all, the paper introduces the mathematical background, defining the general terminology of problem reductions, lattices and the SVP and CVP. The SVP and CVP are reduced from the well-known Partition and Subset Sum problems. As an intermediate step, we also formalize the BHLE problem.

The CVP and SVP are the first problems considered in lattice theory. The paper [67] only treats the exact problems, even though approximation versions are more widely used in real-world cryptography. However, since our paper [67] presents the first formalization of hardness reductions for lattice problems at all, we need to start with the most basic reductions. Even in these well-known foundations, we find a plethora of inaccuracies and even errors in the proofs. Fortunately, we can fix all the gaps and present a full and foundational formalization.

Another main limitation of the reductions is the norm under consideration. We formalize the reduction of the CVP in all p-norms, including the infinity norm. However, a deterministic reduction proof for the SVP only exists for the infinity norm. Ajtai [3] gave a randomized reduction for the SVP in the Euclidean norm. Since there is no formalism for randomized reductions in Isabelle yet, implementing this proof was out of scope.

For computability reasons, the reductions only consider lattices with bases over the integers. Bases over the rationals can always be represented by integer bases as well (by multiplying with the least common multiple of the divisors).

The main part of the paper describes the formalization of the reductions from Subset Sum to CVP and from Partition to BHLE to SVP in the infinity norm. The proof of Subset Sum to CVP follows [82, Chapter 3.2, Thm 3.1]. The pen-and-paper source claims the proof to be true for all $p \geq 1$, including $p = \infty$. However, we find a gap where the proof breaks for the infinity norm. Adding another entry to the reduction function solves this problem. In private communication, Micciancio (an author of [82]) suggested using an additional constant solving the problem as well. The reduction of CVP for p-norms with $1 \leq p < \infty$ is also formalized. There are no problems in the pen-and-paper proof in this case.

For the proposed reduction of CVP to SVP in [82], we give a counter-example showing that the proof is invalid. Therefore, we turn to the original NP-hardness reduction for the SVP in infinity norm by Van Emde Boas [110], taking the BHLE as intermediate step. Note that the terminology of [110] from the 80s is slightly different than the modern terminology (closest / shortest / nearest vector problems).

First, we formalize a reduction from Partition to BHLE in the infinity norm. We adapt the pen-and-paper version for better formalization, restructuring the proof outline and uncovering several inaccuracies. The main problems during formalization were: finding rigorous proofs for proof steps based on intuition; handling index sets, especially when "omitting" an element (which is intuitively easy, but the formalism in Isabelle is way more complex); handling different number systems (which requires a lot of formalism in Isabelle); working with huge sums (rewriting big sums in Isabelle can be tedious, especially when the automation fails).

Second, we formalize the reduction from BHLE to SVP in the infinity norm as well. However, we give examples where the reduction is not entirely correct: altering the reduction function solves these problems. We outline why these alternations are necessary and give counter-examples for when the original proof [110] fails.

In the end, we shortly consider the time complexity of the formalized reduction function. The time complexity is not formalized in Isabelle.

Topics for future work in the area of hardness reductions are developing a framework for randomized reductions in Isabelle in order to a able to formalize more complex reductions, or to formalize other reductions for example for the approximation version of SVP and CVP or other lattice problems.

5.3 Learning With Errors Problem

In the introduction to this chapter, we observed the importance of average-case hardness in cryptography. The results formalized in the previous section, however, were only worst-case reductions. So how can we use this to get interesting results for cryptography? The break-through for lattice problems in cryptography was the worst-case to average-case hardness reduction from the worst-case approximate Shortest Independent Vector Problem (SIVP) to the average-case Shortest Integer Solution (SIS) problem by Ajtai [2]. The approximate SIVP is an extension of the approximation version of the SVP. The SIS problem asks to find a short integer solution to a system of linear equations over $\mathbb{Z}_q$ (where q is prime).

With this starting point on average-case hard problems, many other average-case hard problems useful for cryptography have been developed. The most important problem for this thesis is the LWE. It is the hardness assumption for the crypto system Kyber and the new standard ML-KEM. A hardness reduction to average-case hardness of the LWE was given by Regev [97]. Let us define the LWE formally. Here, the *centred binomial distribution* β_η is defined by choosing η values c_i with $P(c_i = 1) = P(c_i = -1) = 1/4$ and $P(c_i = 0) = 1/2$ and returning the value $\sum_{i=1}^{\eta} c_i$.

Definition 8 (Learning With Errors problem)**.** Let $A \in \mathbb{Z}_q^{m \times k}$ be a matrix over the ring of integers modulo q where all entries are taken uniformly at random from $\mathbb{Z}_q$. Let β_η be the centred binomial distribution with values in $[-\eta, \eta]$. We draw an error term $e \in \mathbb{Z}_q^m$ and secret $s \in \mathbb{Z}_q^k$ randomly and entry-wise from β_η and calculate $t = As + e$. The pair (A, t) is then called a Learning With Errors instance. The **Learning With Errors** (LWE) problem in decision form states:
Decide whether an instance (A, t) is a Learning With Errors instance or is drawn uniformly at random from $\mathbb{Z}_q^{m \times k} \times \mathbb{Z}_q^m$.

Without the error term, the Gauss algorithm can easily solve the system of linear equations given by A and t. However, adding the error term in the LWE instance makes the secret hard to recalculate. A straightforward algorithm for solving the LWE uses a maximum likelihood method. It can be shown that after $\mathcal{O}(n)$ samples, the secret can be approximated such that only one solution in R_q^m remains. This may take $2^{\mathcal{O}(n\log(n))}$ time. The best known algorithm for solving the LWE is by Blum, Kalai and Wassermann [26] using only $2^{\mathcal{O}(n)}$ samples and time. More detailed analyses of the LWE are described by Regev [98].

A big problem of using the LWE in cryptography are the large key sizes it implies. Therefore, algebraic methods to get better information density are used. Lyubashevsky, Peikert and Regev [78] extended the LWE reduction to a version of the LWE over polynomial rings. Albrecht [7] then extended the hardness reduction from the ring LWE to a LWE over modules, allowing vectors over polynomial rings. This module LWE takes advantage of both the polynomial and vector worlds. More formally, the module LWE is defined as follows.

Definition 9 (module Learning With Errors problem)**.** Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$ be the polynomial ring over $\mathbb{Z}_q$ factored by the ideal generated by x^n+1. Let $A \in R_q^{m\times k}$ be a matrix where all entries are taken uniformly at random from R_q. Let β_η be the centred binomial distribution with values in $[-\eta, \eta]$. We draw an error term $e \in R_q^m$ and secret $s \in R_q^k$ randomly entry- and coefficient-wise from β_η and calculate $t = As + e$. The pair (A, t) is then called a module Learning With Errors instance. The **module Learning With Errors** (mLWE) problem in decision form states:
Decide whether an instance (A, t) is a module Learning With Errors instance or is drawn uniformly at random from $R_q^{m\times k} \times R_q^m$.

We can write the mLWE as a game against an adversary. This is depicted in Figure 5.2. With passing time, the mLWE game generates two instances: the mLWE instance and a random instance (marked by the grey parentheses). Then, a coin is tossed to decide which instance is shown to the adversary. The adversary must guess which instance was revealed. He wins if he guesses correctly and loses otherwise. This is how the mLWE is formalized in Isabelle.

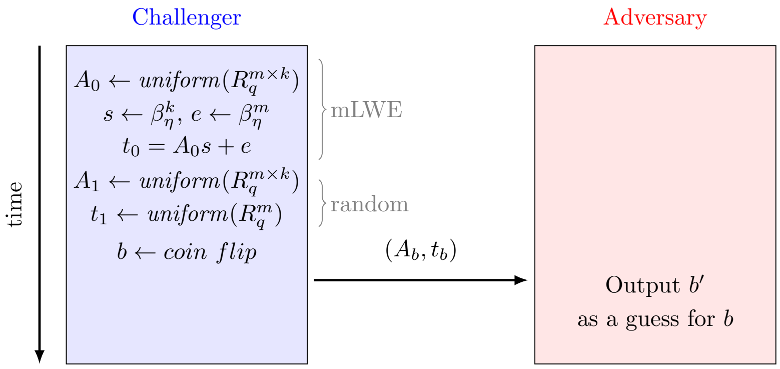

Challenger
Adversary
time
$A_0 \leftarrow uniform(R_q^{m\times k})$
$s \leftarrow \beta_\eta^k,\ e \leftarrow \beta_\eta^m$
$t_0 = A_0 s + e$
$A_1 \leftarrow uniform(R_q^{m\times k})$
$t_1 \leftarrow uniform(R_q^m)$
$b \leftarrow coin\ flip$
mLWE
random
(A_b, t_b)
Output b'
as a guess for b

Figure 5.2: The LWE in game form.

This mLWE is an essential hardness assumption in PQC. It underlies Kyber, Dilithium and the new standard ML-KEM. In the following chapter, we will look more closely at the Kyber PKE on which the Kyber KEM and the standard ML-KEM are based.

6 Kyber — An Example of Post Quantum Cryptography

The first version of the crypto system Kyber [28, 14] was published in 2017 and was submitted to NIST's first round of the standardization process. It was a development of the NewHope key exchange mechanism [8]. However, D'Anvers found an essential security flaw concerning the compression of the public key early on [28]. Therefore, the scheme was changed for the second round [13] omitting the key compression. Furthermore, the modulus was reduced to balance the change of the key size. For the third round [12], Kyber included small tweaks to increase performance such as splitting the variable for the centred binomial distribution into two parts and improving the sampling of the public key matrix. Implementations of Kyber can be found online [102].

Since NIST announced Kyber as a winner to be standardized as the first PQC for encryption [5], Kyber took center stage of PQC research. Having tighter and formally verified security guarantees now gained even more importance.

This chapter deals with a formal verification of the PKE behind Kyber in Isabelle, where I uncovered an essential flaw in the calculation of security guarantees [66, 64]. Also, my work [66, 61, 62] was the first published verification of the Kyber PKE in a theorem prover. Shortly after, Barbosa et al. [10] published a formalization of Kyber in EasyCrypt together with a verified implementation in Jasmin. My work is mostly complementary to Barbosa et al. [10] since I focus on the correctness and security verification whereas Barbosa et al. focus on the verification of the implementation to the specification.

Let us try to understand what is going on in the Kyber crypto system in more detail. For now, we will strip away all technical details and focus solely on the idea behind Kyber. Let us consider Figure 6.1. As Kyber works over a module, the figure depicts matrices and vectors over the module $R_q = \mathbb{Z}_q[x]/(x^n + 1)$. The matrix A and vector t are the public key (in blue) generated together with the secret key s (in green) in the key generation by Alice. Here, t is calculated using an additional error term e (in yellow). The key generation is an instance of the mLWE from Definition 9. In the encryption, Bob uses a personal secret key r (in green) and the public key (in blue) adding error terms e_1 and e_2 (in yellow) and the message (in grey) in the last column. The result is the ciphertext (u, v) (in red). Alice can then decipher (u, v) using her secret key s (in green). She gets the message back with an additional (small) error (in yellow). If the error is small enough, it will only affect the least important bits and we can read off the message using a rounding step. The reader may verify that the decryption indeed yields only message and error terms.

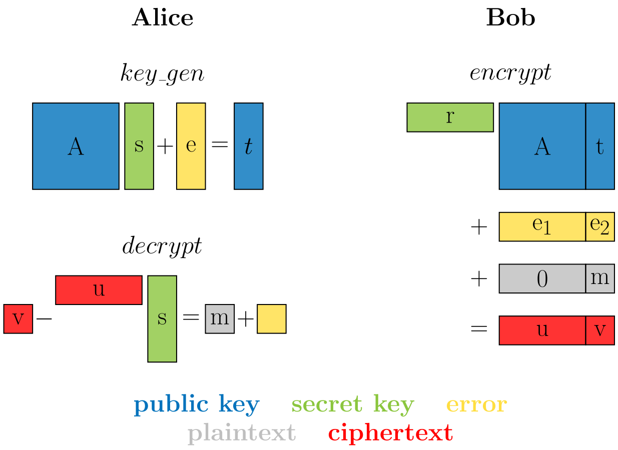


Figure 6.1: The idea behind the Kyber PKE.

6.1 Kyber Algorithms

The actual scheme is a bit more complicated. In order to decrease the ciphertext size, a compression function is used. In this section, we define the Kyber algorithms more formally. First of all, we need to define the compression function. This describes reducing a value to only d bits where d is called the compression depth.

Definition 10 (Compression and Decompression in Kyber)**.** Let d be the compression depth with $2^d < q$ (where q is the modulus in the ring R_q). Then the **compression and decompression** functions are defined as follows:

$$\operatorname{comp}_d(x) = \left\lceil \frac{2^d \cdot x}{q} \right\rfloor \mod 2^d$$

$$\operatorname{decomp}_d(x) = \left\lceil \frac{q \cdot x}{2^d} \right\rfloor$$

Formalizations of the compression and decompression function can be found in [61, Compress.thy].

Another feature intrinsic to Kyber is the “norm” function $\|\cdot\|_\infty$ on R_q. This “norm” is different than anticipated: it is not simply the infinity norm of the representatives in $\mathbb{Z}[x]/(x^n+1)$, but the infinity norm of the *centred* representatives. Unfortunately, the centring gets in the way of the absolute homogeneity, with the result that the function $\|\cdot\|_\infty$ as originally defined [28] is only a pseudo-norm. The pseudo-norm is formalized in [61, Abs_Qr.thy]

This yields a gap in Kyber’s correctness proof [65, Section 6], since homogeneity cannot be applied. With my formalization, I show how to fill this gap considering the implications on corner-cases [65].

We can finally define the Kyber PKE algorithms. These algorithms are probabilistic since we first sample the keys and errors and then calculate the key generation, encryption and decryption functions. We denote by $x \leftarrow \chi$ that x is drawn from the distribution χ, by $\mathrm{unif}(S)$ the uniform distribution on a finite set S and by $A; B$ the sequential execution of first A and then B.

Definition 11 (Kyber PKE)**.** Let R_q be the ring $\mathbb{Z}_q[x]/(x^n+1)$ where q is prime and n a positive integer such that $n = 2^{n'}$ for some $n' \in \mathbb{N}$. Let β_η be the centred binomial distribution on $[-\eta, \eta]$. Let d_u and d_v be the compression depths of the ciphertext u and v, respectively. Then the **key generation, encryption and decryption of Kyber's PKE** are:

$$\mathrm{key\_gen} = \left(\begin{array}{l} A \leftarrow \mathrm{unif}(R_q^{k\times k}); \\ s \leftarrow \beta_\eta^k; \\ e \leftarrow \beta_\eta^k; \\ t = As + e; \\ pk = (A, t); \\ sk = s; \\ \text{return } pk\ sk \end{array} \right)$$

$$\mathrm{encrypt}(pk, m) = \left(\begin{array}{l} r \leftarrow \beta_\eta^k; \\ e_1 \leftarrow \beta_\eta^k; \\ e_2 \leftarrow \beta_\eta; \\ (A, t) = pk; \\ u = A^T r + e_1; \\ v = t^T r + e_2 + \lceil q/2 \rfloor m; \\ c = (\mathrm{compress}_{d_u}(u), \mathrm{compress}_{d_v}(v)); \\ \text{return } c \end{array} \right)$$

$$\mathrm{decrypt}(sk) = \left(\begin{array}{l} (u^*, v^*) = c; \\ u = \mathrm{decomp}_{d_u}(u^*); \\ v = \mathrm{decomp}_{d_v}(v^*); \\ m = \mathrm{comp}_1(v - s^T u); \\ \text{return } m \end{array} \right)$$

The Kyber PKE is formalized in Isabelle [61, 62] in two steps: First, we consider the deterministic calculation only. Second, we add the probabilistic choices of the variables. All fixed parameters are subsumed in the locale context `kyber_spec` [61, Kyber_spec.thy], that can be instantiated for various parameter sets (some examples can be found in [61, Kyber_Values.thy]). The module R_q is formalized as the type `'a qr` [61, Kyber_spec.thy]. The deterministic calculations for the key generation, encryption and decryption are defined as `key_gen`, `encrypt` and `decrypt` [61, Crypto_Scheme.thy].

The second step of including the probability distributions on the input is more complicated to formalize. Here, we need some more theory, namely that of (sub-)probability mass functions, the Giry monad and generative probabilistic values. A **probability mass functions** f is the probability distribution of a discrete random variable X with weight one, i.e. $f(x) = P(X = x)$ and the weight $\sum_x f(x) = 1$. A **sub-probability mass function** f is allowed to have weight less than one, i.e. $\sum_x f(x) \leq 1$. The Isabelle type `spmf` (see [37] for an overview) of sub-probability mass functions can formalize stateless probabilistic algorithms by their distributions. **Generative probabilistic values** (`gpv` [73]) are an Isabelle type class for handling probabilistic algorithms together with a state. The most important concept for handling probabilistic algorithms are monads [91, 92], e.g. the **Giry monad** [47] for probabilities. The monadic structure allows us to consecutively bind two probability distributions. This allows us to consider distributions of entire probabilistic algorithms. Using these formalisms, we formalize the Kyber algorithms in Definition 11 as `pmf_key_gen` and `pmf_encrypt` [62, Correct.thy]. The decryption stays deterministic, thus we need no separate definition.

In the following section, we consider the correctness property of Kyber.

6.2 δ-Correctness

As Kyber obfuscates the message and secret key using errors, we have to make sure that the errors are not too large to interfere with the decryption of the message in the end. Indeed, we can only consider the δ-correctness of the Kyber PKE where δ denotes a bound on the estimated failure probability. Formally, we define a δ-correct PKE as the following.

Definition 12 (δ-correct PKE)**.** Consider a PKE $\mathcal{K}$ given by the algorithms *key_gen*, *encrypt* and *decrypt*. Let $\mathcal{M}$ be the message space. Then the PKE $\mathcal{K}$ is **δ-correct** iff

$$\mathbb{E}\left[\max_{m\in\mathcal{M}} \mathbb{P}[\mathit{decrypt}(sk, \mathit{encrypt}(pk, m)) \neq m]\right] \leq \delta$$

where the expectation $\mathbb{E}$ is taken over the keys (sk, pk) generated by the *key_gen* algorithm.

The δ-correctness terminology is formalized in the locale `pke_delta_correct` [62, Delta_Correct.thy]. When instantiating the locale with a PKE, we can then show the property `delta_correct` for a specific correctness error bound δ. The correctness bound δ is an important security parameter. For real-world crypto schemes, it is essential to show that the bound δ is negligibly small.

6.3 Classical Indistinguishability under Chosen Plaintext Attack

For a crypto system to be of use in real-world scenarios, we need basic security properties. These security properties usually describe a common attack scenario and require the crypto system not to leak information to the attacker. One of the most common security properties is the indistinguishability under chosen plaintext attack (IND-CPA).

The IND-CPA security property requires that an adversary cannot distinguish two ciphertexts even if he could choose the plaintexts. Usually, attacks are modelled in game form: The challenger plays a game against an adversary who tries to attack the cryptosystem and gain information. For the IND-CPA, we consider the game depicted in Figure 6.2. The game proceeds as follows: First, the challenger generates a public and secret key pair and gives the public key to the adversary. Then, the adversary may

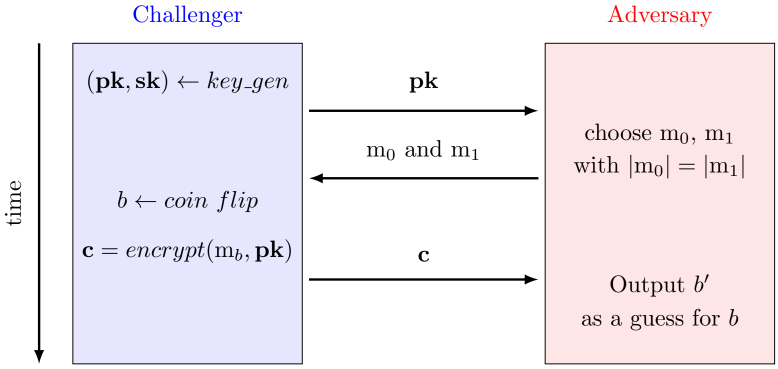

Challenger
Adversary
time
$(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathit{key\_gen}$
$\mathbf{pk}$
choose m_0, m_1
with $|\mathrm{m}_0| = |\mathrm{m}_1|$
m_0 and m_1
$b \leftarrow \mathit{coin\ flip}$
$\mathbf{c} = \mathit{encrypt}(\mathrm{m}_b, \mathbf{pk})$
$\mathbf{c}$
Output b'
as a guess for b

Figure 6.2: The IND-CPA security game

choose two messages of same length. The challenger now flips a coin on which message to encrypt and sends the encryption to the adversary. The adversary must then guess which message way encrypted.

If a crypto system is not secure against IND-CPA, the adversary can distinguish two ciphertexts even when choosing the plaintexts. This may leak partial or structural information. For example, in a block ciphers where some block occurs multiple times the attacker can distinguish repeating blocks and may gain structural knowledge. Even though this does not immediately imply that the key is leaked or the cipher is broken, it still may be pose a threat depending on the security context. As an example: The enigma machine was broken by analysing reoccurring structures in the ciphertext.

For a crypto system to be secure against IND-CPA, the attacker may not gain any substantial amount of information. That is, the difference between the output of an adversary in the IND-CPA game and a coin flip must be negligibly small. In order to prove this, we typically use a number of game-hops transforming the IND-CPA game into a game of a known hard problem. In our case, the IND-CPA game for Kyber is transformed into the mLWE game (which is known to be NP-hard, i.e. the difference of distinguishing between mLWE/uniformly random instances and a coin flip is negligible).

The IND-CPA game is formalized as a locale in CryptHOL [76, IND_CPA.thy]. It can be instantiated by any PKE, for example the key generation, encryption and decryption of the Kyber PKE.

Many implementations of crypto systems use pseudo-random function families (PRFs) to extend a small random seed to a large (equally random) output. A PRF is a family of deterministic functions where an instance of the PRF cannot be distinguished from a purely random function by any efficient algorithm. In cryptography, such purely random functions are called **random oracles**. Since many security proofs require randomness instead of determinism, the **random oracle model** proposes that a change from a deterministic PRF to random oracles is valid and unnoticeable for an adversary. An introduction to the ROM can be found in [83]. As an example of the application of the ROM: In the case of Kyber, the public key part A is generated by a pseudo-random function extending a seed. The ROM allows us to exchange the PRF of the seed by a random oracle with the target distribution. In our formalization, we therefore assume this seed extension to be uniformly random in the ROM.

In this chapter, we only consider classical, polynomial-time adversaries. However, Kyber is supposed to be also IND-CPA secure against quantum attackers. We will go into more detail on quantum attackers and security proofs against them in Chapter 7.

6.4 Paper 2: Formalizing Kyber

In my paper "Verification of Correctness and Security Properties for CRYSTALS-Kyber" [66], I present a formalization of the Kyber PKE, its δ-correctness and IND-CPA security proof. A full version can be found in Appendix B. The main contribution is evidence for a miscalculation in the estimation of the correctness error, leading to a counterexample for the correctness error bound δ as originally defined [28]. These findings were validated in private communication with Kyber authors. Furthermore, my formalization is the first published and publicly available formalization of the Kyber PKE with proofs of the correctness and IND-CPA security in a theorem prover.

Almeida et al. [10, 9] simultaneously developed a formalization of the Kyber specification in EasyCrypt and of the respective implementation in Jasmin. Their paper [10] of June 2023 focuses on the formalization of the implementation and the verification of the implementation with respect to the specification in EasyCrypt. It does not contain a verified proof of correctness or security properties of the specification of Kyber in EasyCrypt. In August 2024, shortly after my paper was published, the EasyCrypt Team [9] also published a formalization of the correctness and security properties of Kyber and the extension to ML-KEM in EasyCrypt. In accordance to my findings, the original correctness error bound could not be formalized. The EasyCrypt formalization [9] takes the unreduced error bound (that I also use), splits it into two parts and additionally bounds one of the error terms to allow easier approximations. Therefore, the bound in EasyCrypt should be larger than the bound that I proved. In comparison with the works in EasyCrypt, my paper shows a detailed analysis of why the original δ-correctness proof does not work out, giving several counter-examples for small dimensions. Furthermore, the formal proofs in Isabelle are all foundational, in contrast to the EasyCrypt proofs.

First of all, the paper [66] discusses the formalization of the context of the Kyber algorithms. This includes the modelling of the polynomial quotient ring R_q used in Kyber as a type class, as well as a model of the parameter sets as a locale. A locale bundles all assumptions on parameters and can be instantiated for various parameter sets. Therefore, the formalization is valid for several security levels, e.g. Kyber768 and Kyber1024. For the third round implementation of Kyber512 [12], the parameter η for the centred binomial distribution was split into two different values. This split has not been formalized and it remains unclear whether the security proof still remains valid with this split. The formalization in EasyCrypt [10] also omits this split.

Other basics in the formalization include the formalization of the mLWE game, the compression and decompression functions and the norm-like function used in the correctness error bound. During the formalization I found a gap in the correctness proof due to the misconception that the norm-like function (that is called a norm in [28]) is actually only a pseudo-norm. I give an example in a border case where this pseudo-norm breaks the reasoning in the correctness proof. However, I propose an alternative proof for this problem by including the assumption $q \equiv 1 \mod 4$ on the modulus q. This assumption is fulfilled by all moduli q that are used for the number-theoretic transform (NTT) in Kyber. The NTT for Kyber is also formalized together with its convolution theorem. A more detailed discussion is presented in my preprint [65] for the FAVPQC workshop.

Then, the probabilistic sampling and the deterministic calculations in the Kyber PKE are discussed. Together, the Kyber algorithms are formalized as probabilistic algorithms in the Giry monad [47]. A shortcoming of my formalization is that we sample the matrix A (that is a part of the public key) uniformly random form $R_q^{k\times k}$. In the actual scheme, A is extended from a seed by a pseudo-random function since A to keep the size of the public key as small as possible. The pseudo-random function for extension is assumed to generate uniformly random values in the original paper [28]. As we work in the ROM, substituting PRFs by uniformly random variables is reasonable and simplifies the formalizations.

The next step in the formalization is the δ-correctness of these algorithms. However, I could not formalize the δ-correctness as originally published [28]. The reason is an error in the correctness bound estimation. After an author of Kyber validated this error in private communication, I found a counterexample in a very small parameter set and confirmed this error in a statistical analysis for various other (but still quite small) parameter sets. In my formalization, I give an alternative bound δ' for which we can formally prove correctness. However, I did not find a way to easily approximate δ'. As mentioned, the EasyCrypt team [9] found the same problems but give an alternative solution with a larger bound.

The last part of my Kyber formalization is a verification of the IND-CPA security proof against the Kyber PKE in the classical ROM. The locale for the IND-CPA game is taken from CryptHOL [76]. Again, I use the monadic representation of probabilistic algorithms, sub-probabilistic mass functions and the type for generative probabilistic values from CryptHOL [73, 76] to formalize the IND-CPA game for Kyber. Using game-hops, I formalize the reduction of the IND-CPA game for Kyber to the mLWE games. In this case, the formalization follows closely the pen-and-paper proof and no major gaps were discovered. Still, the IND-CPA security proof covers a major part of the formalization since it requires several complicated rewriting steps in the game-hops where the automation mostly fails.

The main problems encountered during the formalization of the Kyber PKE were problems and gaps in the pen-and-paper proofs. When communicating with authors of Kyber and the EasyCrypt team working on a similar formalization, it became apparent that other parties also experienced these problems. In the case of the error bound δ this also lead to a concrete counter-example to the existing pen-and-paper proof [28]. Other problems included the huge game descriptions in the IND-CPA proofs. Here, the automation often failed rewriting the whole game so that I had to break it down into several smaller steps.

As a conclusion, this formalization is a starting point for a fully foundational verification of Kyber. The next steps include formalizing the FO transform to get the Kyber KEM from the PKE, verifying the specification against the implementation or generating a verified implementation from the formalization, as well as verifying security properties against quantum adversaries.

7 Security against Quantum Adversaries

In the previous chapter, we only considered classical attackers in the security games. However, since PQC is developed to counter the threat by quantum computers, we also need to prove security against attackers with access to quantum computers. Therefore, we need to extend our adversarial model: Quantum adversaries are adversaries with access to classical and quantum computers. We often model the classical part as a sub-register of the quantum part.

The first question that arises when mentioning quantum computers is: What is the difference between classical and quantum attackers in the security context? A very intuitive example is the "reprogramming" of a function (e.g. a hash function) where we change the outcome of the function on a subset of inputs. Classically, the adversary can only realize that we reprogrammed some outputs when he queries exactly these outputs. If the input space is comparatively large with respect to the change set, the probability that the adversary notices is relatively small. However, in the quantum case, the adversary may query every input in superposition at once. Therefore, it is not clear whether we can reprogram the function at all or whether the adversary can notice the change.

In this chapter, I give a short overview on basic quantum computation and the modelling of quantum adversaries. However, we will not prove or formalize the security of Kyber in the quantum setting. This is still out of reach in Isabelle, since we lack many foundational theorems needed in the quantum world. Still, we take a step in this direction and describe the formalization of a central theorem for security proofs against quantum adversaries, namely the One-way to Hiding (O2H) Theorem.

7.1 Basics of Quantum Computation

To understand quantum attackers better, we must look deeper into quantum computing. More detailed information on quantum computing and quantum information theory can be found in introductory books to this topic (e.g. [88, 111]). First of all, what exactly is a quantum computer and what can it do? A quantum computer is a computer exploiting quantum mechanics to "calculate". We can use the laws of quantum mechanics to model a quantum computer. We will not concern ourselves with the physical realization of a quantum computer but keep to the abstract model.

Formally, a quantum computer is modelled by its state register and transitions on these states. For simplicity, we will introduce the notion of quantum states over finite dimensional Hilbert spaces only. The generalization to infinite dimensions is technically more refined and requires additional assumptions for working with bounded operators. An important class of operators over infinite-dimensional Hilbert spaces are **trace-class**

operators, which are operators whose trace converges and is well-defined. Generally speaking, infinite-dimensional quantum states can be represented by trace-class operators.

The smallest unit of quantum information is a **qubit**. A qubit can be represented by a complex Hilbert space vector in $\mathbb{C}^2$ or in the ket notation $|\cdot\rangle$. The computational basis of a qubit are the states $|0\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and $|1\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$, the embeddings of a classical bit into the qubit. A general qubit has the state $\alpha\,|0\rangle + \beta\,|1\rangle$ where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. Equivalently, the qubit can be represented by the vector $\left(\begin{smallmatrix}\alpha\\\beta\end{smallmatrix}\right)$ with norm 1. We call these qubits a **superposition** of the computational basis.

A quantum register is a system of multiple qubits. We embed a classical bit-string by taking the tensor product of the computational basis qubits, e.g. the bit-string 101 is represented by the tensor $|1\rangle \otimes |0\rangle \otimes |1\rangle$ or a vector in the Hilbert space $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$. A quantum register can also be in the superposition of a subset of the computational basis. For example, in a two qubit register, the states $|0\rangle \otimes |1\rangle$ and $|1\rangle \otimes |0\rangle$ are part of the computational basis, whereas the state $\frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$ is a superposition. When considering multiple qubits, another essential quantum mechanics effect comes into play: the quantum entanglement. Consider the example of a two qubit register. Then, the superposition state $\frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$ cannot be represented by a single tensor product. We call such states **entangled**. An example of a superposition that is not entangled is the state $\frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle + |0\rangle \otimes |0\rangle) = |0\rangle \otimes \left(\frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)\right)$, since we can write the state as a tensor product of two qubits.

Quantum states are typically divided into pure and mixed states. A **pure state** is a quantum state where we have full information over the outcomes. They can be represented by a complex Hilbert space vector ψ of norm 1 or its density operator $\psi\psi^*$ (where $*$ denotes the conjugate transpose). For example, all computational basis states and superpositions thereof are pure. In contrast, **mixed states** describe quantum systems whose information is not fully known. For example, when our quantum register is entangled with another register we do not have control over, our register is in a mixed state. Mixed states cannot be described by Hilbert space vectors of norm 1, but will be denoted as density operators. For example, a probability distribution on pure states ψ_i with probabilities p_i is a mixed state that can be represented by the operator $\sum_i p_i \psi_i \psi_i^*$.

In Isabelle, we work with several types:

- `'a ell2` is the type of square-summable functions denoting Hilbert space vectors indexed by the type `'a`
- `('a,'b) cblinfun` is the type of complex bounded linear functions from type `'a` to type `'b`. This denotes operators from `'a` to `'b`
- `'a update` is a type shorthand for `('a ell2, 'a ell2) cblinfun`. This denotes operators over a Hilbert space.
- `('a,'b) trace_class` is the type of trace-class operators from `'a` to `'b`.

Using these type classes, we can formalize the quantum adversarial model described in the next section.

7.2 Quantum Adversarial Model

When defining the quantum adversary, we first have to extend the classical ROM to a quantum correspondence. The **quantum random oracle model** (QROM) [27] allows the adversary to quantumly access a classical random oracle function (e.g. querying values in superposition). All our security proofs will be performed in the quantum random oracle model.

Since the classical memory is embedded in the quantum register (as the computational basis states), we need to define how the quantum computer can access the oracle.

Definition 13 (Quantum access to random oracles)**.** Let $H : X \to Y$ be a random oracle. Let X and Y be sets of classical memory embedded in the quantum register of the adversary. Then the **quantum oracle query** is a unitary U_H defined by its behaviour on the computational basis $|x\rangle \otimes |y\rangle$ of $X \otimes Y$:

$$U_H(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |H(x) + y\rangle$$

The function U_H is formalized as `Uquery H` in Isabelle. It is defined as the extension of the classical operator U_H defined on the basis $|x\rangle \otimes |y\rangle$ for $x \in X$ and $y \in Y$.

An oracle query that is performed on quantum memory must be reversible. So, both input and output domains of the oracle function H must be embedded into the quantum register. Furthermore, the oracle query itself must also be reversible. Therefore, we still need to embed the above function into the memory register.

This formalization is slightly more complicated. Let `mem` be the quantum register with the classical sets X and Y embedded as two subregisters `X` and `Y` in `mem`. We write $XY(U_H)$ for the extension of the application of U_H on `mem`. This is formalized as `(X;Y)(Uquery H)`, with the formalism of subregisters taken from the AFP entry on quantum registers [108].

Consider an adversary that has access to a random oracle d times. Then, we should model the adversary to do any computation they want before, between and after the oracle calls. Here, $\circ$ denotes functional composition.

Definition 14 (Adversary call)**.** An adversary $\mathcal{A}$ with access to an oracle function O at most d times can be modelled by functions $\{f_i\}_{i\in\{0,\dots,d\}}$. Then the **adversary call** on an input x is:

$$\mathcal{A}(x) = f_d \circ O \circ f_{d-1} \circ \cdots \circ f_1 \circ O \circ f_0(x)$$

The above definition abuses the formal notation: depending on the type of the functions f_i, the adversary can be defined for Hilbert space vectors or operators. However, the oracle query function O also has to be adapted, embedding the oracle into the quantum register and extending it to the Hilbert space vectors/operators we work with. In Isabelle, this universal definition of an adversary call is formalized by several functions, depending whether we work with pure/mixed states or on Hilbert space vectors/operators/trace-class operators.

In the setting of the O2H Theorem, we need to distinguish pure and mixed adversaries, that is adversaries that run only on pure states or that also allow mixed states. For pure adversaries, the update functions are **unitaries** U_i. Since unitaries are norm-preserving, pure states also remain pure.

For mixed adversaries, the update functions are probability distributions on pure updates. The mathematical equivalent are **Kraus maps**. A Kraus map $\mathcal{E}$ consists of a set of operators E_j with $\sum_j E_j^* E_j = \mathbb{I}$ (where $\mathbb{I}$ is the identity). An application of the Kraus map $\mathcal{E}$ on a mixed state ρ is defined as $\mathcal{E}(\rho) = \sum_j E_j \rho E_j^*$.

The formalizations of pure and mixed adversaries follow Definition 14 with different type classes and properties for unitaries or Kraus maps. For example, the function `run_pure_adv` formalizes a pure adversary on the type of Hilbert space vectors, whereas `run_pure_adv_update` lifts this definition to operators and `run_pure_adv_tc` to trace-class operators. Mixed adversaries are defined as `run_mixed_adv` over trace-class operators. Formalizations can be found online [54, Run_Adversary.thy].

Another important concept for the O2H Theorem are the **semi-classical adversary** calls (also called punctured adversary calls). Let $\mathcal{A}$ be an adversary with quantum access to an oracle H. We want to "puncture" the oracle on a change set S, i.e. detect if the adversary notices any changes in values in S. Then, $\mathcal{A}^{H \setminus S}$ defines the adversary with access to the semi-classical oracle: The oracle H is queried as a normal quantum random oracle, but additionally we measure whether the input is in the change set S.

7.3 One-way to Hiding Theorem

As described in the introduction to this chapter, "reprogramming" an oracle function in the quantum setting is not trivial. Since many classical cryptographic proofs make use of this technique, a similar method in the quantum setting is much desired. The One-way to Hiding (O2H) Theorem yields such an analogue by bounding the probability that the adversary can distinguish two games where the oracle was "reprogrammed" on a change set S.

Intuitively, the O2H Theorem [11, Theorem 1] states the following: For two oracles H and G that agree everywhere but on a change set S, we consider two games: the first is an adversarial run with access to H and the second with access to G, both at most d times. We denote by P_{find} the probability that the adversary queries values in S. Then the difference between the two games is at most $2\sqrt{(d+1)P_{\text{find}}}$. The proof of the O2H Theorem goes deep into quantum computing and may be hard to follow, even if the presentation in [11] is quite accurate. Therefore, a formalization provides more trust in the theoretical background of the O2H.

The O2H is used in many variations and in many quantum security proofs. For example, there are several quantum security proofs of the FO-transform using versions of the O2H resulting in different bounds: Kuchta et al. [69] use the measure-rewind-measure O2H; Unruh [107] or Hövelmanns et al. [57] use the semi-classical O2H. A more general version of the FO transform is the generic authenticated key exchange. Hövelmanns et al. [58] show its quantum security proof making use of the semi-classical O2H Theorem as well. The most well-known versions of the O2H Theorem include the original version [105], the semi-classical O2H [11], the double-sided O2H [22], the measure-rewind-measure O2H [69], the O2H on compressed oracles [34] and an O2H for adaptively chosen positions [50].

We formalized the following version of the semi-classical O2H Theorem. It is a slightly weaker version as [11, Theorem 1] by an additional factor of 2 in the inequality (7.1).

Theorem 7.3.1 (O2H Theorem)**.** *Let $\mathcal{A}$ be a quantum adversary with query depth d, $G, H : X \to Y$ be two oracle functions, $S \subseteq X$ a change set such that $G(x) = H(x)$ for all $x \notin S$, and z a bit-string. G, H, S and z may be chosen by a joint distribution. We denote by $\mathcal{A}^O(z)$ the adversary with access to the oracle O and input string z (here O can be replaced by both oracles G and H). Let $\mathcal{A}^{H\setminus S}$ be the adversary with access to the semi-classical oracle. Let* Find *denote the event that we measured an oracle input in S. Let furthermore:*

$$\begin{aligned}
P_{\text{left}} &= Pr[b = 1 : b \leftarrow \mathcal{A}^H(z)] \\
P_{\text{right}} &= Pr[b = 1 : b \leftarrow \mathcal{A}^G(z)] \\
P_{\text{find}} &= Pr[\text{Find} : \mathcal{A}^{H\setminus S}(z)]
\end{aligned}$$

*Then the **One-way to Hiding** (O2H) **Theorem** states that*

$$\left|P_{left} - P_{right}\right| \leq 4\sqrt{(d+1) \cdot P_{find}} \tag{7.1}$$

$$\left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \leq 2\sqrt{(d+1) \cdot P_{find}}$$

In the following section, we summarize the contribution of formalizing the O2H in Isabelle.

7.4 Paper 3: Formalizing the One-way to Hiding Theorem

In the paper "Formalizing the One-way to Hiding Theorem" [55], we present a formalization of the semi-classical O2H Theorem by Ambainis, Hamburg and Unruh [11] in Isabelle. This paper is joint work with Dominique Unruh and can be found in Appendix C. The formalizations can be found online [54]. This is the first foundational formalization of the O2H. Furthermore, the paper extends the O2H Theorem to infinite-dimensional Hilbert spaces and non-terminating adversaries. For the O2H Theorem with mixed states, an alternative and novel proof is given.

For the formalization of the O2H, we first extend some foundations. Building on the formalization of quantum registers [108] and Kraus maps [109, `qrhl-tool/isabelle-thys/Kraus_Maps.thy`], we develop a formal model of quantum adversaries in Isabelle. Here, we distinguish between pure and mixed adversaries, i.e. adversaries that calculate on pure states only or that also allow mixed states. We also take account of the embedding of a classical oracle function in the quantum register and their oracle queries. The adversary runs are also generalized to density operators over infinite-dimensional Hilbert spaces.

A shortcoming of the formalization of the oracle queries by the adversary is that the formalization only allows sequential queries instead of potentially parallel queries. The reason is that the type of parallel queries would depend on a parameter (i.e. the number of parallel queries in each oracle call). Since Isabelle does not have dependent types, this formalization needs careful treatment.

The formalization of the O2H proof is split into two parts (as is the pen-and-paper proof): First, the O2H is proven for pure states and pure adversaries only. Second, the O2H for mixed states and mixed adversaries is separated into many instances of the pure O2H. For the first step, the formalization closely follows the pen-and-paper proof in [11]. However, for the second step, we give an alternative proof. The pen-and-paper version [11] heavily uses the Bures distance and fidelity of quantum states. Formalizing these in Isabelle would have taken too much time, so we give an alternative proof instead. A slight drawback is that the resulting final inequality is weaker by a factor of 2 (only in the inequality (7.1)). However, we generalize the O2H in other directions by extending to infinite dimensional Hilbert spaces and taking non-terminating adversaries into account. The non-termination yields an additional factor in the final bound.

A shortcoming in the formalization of the theorem is that, up to now, we only consider discrete distributions on the oracles H, G, the change set S and input z. A generalization to arbitrary distributions would certainly be possible but entails more work on convergence arguments.

The main challenge during the formalization process was adapting the proofs to work with concepts that were already formalized or did not require formalizing new concepts (like the Bures distance and fidelity). A more technical detail was the lifting of different types through several layers of generalizations: from Hilbert space vectors for pure states to operators for mixed states and trace-class operators for convergence properties. Other challenges included the exact formalism for definitions, index sets and decomposition of mixed adversaries into a linear combination of pure adversaries. Furthermore, the formalization is much more explicit concerning convergence arguments. Where the pen-and-paper proof often abstracts away the convergence of sums, traces or linear combinations, we often had to prove these facts explicitly in Isabelle.

For the future, one can now formalize the concrete bound on the probability of finding a reprogrammed value as in [11, Theorem 2]. For the application to cryptography, the formalization of different versions of the O2H is also very interesting. The most interesting follow-up work is the connection of the O2H formalization to the qrhl-tool. The biggest challenge here is aligning the adversarial models (this need solving the parallel queries issue in our model).

8 Conclusion & Outlook

This thesis takes several steps towards a foundational formalization of post-quantum cryptography. First, we give a short introduction to post-quantum cryptography in general and lattice-based schemes in more detail. Then, we give an overview on the state of the art on formalizations in post-quantum cryptography. The main contributions to this thesis can be divided in three parts:

1. **Hardness assumptions:** We introduce basic lattice problems such as the Shortest Vector Problem and the Closest Vector Problem, as well as the module Learning With Errors problem. We discuss my paper "Verification of NP-hardness Reduction Functions for Exact Lattice Problems" [67] which describes a formalization of the NP-hardness reduction functions for the Shortest and Closest Vector Problems. This is the first formalization of hardness reductions for post-quantum cryptography.
2. **Kyber:** As a prominent example for post-quantum cryptography, we consider the public key encryption scheme of Kyber. We define the algorithms, correctness terminology and security against the indistinguishability under chosen plaintext attack. My contribution is a formalization of the Kyber public key encryption scheme, its correctness and security property in the paper "Verification of Correctness and Security Properties for CRYSTALS-Kyber" [66]. A major outcome of my formalization is the discovery of an error in the correctness bound calculation. I find a counter-example with a small parameter set showing that the proof has an essential flaw and propose an alternative bound. My formalization was the first publicly available formalization of the Kyber public key encryption scheme.
3. **Quantum adversaries:** The security against quantum adversaries makes post-quantum cryptography interesting for the future. We give a short introduction to quantum computing and the quantum adversarial model. Then, we introduce the One-way to Hiding Theorem, a central theorem used in many security proofs against quantum adversaries. My paper "Formalizing the One-way to Hiding Theorem" [55] describes the first foundational formalization of the One-way to Hiding Theorem in Isabelle. Many tools for reasoning against quantum attackers are being developed, but none formalized the One-way to Hiding Theorem foundationally so far. My work provides an essential foundation for these tools.

Since the verification and formalization of post-quantum cryptography is a vast and open field, this thesis only takes small steps to broaden the foundational work needed. Still, many areas and research questions remain open for future work. Going along the lines of my contributions, new follow-up research questions can be posed:

1. **Hardness assumptions:** Can we formalize the probabilistic reductions for approximate lattice problems or the worst-case to average-case reductions needed for cryptography? Can we develop a reasonable framework for probabilistic reduction proofs in a theorem prover such as Isabelle? And can we, ultimately, show average-case NP-hardness for lattice problems used in post-quantum cryptography such as the module Learning With Errors problem?

2. **Kyber:** Can we find an alternative correctness error bound showing the claimed correctness in [28] and formalize it? And if we find a better error bound, can we also formalize its estimation? Can we formalize the security proofs against quantum computers as well (and can we do this foundationally)? Ultimately, can we get a fully verified implementation from the formalized specification or formally show that the current implementation matches the formalized specification?

3. **Quantum adversaries:** Can we foundationally formalize the needed theorems for security proofs against quantum adversaries? Once we have, can we connect these formalizations to higher-level tools developed for this purpose such as qrhl-tool? And, ultimately, can we foundationally prove security properties for Kyber in the quantum setting?

These and many more questions remain for future work in the area of verification and formalization of post-quantum cryptography. With this thesis, we have come a step closer to the formalization, verification and understanding of post-quantum cryptography.

Bibliography

[1] Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature 638, 920–926*, December 2024. `doi:10.1038/s41586-024-08449-y`.

[2] Miklós Ajtai. Generating Hard Instances of Lattice Problems. *Electron. Colloquium Comput. Complex.*, 3:99–108, 1996. `doi:10.1145/237814.237838`.

[3] Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98*, STOC '98, pages 10–19, Dallas, Texas, United States, 1998. ACM Press. `doi:10.1145/276698.276705`.

[4] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing - STOC '97*, STOC '97, page 284–293. ACM Press, 1997. `doi:10.1145/258533.258604`.

[5] Gorjan Alagic, David A. Cooper, Quynh Dang, Thinh Dang, John M. Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl A. Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Daniel Apon. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process.* NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2022-07-05 04:07:00 2022. `doi:10.6028/NIST.IR.8413-upd1`.

[6] Martin R. Albrecht, Sofía Celi, Benjamin Dowling, and Daniel Jones. Practically-exploitable Cryptographic Vulnerabilities in Matrix. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 164–181, Los Alamitos, CA, USA, May 2023. IEEE Computer Society. `doi:10.1109/SP46215.2023.10351027`.

[7] Martin R. Albrecht and Amit Deo. Large Modulus Ring-LWE $\geq$ Module-LWE. In *Advances in Cryptology – ASIACRYPT 2017*, page 267–296. Springer International Publishing, 2017. `doi:10.1007/978-3-319-70694-8_10`.

[8] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange: a new hope. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 327–343, USA, 2016. USENIX Association.

[9] José Bacelar Almeida, Santiago Arranz Olmos, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Jean-Christophe Léchenet, Cameron Low, Tiago Oliveira, Hugo Pacheco, Miguel Quaresma, Peter Schwabe, and Pierre-Yves Strub. Formally Verifying Kyber: Episode V: Machine-Checked IND-CCA Security and Correctness of ML-KEM in EasyCrypt. In *Advances in Cryptology – CRYPTO 2024*, page 384–421. Springer Nature Switzerland, 2024. `doi:10.1007/978-3-031-68379-4_12`.

[10] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Vincent Laporte, Jean-Christophe Léchenet, Tiago Oliveira, Hugo Pacheco, Miguel Quaresma, Peter Schwabe, Antoine Séré, and Pierre-Yves Strub. Formally verifying Kyber Episode IV: Implementation Correctness. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, page 164–193, June 2023. `doi:10.46586/tches.v2023.i3.164-193`.

[11] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum Security Proofs Using Semi-classical Oracles. In *Advances in Cryptology – CRYPTO 2019*, page 269–295. Springer International Publishing, 2019. `doi:10.1007/978-3-030-26951-7_10`.

[12] Roberto Maria Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.0), 01/10/2020. `https://pq-crystals.org/kyber/data/kyber-specification-round3.pdf`, accessed: 2024-10-22.

[13] Roberto Maria Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 2.0), 30/03/2019. `https://pq-crystals.org/kyber/data/kyber-specification-round2.pdf`, accessed: 2024-10-22.

[14] Roberto Maria Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation, 30/11/2017. `https://pq-crystals.org/kyber/data/kyber-specification.pdf`, accessed: 2024-10-22.

[15] Frank J. Balbach. The Cook-Levin theorem. *Archive of Formal Proofs*, January 2023. `https://isa-afp.org/entries/Cook_Levin.html`, Formal proof development.

[16] Philip Ball. Physicists in China challenge Google's 'quantum advantage'. *Nature*, 588(7838):380–380, December 2020. `doi:10.1038/d41586-020-03434-7`.

[17] Clemens Ballarin. Locales and Locale Expressions in Isabelle/Isar. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *Types for Proofs and*

Programs, pages 34–50, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. doi:10.1007/978-3-540-24849-1_3.

[18] Manuel Barbosa, Gilles Barthe, Xiong Fan, Benjamin Grégoire, Shih-Han Hung, Jonathan Katz, Pierre-Yves Strub, Xiaodi Wu, and Li Zhou. EasyPQC: Verifying Post-Quantum Cryptography. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 2564–2586, New York, NY, USA, 2021. ACM. doi:10.1145/3460120.3484567.

[19] Manuel Barbosa, François Dupressoir, Andreas Hülsing, Matthias Meijers, and Pierre-Yves Strub. A Tight Security Proof for SPHINCS+, Formally Verified. Cryptology ePrint Archive, Paper 2024/910, 2024. https://eprint.iacr.org/2024/910, preprint, to be published at ASIACRYPT24.

[20] Gilles Barthe, François Dupressoir, Benjamin Grégoire, César Kunz, Benedikt Schmidt, and Pierre-Yves Strub. EasyCrypt: A Tutorial. In *Lecture Notes in Computer Science, vol 8604*, page 146–166. Springer International Publishing, 2014. doi:10.1007/978-3-319-10082-1_6.

[21] David A. Basin, Andreas Lochbihler, and S. Reza Sefidgar. CryptHOL: Game-Based Proofs in Higher-Order Logic. *Journal of Cryptology*, 33(2):494–566, January 2020. doi:10.1007/s00145-019-09341-z.

[22] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter Proofs of CCA Security in the Quantum Random Oracle Model. In *Theory of Cryptography*, page 61–90. Springer International Publishing, 2019. doi:10.1007/978-3-030-36033-7_3.

[23] Bruno Blanchet. A Computationally Sound Mechanized Prover for Security Protocols. *IEEE Transactions on Dependable and Secure Computing*, 5:193 – 207, 01 2009. doi:10.1109/TDSC.2007.1005.

[24] Bruno Blanchet. CryptoVerif: Cryptographic protocol verifier in the computational model, 2024. https://bblanche.gitlabpages.inria.fr/CryptoVerif/, accessed: 2024-07-17.

[25] Bruno Blanchet and Charlie Jacomme. Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges. In *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*, pages 515–528, Los Alamitos, CA, USA, jul 2024. IEEE Computer Society. doi:10.1109/CSF61375.2024.00050.

[26] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003. doi:10.1145/792538.792543.

[27] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In *Advances in*

Cryptology – ASIACRYPT 2011, page 41–69. Springer Berlin Heidelberg, 2011. `doi:10.1007/978-3-642-25385-0_3`.

[28] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS — Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367, 2018. `doi:10.1109/EuroSP.2018.00032`.

[29] Ralph Bottesch, Jose Divasón, Max W. Haslbeck, Sebastiaan J. C. Joosten, René Thiemann, and Akihisa Yamada. A verified LLL algorithm. *Archive of Formal Proofs*, February 2018. `https://isa-afp.org/entries/LLL_Basis_Reduction.html`, Formal proof development.

[30] Chad Boutin. NIST Announces First Four Quantum-Resistant Cryptographic Algorithms, 2022. `https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms`, accessed: 2024-07-23.

[31] David Butler and Andreas Lochbihler. Sigma protocols and commitment schemes. *Archive of Formal Proofs*, October 2019. `https://isa-afp.org/entries/Sigma_Commit_Crypto.html`, Formal proof development.

[32] Wouter Castryck, Tanja Lange, Chloe Martindale, Joost Renes, and Lorenz Panny. CSIDH: An efficient post-quantum commutative group action, 2018. `https://csidh.isogeny.org/`, accessed: 2024-07-23.

[33] Ming-Shing Chen, Jintain Ding, Matthias Kannwischer, Jacques Patarin, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang. PQCRainbow, 2020. `https://www.pqcrainbow.org/`, accessed: 2024-07-23.

[34] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability, 2019. URL: `https://arxiv.org/abs/1904.11477`, `doi:10.48550/ARXIV.1904.11477`.

[35] Martin E. Dyer and Alan M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451–489, December 1989. `doi:10.1016/0196-6774(89)90001-1`.

[36] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM. In *Progress in Cryptology – AFRICACRYPT 2018*, page 282–305. Springer International Publishing, 2018. `doi:10.1007/978-3-319-89339-6_16`.

[37] Manuel Eberl, Johannes Hölzl, and Tobias Nipkow. A Verified Compiler for Probability Density Functions. In Jan Vitek, editor, *Programming Languages and Systems*, pages 80–104, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. doi:10.1007/978-3-662-46669-8_4.

[38] David Jao et al. SIKE – Supersingular Isogeny Key Encapsulation, 2021. https://sike.org/, accessed: 2024-07-23.

[39] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. doi:10.1038/s41586-019-1666-5.

[40] Manuel Eberl et al. Archive of formal proofs, 2024. https://www.isa-afp.org/, accessed: 2024-06-21.

[41] Simon Rosskopf et. al. poly-reductions, 2024. https://github.com/rosskopfs/poly-reductions/tree/master/Karp21, accessed: 2024-10-18.

[42] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon, 2021. https://falcon-sign.info/, accessed: 2024-08-30.

[43] FrodoKEM team. FrodoKEM, 2023. https://frodokem.org/, accessed: 2024-08-28.

[44] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Advances in Cryptology — CRYPTO' 99*, page 537–554. Springer Berlin Heidelberg, 1999. doi:10.1007/3-540-48405-1_34.

[45] Lennard Gäher and Fabian Kunze. Mechanising Complexity Theory: The Cook-Levin Theorem in Coq. In Liron Cohen and Cezary Kaliszyk, editors, *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193, pages 20:1–20:18, Dagstuhl, Germany, Jan 2021. Schloss Dagstuhl -Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITP.2021.20.

[46] Michael Garey and David Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.

[47] Michèle Giry. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, page 68–85. Springer Berlin Heidelberg, 1982. doi:10.1007/bfb0092872.

[48] GitHub. EasyCrypt, 2022. https://github.com/EasyCrypt/easycrypt, accessed: 2024-07-26.

[49] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology — CRYPTO '97*, page 112–131. Springer Berlin Heidelberg, 1997. doi:10.1007/bfb0052231.

[50] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight Adaptive Reprogramming in the QROM. In *Advances in Cryptology – ASIACRYPT 2021*, page 637–667. Springer International Publishing, 2021. doi:10.1007/978-3-030-92062-3_22.

[51] Florian Haftmann and Makarius Wenzel. Constructive Type Classes in Isabelle. In Thorsten Altenkirch and Conor McBride, editors, *Types for Proofs and Programs*, pages 160–174, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74464-1_11.

[52] Shai Halevi. A plausible approach to computer-aided cryptographic proofs. Cryptology ePrint Archive, Paper 2005/181, 2005. https://eprint.iacr.org/2005/181.

[53] Maximilian P. L. Haslbeck. NREST: Nondeterministc RESult monad with Time. *Archive of Formal Proofs*, September 2024. https://isa-afp.org/entries/NREST.html, Formal proof development.

[54] Katharina Heidler and Dominique Unruh. One-way to Hiding Formalization – Formalizing the O2H Theorem in Isabelle, 2024. https://zenodo.org/doi/10.5281/zenodo.14215773, accessed: 2024-12-20. doi:https://doi.org/10.5281/zenodo.14278513.

[55] Katharina Heidler and Dominique Unruh. Formalizing the one-way to hiding theorem. In *Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP '25, page 243–256, New York, NY, USA, 2025. ACM. doi:10.1145/3703595.3705887.

[56] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*, page 267–288. Springer Berlin Heidelberg, 1998. doi:10.1007/bfb0054868.

[57] Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Failing gracefully: Decryption failures and the Fujisaki-Okamoto transform. Cryptology ePrint Archive, Paper 2022/365, 2022. https://eprint.iacr.org/2022/365.

[58] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic Authenticated Key Exchange in the Quantum Random Oracle Model. In *Public-Key Cryptography – PKC 2020*, page 389–422. Springer International Publishing, 2020. doi:10.1007/978-3-030-45388-6_14.

[59] Andreas Hülsing, Matthias Meijers, and Pierre-Yves Strub. Formal Verification of Saber's Public-Key Encryption Scheme in EasyCrypt. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 622–653, Cham, 2022. Springer Nature Switzerland. doi:10.1007/978-3-031-15802-5_22.

[60] Florian Kammüller, Markus Wenzel, and Lawrence Paulson. Locales A Sectioning Concept for Isabelle. In Yves Bertot, Gilles Dowek, Laurent Théry, André Hirschowitz, and Christine Paulin, editors, *Theorem Proving in Higher Order Logics*, pages 149–165, Berlin, Heidelberg, 09 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-48256-3_11.

[61] Katharina Kreuzer. CRYSTALS-Kyber. *Archive of Formal Proofs*, September 2022. https://isa-afp.org/entries/CRYSTALS-Kyber.html, Formal proof development.

[62] Katharina Kreuzer. CRYSTALS-Kyber Security. *Archive of Formal Proofs*, December 2023. https://isa-afp.org/entries/CRYSTALS-Kyber_Security.html, Formal proof development.

[63] Katharina Kreuzer. Hardness of Lattice Problems. *Archive of Formal Proofs*, February 2023. https://isa-afp.org/entries/CVP_Hardness.html, Formal proof development.

[64] Katharina Kreuzer. Kyber error bound, 2023. https://github.com/ThikaXer/Kyber_error_bound, accessed: 2023-12-19.

[65] Katharina Kreuzer. Verification of the (1–δ)-Correctness Proof of CRYSTALS-KYBER with Number Theoretic Transform. Cryptology ePrint Archive, Paper 2023/027, 2023. https://eprint.iacr.org/2023/027.

[66] Katharina Kreuzer. Verification of Correctness and Security Properties for CRYSTALS-KYBER. In *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*, volume 2283 of LNCS, page 511–526. IEEE, July 2024. doi:10.1109/csf61375.2024.00016.

[67] Katharina Kreuzer and Tobias Nipkow. Verification of NP-Hardness Reduction Functions for Exact Lattice Problems. In *Automated Deduction – CADE 29*, page 365–381. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-38499-8_21.

[68] KU Leuven ESAT/COSIC. Saber, 2022. https://www.esat.kuleuven.be/cosic/pqcrypto/saber/resources.html, accessed: 2022-11-15.

[69] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shi-Feng Sun. Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security. In *Advances in Cryptology – EUROCRYPT 2020*, page 703–728. Springer International Publishing, 2020. doi:10.1007/978-3-030-45727-3_24.

[70] Vincent Laporte. Jasmin, 2024. https://github.com/jasmin-lang/jasmin/wiki, accessed: 2024-07-26.

[71] Arjen. Lenstra, Hendrik Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *MATH. ANN*, 261:515–534, 1982. doi:10.1007/BF01457454.

[72] Christina Lindenberg and Kai Wirt. SHA1, RSA, PSS and more. *Archive of Formal Proofs*, May 2005. https://isa-afp.org/entries/RSAPSS.html, Formal proof development.

[73] Andreas Lochbihler. CryptHOL. *Archive of Formal Proofs*, May 2017. https://isa-afp.org/entries/CryptHOL.html, Formal proof development.

[74] Andreas Lochbihler and S. Reza Sefidgar. A tutorial introduction to CryptHOL. Cryptology ePrint Archive, Paper 2018/941, 2018. https://eprint.iacr.org/2018/941.

[75] Andreas Lochbihler and S. Reza Sefidgar. Constructive Cryptography in HOL: the Communication Modeling Aspect. *Archive of Formal Proofs*, March 2021. https://isa-afp.org/entries/Constructive_Cryptography_CM.html, Formal proof development.

[76] Andreas Lochbihler, S. Reza Sefidgar, and Bhargav Bhatt. Game-based cryptography in HOL. *Archive of Formal Proofs*, May 2017. https://isa-afp.org/entries/Game_Based_Crypto.html, Formal proof development.

[77] Joey Lupo. cryptolib, 2024. https://github.com/JoeyLupo/cryptolib/tree/main, accessed: 2024-07-26.

[78] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In *Advances in Cryptology – EUROCRYPT 2010*, page 1–23. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-13190-5_1.

[79] Robertas Maleckas, Kenneth G. Paterson, and Martin R. Albrecht. Practically-exploitable Vulnerabilities in the Jitsi Video Conferencing System. Cryptology ePrint Archive, Paper 2023/1118, 2023. https://eprint.iacr.org/2023/1118.

[80] Robert J McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *Deep Space Network Progress Report*, 42(nr.44):114–116, 1978.

[81] Ralph Charles Merkle. *Secrecy, authentication, and public key systems.* PhD thesis, Stanford, CA, USA, 1979. AAI8001972.

[82] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems.* Springer US, Boston, MA, 2002.

[83] Arno Mittelbach and Marc Fischlin. *The Theory of Hash Functions and Random Oracles: An Approach to Modern Cryptography.* Springer International Publishing, 2021. doi:10.1007/978-3-030-63287-8.

[84] Stephan Müller. Lean Crypto Library, 2024. `https://leancrypto.org/`, accessed: 2024-07-26.

[85] Technische Universität München and Cambridge University. Isabelle, 2022. `https://isabelle.in.tum.de/index.html`, accessed 2022-07-26.

[86] Hartmut Neven. "meet willow, our state-of-the-art quantum chip", 2024. `https://blog.google/technology/research/google-willow-quantum-chip/`, accessed: 2024-12-10.

[87] Phong Nguyen. Cryptanalysis of the goldreich-goldwasser-halevi cryptosystem from crypto '97. In *Advances in Cryptology — CRYPTO' 99*, page 288–304. Springer Berlin Heidelberg, 1999. `doi:10.1007/3-540-48405-1_18`.

[88] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[89] Tobias Nipkow and Gerwin Klein. *Concrete Semantics with Isabelle/HOL.* Springer, 2014. `http://concrete-semantics.org`, accessed: 2024-10-22.

[90] Tobias Nipkow, Lawrence Paulson, and Markus Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[91] nLab authors. monad. `https://ncatlab.org/nlab/show/monad`, November 2022. Revision 97.

[92] nLab authors. monads of probability, measures, and valuations. `https://ncatlab.org/nlab/show/monads%20of%20probability%2C%20measures%2C%20and%20valuations`, November 2022. Revision 32.

[93] National Institute of Standards and Technology. NIST - Post-Quantum Cryptography, 2023. `https://csrc.nist.gov/projects/post-quantum-cryptography`, accessed: 2023-05-12.

[94] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard.* NIST, August 2024. `doi:10.6028/nist.fips.204`.

[95] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard.* NIST, August 2024. `doi:10.6028/nist.fips.203`.

[96] Adam Petcher and Greg Morrisett. The Foundational Cryptography Framework. In *Principles of Security and Trust*, page 53–72. Springer Berlin Heidelberg, 2015. `doi:10.1007/978-3-662-46666-7_4`.

[97] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC05. ACM, May 2005. `doi:10.1145/1060590.1060603`.

[98] Oded Regev. The learning with errors problem (invited survey). In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 191–204, 2010. `doi:10.1109/CCC.2010.26`.

[99] Peter Schwabe. NewHope – Post-quantum key encapsulation, 2020. `https://newhopecrypto.org/index.shtml`, accessed: 2024-08-28.

[100] Peter Schwabe. Ntru – a submission to the nist post-quantum standardization effort, 2020. `https://ntru.org/`, accessed: 2024-07-23.

[101] Peter Schwabe. CRYSTALS – Cryptographic Suite for Algebraic Lattices – Dilithium, 2022. `https://pq-crystals.org/dilithium/`, accessed: 2024-07-23.

[102] Peter Schwabe. CRYSTALS – Cryptographic Suite for Algebraic Lattices – Kyber, 2022. `https://pq-crystals.org/kyber/`, accessed: 2024-07-23.

[103] Peter Schwabe. SPHINCS+, 2023. `https://sphincs.org/`, accessed: 2024-07-23.

[104] Peter Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, SFCS-94. IEEE Comput. Soc. Press, 1994. `doi:10.1109/sfcs.1994.365700`.

[105] Dominique Unruh. Revocable quantum timed-release encryption. In *Advances in Cryptology – EUROCRYPT 2014*, page 129–146. Springer Berlin Heidelberg, 2014. `doi:10.1007/978-3-642-55220-5_8`.

[106] Dominique Unruh. Quantum Relational Hoare Logic. *Proceedings of the ACM on Programming Languages*, 3:1–31, 02 2018. `doi:10.1145/3290346`.

[107] Dominique Unruh. Post-Quantum Verification of Fujisaki-Okamoto. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 321–352, Cham, 2020. Springer International Publishing. `doi:10.1007/978-3-030-64837-4_11`.

[108] Dominique Unruh. Quantum and Classical Registers. *Archive of Formal Proofs*, October 2021. `https://isa-afp.org/entries/Registers.html`, Formal proof development.

[109] Dominique Unruh. qrhl-tool, 2024. `https://github.com/dominique-unruh/qrhl-tool`, accessed: 2024-07-16.

[110] Peter van Emde Boas. Another NP-Complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. tech. report 81-04. Technical report, Mathematisch Instituut, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands, 1981.

[111] John Watrous. *The Theory of Quantum Information.* Cambridge University Press, April 2018.

[112] A. Whitley. Cryptographic Standards. *Archive of Formal Proofs*, June 2023. `https://isa-afp.org/entries/Crypto_Standards.html`, Formal proof development.

A Paper 1: Verification of NP-Hardness Reduction Functions for Exact Lattice Problems

Reference. Katharina Kreuzer and Tobias Nipkow. *Verification of NP-Hardness Reduction Functions for Exact Lattice Problems.* In Automated Deduction – CADE 29, page 365–381. Springer Nature Switzerland, 2023. DOI: `10.1007/978-3-031-38499-8_21`.

Synopsis. This paper describes the formalization of the NP-hardness reduction functions for the Shortest and Closest Vector Problems in Isabelle. Inaccuracies and gaps in the original proofs are discovered, endorsed by concrete counter-examples, and alternative proofs are shown. A full summary of the paper is discussed in Section 5.2.

Contributions. I have contributed all formalizations for this project. Ideas for the formalization were discussed with Tobias Nipkow. The paper was written together with Tobias Nipkow.

Verification of NP-hardness Reduction Functions for Exact Lattice Problems *

Katharina Kreuzer[0000−0002−4621−734X] and
Tobias Nipkow[0000−0003−0730−515X]

Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany

Abstract. This paper describes the formal verification of NP-hardness reduction functions of two key problems relevant in algebraic lattice theory: the closest vector problem and the shortest vector problem, both in the infinity norm. The formalization uncovered a number of problems with the existing proofs in the literature. The paper describes how these problems were corrected in the formalization. The work was carried out in the proof assistant Isabelle.

Keywords: verification · NP-hardness · lattice problems · integer programming.

1 Introduction

In recent years, algebraic lattices have received increasing attention for their use in post-quantum cryptography. Algebraic lattices are additive, discrete subgroups of $\mathbb{R}^n$, i.e. a set of points in $\mathbb{R}^n$ with certain structures. One can also define lattices over finite fields, rings or modules as used in many modern post-quantum crypto systems such as the CRYSTALS suites, NTRU and Saber.

Two problems form the very basis for computationally hard problems on lattices, namely the closest vector problem (CVP) and the shortest vector problem (SVP). Given a finite set of basis vectors in $\mathbb{R}^n$, the set of all linear combinations with integer coefficients forms a lattice. In optimization form, the SVP asks for the shortest vector in the lattice and the CVP asks for the lattice vector closest to some given target vector, both with respect to some given norm.

When working over the reals, the p-norm (for $p \geq 1$) is defined as $\sqrt[p]{\sum_i |x_i|^p}$. The most common examples are the Euclidean norm $\|x\|_2$ and the infinity norm $\|x\|_\infty = \max_i\{|x_i|\}$, which is the limit for $p \to \infty$.

We have formalized, corrected and verified a number of NP-hardness proofs from the literature, uncovering a number of mistakes along the way. The first NP-hardness proof of the CVP and SVP in infinity norm is due to van Emde-Boas [7]. For other norms (especially for the Euclidean norm), there is only a randomized reduction for the NP-hardness of the SVP so far [2]. For the CVP,

* This work was supported by the Research Training Group GRK 2428 CONVEY of the German Research Council (DFG).

NP-hardness has been shown in any p-norm for $p \geq 1$. One exemplary proof can be found in the book by Micciancio and Goldwasser [15, Chapter 3, Thm 3.1].

The CVP and SVP were the starting point for lattice-based post-quantum cryptography [16]. Moreover, the relevance of these problems can also be seen from the rich literature on approximation results. For example, the LLL-algorithm by Lenstra, Lenstra and Lovász [12] gives a polynomial-time algorithm for lattice basis reduction which solves integer linear programs in fixed dimensions. Using this reduced basis, one can find good approximations to the CVP using Babai's algorithm [3] for certain approximation factors. Still, for arbitrary dimensions, the problem remains NP-hard. Further approximation results for the CVP, SVP and integer programming can be found elsewhere [6, 9, 10, 14, 19]. These approximation problems are used in cryptography. However, we will focus on the exact CVP and SVP in this paper.

A number of more basic NP-hardness proofs have been formalized in several theorem provers so far. For example, there are formalizations of the Cook-Levin Theorem in Coq [8] and Isabelle [4]. Formalizing Karp's 21 NP-hard problems (including the Subset Sum and Partition Problems assumed to be NP-hard in this paper) in Isabelle is an ongoing project.

1.1 Contributions

In this paper we present NP-hardness proofs of the CVP and SVP in infinity norm that have been verified in a proof assistant. We roughly follow the book by Micciancio and Golwasser [15, Chapter 3, Thm 3.1] and the report by van Emde-Boas [7]. However, many problems with the original proofs were encountered during the formalization efforts. We will have a look at different approaches and their advantages or problems.

We also verified the proof of NP-hardness of the CVP for any finite $p \geq 1$ from the book by Micciancio and Goldwasser. This verification did not uncover any problems with the informal proof. Thus we do not discuss it in detail.

These formalizations were carried out with the help of the proof assistant Isabelle [17, 18] and are available online [11]. They comprise 5200 lines. To the authors knowledge, they are the first formalizations of hardness proofs for lattice problems. Because of the importance of the SVP and CVP and the problems in existing proofs, we consider our proofs a contribution to the foundations of verified cryptography. However, we do not claim that these hardness results directly imply quantum-resistance of any lattice-based cryptosystems.

1.2 Overview

The paper is structured as follows. Section 2 introduces the foundations. The rest of the paper is dedicated to the proofs, which are phrased as the following two polynomial time reduction chains:

- Subset Sum $\leq_p$ CVP
- Partition $\leq_p$ Bounded Homogeneous Linear Equations $\leq_p$ SVP

Subset Sum and Partition are famous fundamental problems whose NP-hardness has been proved many times in the literature and which we take for granted.

Section 3 presents the reduction of Subset Sum to the CVP. Differences between our formalization and the book by Micciancio and Goldwasser [15] are presented with examples that demonstrate problems with the original proof. Moreover, an example is given why the generalization to the SVP given in [15] does not work.

Therefore we turn to the early proof of NP-hardness of the SVP by van Emde Boas [7]. This proof uses the Bounded Homogeneous Linear Equations problem (BHLE) which is introduced in Section 4. The formalization of this proof is one of the major achievements in this paper. It posed a significant challenge since it often relied on human intuition and had to be restructured appropriately to allow a formal proof. The main proof steps are explained and difficulties in the formalization effort are described. This proof only works in infinity norm and we explain why. In Section 5, the reduction from BHLE to the SVP is given. Again, this proof was quite elaborate to formalize as there were inaccuracies and a lot of intuition was involved. Differences between the formal proof and [7] are explained by examples.

In Section 6, we have a quick look at the reduction proof for the CVP in p-norm (for finite $p \geq 1$). In the case of the SVP there only exists a randomized hardness proof in Euclidean norm by Ajtai [1] up to now.

Finally, the time complexity of the reduction functions are considered in Section 7. We conclude the paper with a short summary and outlook.

2 Foundations

This section introduces known foundations mainly to fix the terminology and notation: problem reductions, lattices, and the combinatorial problems under consideration (CVP, SVP, Partition and Subset Sum).

2.1 Problem Reductions

Formally, a *decision problem* is given by the set of *YES-instances* P and a set Γ of problem *instances*, where $P \subseteq \Gamma$. We often associate the decision problem with the set of YES-instances, when the instance set Γ is obvious and not explicitly defined. In this paper we will often phrase problems informally (e.g. "decide if p is prime") rather than give them explicitly as sets. For example, the decision problem "decide if a natural number p is prime" will be formalized in the following way: the set of problem instances is $\Gamma = \mathbb{N}$ (in Isabelle these are all elements of type *nat*); and the YES-instances are $P = \{p \in \mathbb{N} \mid p \text{ is prime}\}$ (in Isabelle this is a set of type *nat set*).

Definition 1 (Problem reduction). *Let $A \subseteq \Gamma$ and $B \subseteq \Delta$ be two problems. A function $f : \Gamma \to \Delta$ is a reduction from A to B if it fulfills the following properties:*

- $\forall a \in \Gamma.\ a \in A \Leftrightarrow f(a) \in B$
- *f can be computed in polynomial time*

If A is NP-hard, a reduction to B proves NP-hardness of B.

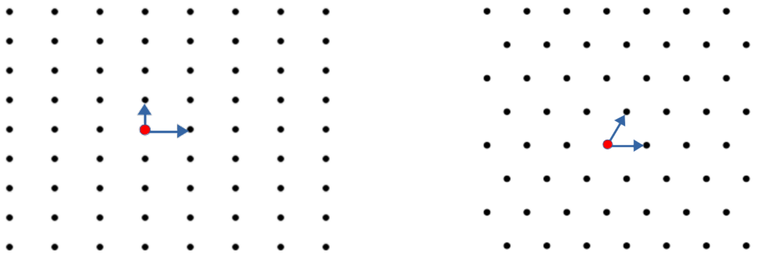
In this paper we present reduction functions informally (e.g. “an a is reduced to a b that is constructed like this”) and often with copious amounts of “…” to construct vectors etc. Of course in the formalization these reduction functions are spelled out in complete detail. Since all operations used in the reduction functions in this paper are elementary, the polynomial time property has not been formalized but is briefly discussed in Section 7. The focus of our paper are the proofs $a \in A \Leftrightarrow f(a) \in B$.

2.2 Lattice-based Computational Problems

To have a better understanding, we will first introduce lattices as such. Lattices are a structured set of points. They form an additive, discrete subgroup of $\mathbb{R}^n$. Formally, we define the following.

Definition 2 (Lattice). *Let $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^n$ be a set of linearly independent vectors. Then the integer span of A forms a lattice $\mathcal{L}$, that is:*

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} c_i a_i \mid c_i \in \mathbb{Z} \right\}$$



(a) Lattice with rectangular basis vectors (b) Lattice with triangular basis vectors

Fig. 1: Two exemplary lattices in $\mathbb{R}^2$

Example 1. In Figure 1 two examples of lattices in $\mathbb{R}^2$ are depicted. The red point is the origin. The two blue arrows show the basis vectors a_1 and a_2 that are linearly independent and span the lattice. Every integer combination of the two blue arrows is a black point, an element of the lattice.

We can see that the grid spanned by the basis vectors is discrete and has some recurring structures. These structures are determined by the basis vectors: the angle between them and their length. In Figure 1a, the angle between the two basis vectors is 90° yielding a rectangular fundamental domain. Whereas in Figure 1b, we have an angle of 60° between the basis vectors and equal length. This produces a fundamental domain of an equilateral triangle.

Indeed, the automorphism group of a lattice is a symmetry group, see Conway [5, Chapter 3.4]. For example, in Figure 1a the symmetry group is **pmm** and in Figure 1b is it **p3m1** [13].

In the rest of the text and in the formalization we restrict to finite bases over $\mathbb{Z}$ (instead of $\mathbb{R}$), simply for computability reasons. Of course bases over $\mathbb{Q}$ can be transformed into bases over $\mathbb{Z}$ by scaling all basis vectors.

The starting point of most known hard problems on lattices are the shortest vector problem and the closest vector problem. They are defined below (as usual in decision and not in optimization form). The lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ is assumed to be generated by a finite basis in $\mathbb{Z}^n$.

Definition 3 (Closest Vector Problem (CVP)). *Given a lattice $\mathcal{L}$, a vector $b \in \mathbb{Z}^n$ and an estimate k, decide whether there exists a vector $v \in \mathcal{L}$ such that*

$$\|v - b\| \leq k$$

Definition 4 (Shortest Vector Problem (SVP)). *Given a lattice $\mathcal{L}$ and an estimate k, determine whether there exists a vector $v \in \mathcal{L}$ such that*

$$\|v\| \leq k \text{ and } v \neq 0$$

2.3 Partition and Subset Sum Problems

Recall that we plan to prove NP-hardness of the CVP and SVP in the case of the infinity norm by reducing the well-studied NP-complete Subset Sum and Partition problems to the CVP and SVP. We state the definitions.

Definition 5 (Partition problem). *Given a finite list of integers $a_1, \ldots, a_n$, does there exist a partition of $\{1 \ldots n\}$ into subsets I and $\{1 \ldots n\} \setminus I$ such that*

$$\sum_{i \in I} a_i = \sum_{i \in \{1 \ldots n\} \setminus I} a_i$$

The Partition problem can be seen as a special case of the Subset Sum problem.

Definition 6 (Subset Sum problem). *Given a finite list of integers $a_1, \ldots, a_n$ and an integer s, decide whether there exists a subset S of $\{1 \ldots n\}$ such that*

$$\sum_{i \in S} a_i = s$$

2.4 Notation

Throughout the paper we use traditional mathematical notation, in particular the graphical "...". The formal Isabelle notation is by necessity more verbose (and precise). Our formalization employs both lists and vectors as a type for finite sequences and converts between them where necessary. For reasons of presentation we blur this distinction in the paper.

3 CVP

In this section, we formalize the proof of the NP-hardness of the CVP in the infinity norm along the lines of [15, p 48., Chapter 3.2, Thm 3.1] by reducing Subset Sum to the CVP.

An instance $a_1, \dots, a_n, s$ of Subset Sum is mapped to the following instance of the CVP:

$$\mathcal{L} = \begin{pmatrix} a_1 & \cdots & a_n \\ a_1 & \cdots & a_n \\ 2 & & 0 \\ & \ddots & \\ 0 & & 2 \end{pmatrix} \cdot \mathbb{Z}^n \qquad b = \begin{pmatrix} s-1 \\ s+1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad k = 1 \tag{1}$$

We proved the following theorem:

Theorem 1. *The above mapping is a reduction from the Subset Sum problem to the CVP (in infinity norm).*

This implies that the CVP (in infinity norm) is an NP-hard problem.

The reduction function used by Micciancio and Goldwasser [15] actually looks a bit different. The image of $a_1, \dots, a_n, s$ would be

$$B = \begin{pmatrix} a_1 & \cdots & a_n \\ 2 & & 0 \\ & \ddots & \\ 0 & & 2 \end{pmatrix} \qquad \mathcal{L} = B \cdot \mathbb{Z}^n \qquad b = \begin{pmatrix} s \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad k = 1 \tag{2}$$

However, the proof in [15, p.49] with this reduction function works only for $p < \infty$. It goes along the lines of the following idea: Take $k = \sqrt[p]{n}$. In the case of $p = \infty$, we get $k = \lim_{p \to \infty} \sqrt[p]{n} = 1$. Then we can formulate the following equality (equation (3.5) in [15, p.49]):

$$\|Bx - b\|_p^p = \left| \sum_{i=1}^{n} a_i x_i - s \right|^p + \sum_{i=1}^{n} |2x_i - 1|^p \tag{3}$$

Given a YES-instance $a_1, \dots, a_n, s$ of Subset Sum, there exists a vector $x = (x_1, \dots, x_n) \in \{0,1\}^n$, such that $\sum_{i=1}^n a_i x_i - s = 0$ and $|2x_i - 1| = 1$. Then $\|Bx - b\|_p^p = n$ which proves this case.

Given a YES-instance of the CVP defined by $\mathcal{L}$, t and k that are the image of $a_1, \dots, a_n, s$ under the reduction function as in (2), we get $\|Bx - b\|_p^p \leq n$. Since all values are integers, we have $|2x_i - 1| \geq 1$. It follows that $\sum_{i=1}^n a_i x_i - s = 0$ and $|2x_i - 1| = 1$. Thus, we can deduce that $a_1, \dots, a_n, s$ was indeed a YES-instance of Subset Sum.

The major problem we encountered was that this proof works fine for $p < \infty$ but for $p = \infty$, the sum in (3) becomes a maximum instead. The equation then reads

$$\|Bx - b\|_\infty = \max \left(\left| \sum_{i=1}^n a_i x_i - s \right|, |2x_i - 1| \text{ for } 1 \leq i \leq n \right)$$

This invalidates the arguments in the proof since $|\sum_{i=1}^n a_i x_i - s|$ can now be in the range $\{-1, 0, 1\}$. The constraints are too lax to ensure the equality to zero.

A solution was to alter the matrix and target vector and add another entry. The matrix and target vector we used are given in equation (1). The alternation to $s - 1$ and $s + 1$ forces a linear combination of the a_i to be exactly s in the hardness proof, since $|\sum_i c_i a_i - (s \pm 1)| \leq 1$.

After communicating with Daniele Micciancio, one of the authors of [15], he suggested using a constant $c > 1$ and the generating instance

$$\mathcal{L} = \begin{pmatrix} c \cdot a_1 & \cdots & c \cdot a_n \\ 2 & & 0 \\ & \ddots & \\ 0 & & 2 \end{pmatrix} \cdot \mathbb{Z}^n \qquad b = \begin{pmatrix} c \cdot s \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad k = 1$$

This solves the problem as well and can be implemented using e.g. $c = 2$. This technique is described later in the book [15, p.49-51] when trying to explain the NP-hardness proof for the SVP in the infinity norm.

3.1 Towards the SVP

The authors of [15] argue that the reduction argument of the SVP can be deduced generating an instance of the SVP using the Subset Sum instance $a_1, \dots, a_n, s$ in the following way. For $c > 1$, e.g. $c = 2$, take

$$B = \begin{pmatrix} c \cdot a_1 & \cdots & c \cdot a_n & c \cdot s \\ 2 & & 0 & 1 \\ & \ddots & & 1 \\ 0 & & 2 & 1 \end{pmatrix} \qquad \mathcal{L} = B \cdot \mathbb{Z}^{n+1} \qquad k = 1$$

The authors claim that every shortest vector in the image of the reduction function has -1 as last coefficient. For example, let a YES-instance of the SVP be defined by the generating matrix B of the lattice and let $x = (x_1, \dots, x_n, -1)^T$

be the coefficients such that Bx is a shortest vector. Then we know that

$$\|Bx\|_\infty = \left\| \begin{pmatrix} c \cdot (x_1 a_1 + \cdots + x_n a_n - s) \\ 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \end{pmatrix} \right\|_\infty \leq 1$$

Since $c > 1$, it follows, that $x_1 a_1 + \cdots + x_n a_n - s = 0$, which yields a solution for the given Subset Sum instance $a_1, \ldots, a_n, s$.

However, this reduction does not always work as the following example shows:

Example 2. Given the Subset Sum instance $(a_1, a_2, a_3, s) = (1, 1, 1, 1)$. This is a YES-instance, since a solution is given by $x_1 = 1$, $x_2 = 0$ and $x_3 = 0$. The basis matrix of the corresponding SVP would be (with $c > 1$)

$$B = \begin{pmatrix} c & c & c & c \\ 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 1 \end{pmatrix}$$

Take for example the vector $v = B \cdot (-1, -1, -1, 3)^T = (0, 1, 1, 1)^T$. It has infinity norm 1 and is thus a shortest vector in the lattice generated by B. However, this vector has the last coefficient 3 and not -1, even though it clearly is a shortest vector of the lattice given by B. The corresponding scaled "solution" for Subset Sum would be $(1/3, 1/3, 1/3, -1)$ but since only integer values are allowed in the solution space, this is not a solution in our sense.

We consider another example. Let the Subset Sum instance be $a'_1 = 3, s' = 1$. We can easily see that this is not a YES-instance, i.e. there exists no solution. Still, the corresponding SVP instance given via the reduction function is generated by the matrix

$$B' = \begin{pmatrix} c \cdot 3 & c \cdot 1 \\ 2 & 1 \end{pmatrix}$$

In this case the coefficients $(-1, 3)^T$ yield a shortest vector in the lattice spanned by B', since

$$\left\| B' \begin{pmatrix} -1 \\ 3 \end{pmatrix} \right\|_\infty = \left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\|_\infty \leq 1$$

Thus, B' defines a YES-instance of the SVP, but the original Subset Sum instance is not a YES-instance.

In [15], it is stated for the infinity norm that any shortest vector yields a solution for the Subset Sum Problem, which is not the case in these examples: we cannot ensure that a shortest vector always has -1 as a last coordinate.

Although the proof in [15] does not work out as expected, there is still the reduction proof by van Emde-Boas [7] which reduces a problem called the Bounded Homogeneous Linear Equation problem to the SVP in infinity norm. This will be discussed in the next two sections.

4 Bounded Homogeneous Linear Equations

A technical report by Peter van Emde-Boas [7] gives another reduction proof for the NP-hardness of the SVP in infinity norm. The author first reduces the Partition Problem to a problem called Bounded Homogeneous Linear Equation (BHLE) which is then reduced to the SVP.

Definition 7 (Bounded Homogeneous Linear Equations problem). *Given a finite vector of integers $b \in \mathbb{Z}^n$ and a positive integer k, decide whether there exists an $x \in \mathbb{Z}^n \setminus \{0\}$ with $\|x\|_\infty \leq k$ such that*

$$\langle b, x \rangle = 0$$

We have verified a reduction from Partition to BHLE, and thus BHLE is NP-hard.

Theorem 2. *There is a reduction from Partition to BHLE in infinity norm.*

The proof is carefully engineered and rather intricate. Differences to the original proof and problems encountered during the formalization are:

- Our formal proof has a different structure than the proof in the technical report [7]. Indeed, the technical report first proves the reduction of a weaker form of Partition to BHLE and then argues that “omitting” an element yields the desired result as it adds stricter constraints. In the formalization we skip this intermediate step and directly prove the existence of an appropriate reduction function.
- Steps that seem trivial in the technical report often require a long formal proof. What can be reasoned by intuition in a pen-and-paper proof has to be elaborated in the formal proof. Intuition is also sometimes used for hand-waving over small gaps or imprecisions.
- Indexing vectors and lists has been a problem in the formalization. In pen-and-paper proofs, one can argue easily about “omitting” an element of a list even though this is imprecise and often misuses the notation. In the formalization one cannot simply skip an index. All indexing functions in the formalization have to be total. “Omitting” an element can only be solved by re-indexing and re-structuring the lists in the proof.
- Numbers are interpreted in different number systems during the proof. In contrast to the original proof, the formalization has to explicitly state the digits for a change of basis and show equivalence. This leads to verbose and elaborate proofs. To make proofs easier, we use the concrete basis $d = 5$ instead of an unspecified basis $d > 4$ as in [7]. Furthermore, the number M must use the absolute values of the a_i (omission in the definition of M in [7]). The formal definition is stated below.
- The proof involved many arguments about manipulations of huge sums. Working with huge sums entails very large proof states where the existing proof automation mostly failed on. These proof states require detailed (but still readable) proofs and occasional manual instantiation of theorems. Another possible solution to get smaller proof states is to introduce local abbreviations for subterms.

Let us have a look at the proof and its difficulties in the formalization in more detail. We start from a Partition instance $a = a_1, \ldots, a_n$. Note that we ignore the trivial case $n = 0$ in this presentation (but deal with it in the formal proofs) — this means $n - 1 \geq 0$. We reduce a to a BHLE instance b as follows:

– Define

$$M = 2 \cdot (\sum_{i=1}^{n} |a_i|) + 1 \tag{4}$$

– For $1 \leq i < n$ generate a 5-tuple

$$\begin{aligned} b_{i,1} &= a_i + M \cdot (5^{4i-4} + 5^{4i-3} + 5^{4i-1}) && (5)\\ b_{i,2} &= M \cdot (5^{4i-3} + 5^{4i}) \\ b_{i,3} &= M \cdot (5^{4i-4} + 5^{4i-2}) \\ b_{i,4} &= a_i + M \cdot (5^{4i-2} + 5^{4i-1} + 5^{4i}) \\ b_{i,5} &= M \cdot (5^{4i-1}) \\ b_i &= b_{i,1}, b_{i,2}, b_{i,4}, b_{i,5}, b_{i,3} \end{aligned}$$

Note that $b_{i,3}$ has moved to the last position in b_i.

– For $i = n$ generate only a 4-tuple:

$$\begin{aligned} b_{n,1} &= a_n + M \cdot (5^{4n-4} + 5^{4n-3} + 5^{4n-1}) \\ b_{n,2} &= M \cdot (5^{4n-3} + 1) \\ b_{n,4} &= a_n + M \cdot (5^{4n-2} + 5^{4n-1} + 1) \\ b_{n,5} &= M \cdot (5^{4n-1}) && (6)\\ b_n &= b_{n,1}, b_{n,2}, b_{n,4}, b_{n,5} \end{aligned}$$

Note that

- $b_{n,3}$ is omitted from b_n to restrict the constraints necessary for the proof and
- that in $b_{n,2}$ and $b_{n,4}$ the last summand changes to a $+1$ in comparison to the other $b_{i,2}$ and $b_{i,4}$.

In summary, the entry $b_{i,3}$ is uniformly in the last position in the b_i but omitted from the final b_n.

The Partition instance a of length n is reduced to a vector b of length $5n - 1$:

$$b = (b_1, \ldots, b_{n-1}, b_n) \tag{7}$$

The NP-hardness proof now follows in three steps:

1. We need to show an auxiliary lemma.
2. We show that a YES-instance of Partition is reduced to a YES-instance of BHLE.
3. We show that the pre-image of a YES-instance of BHLE is indeed a YES-instance in Partition.

4.1 Auxiliary Lemma

As a first step, the proof needs a short auxiliary lemma from number theory.

Lemma 1. *Let $x, y, c \in \mathbb{Z}^n$ and M be an integer. Assume that $M > \sum_{i=1}^{n} |x_i|$ and that $|c_i| \leq 1$ for all $1 \leq i \leq n$. Furthermore, let the following equation hold:*

$$\sum_{i=1}^{n} c_i \cdot (x_i + M \cdot y_i) = 0 \tag{8}$$

Then we have

$$\langle c, x \rangle = 0 \quad \textit{and} \quad \langle c, y \rangle = 0$$

In this lemma, we can reinterpret $x_i + M \cdot y_i$ from (8) as a number in basis M with lowest digit x_i. Even with a coefficient c_i, the lowest digit in basis M has to be zero, as well as the rest. By splitting off the lowest digits consecutively, we can show, that indeed all digits in basis M have to equal zero.

4.2 $a \in$ Partition $\Longrightarrow b \in$ BHLE

This direction is quite easy. Let $a_1, \ldots, a_n$ be a YES-instance of partition with partitioning set I. We will show that the following vector x is a solution to the corresponding BHLE:

$$\begin{aligned}
x &= (x_1, \ldots, x_{n-1}, x_n) \\
x_i &= \begin{cases} 1, -1, 0, -1, 0 & i \in I \wedge n-1 \in I \\ 0, 0, -1, 1, 1 & i \in I \wedge n-1 \notin I \\ 0, 0, -1, 1, 1 & i \notin I \wedge n-1 \in I \\ 1, -1, 0, -1, 0 & i \notin I \wedge n-1 \notin I \end{cases} \qquad 1 \leq i < n \\
x_n &= 1, -1, 0, -1
\end{aligned}$$

We have to show that $\langle b, x \rangle = 0$. This is proven by plugging in the definitions and rearranging terms in the sum of the scalar product such that they cancel out. As a last step in the proof, we need to show that $\|x\|_\infty \leq 1$. For the infinity norm this is quite easy. However, it would not be true for other norms. For $p \geq 1$ and $p < \infty$ we have for $n \geq 1$:

$$\|x\|_p = \sqrt[p]{3n} > 1$$

Thus, the chosen constraints x only work in infinity norm.

4.3 $a \in$ Partition $\Longleftarrow b \in$ BHLE

This direction is harder. Let b be a YES-instance of BHLE. That is, there exists a nonzero x such that $\langle b, x \rangle = 0$ and $\|x\|_\infty \leq 1$. We have to show that there is a partition I on $a_1, \ldots, a_n$ with $\sum_{i \in I} a_i = \sum_{i \in \{1 \ldots n\} \setminus I} a_i$.

The proof idea works as follows. First, we apply the auxiliary lemma and get a constraint on the a_i on the one hand, and a condition on the x_i with coefficients that are powers of 5 on the other hand. Using this condition on the x_i, we generate equational constraints on the entries of x by looking at the digits in basis 5. We argue that a number equals zero if and only if all its digits are zero.

The generated equations lead to a good characterisation of x, namely the weight $w = x_{5(n-1)+1}$. From the assumption that $\|x\|_\infty \leq 1$, we deduce $|w| \leq 1$. Again, this step can only be reasoned in the infinity norm. For other p-norms, this argumentation breaks as we need the property $|w| \leq 1$ to complete the proof. Using the value of w, we can constuct a partitioning set I with the required property from the equation on the a_i.

5 SVP

Knowing that the BHLE is indeed an NP-hard problem, we reduce it to the SVP. Then we can conclude that the SVP in infinity norm is NP-hard.

Theorem 3. *There is a reduction from BHLE to the SVP in infinity norm.*

Again some difficulties were met when formalizing the proof for the above theorem. First of all, note that the terminology in [7] and nowadays is a bit different. In [7], the shortest vector problem only denotes the shortest vector problem in the Euclidean norm. What we call the shortest vector problem in the infinity norm is named closest vector problem in [7]. To make terminology even more confusing, our understanding of the closest vector problem is called the nearest vector problem in [7]. To make the notation clear, we provide a table for reference in Figure 2.

technical report [7]	our notation
closest vector problem	SVP in infinity norm
shortest vector problem	SVP in Euclidean norm
nearest vector problem	CVP

Fig. 2: Notation

A more mathematical problem encountered was that the reduction itself used in [7] was not entirely correct. In the reduction two factors $k' = k+1$ and k'' were introduced. These factors should have certain properties to allow the arguments of the reduction proof to go through. However, this is only true when tweaking these factors a bit to make the whole proof watertight. We will now have a closer look.

Given the BHLE instance $b = (b_1, \dots, b_n)$ and k, create the following SVP instance:

$$\mathcal{L} = \begin{pmatrix} 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \\ \multicolumn{3}{c}{-\,(k+1)\cdot b} & -\,k'' \end{pmatrix} \cdot \mathbb{Z}^n \qquad k = k$$

where k'' is the factor in question. In the technical report, we have

$$k'' = 2 \cdot (k+1) \cdot (\sum_i b_i) + 1$$

The following example however shows that this factor is not enough.

Example 3. Consider the BHLE instance given by $b = (1, -1)$ and $k = 1$. This is a YES-instance, since the vector $(1, 1)$ yields the expected properties.

Define the following matrices.

$$B_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix} \qquad B_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -2 & 9 \end{pmatrix} \qquad B_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 6 & -6 & 25 \end{pmatrix}$$

The associated SVP instance is the lattice generated by B_0. Then the vector $(0, 0, 1)^T$ with infinity norm 1 is a solution to the SVP instance generated by the basis matrix B_0. However, since the last entry is nonzero, this does not provide a solution for BHLE. Contrary to this example, the proof in the technical report shows that for all SVP solutions the last entry must be zero.

The reason, why the argument in the technical report breaks at this point is because $b_1 + b_2 = 0$, thus making $k'' = 1$ very small. One step to prevent this is to use the absolute values of the b_i in k'' instead. The new k''_1 we consider is

$$k''_1 = 2 \cdot (k+1) \cdot (\sum_i |b_i|) + 1$$

With this new factor k''_1 we get the generating matrix B_1 and the vector $(0, 0, 1)$ is no longer a shortest vector.

Still, this is not enough. Consider the same $b = (1, -1)$ as above, but let $k = 5$. Then we get B_2 as the generating matrix of the SVP lattice. The vector $x = (0, 5, 1)^T$ is a shortest vector whose last entry is nonzero. Again it contradicts the proof in the technical report. The reason this time is the following: the argument that $(k+1)\,(\sum_{i=1}^n x_i b_i)$ and k''_1 have different relative sizes fails. Indeed, we have

$$\left\lVert \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 6 & -6 & 25 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 5 \\ 1 \end{pmatrix} \right\rVert_\infty = \left\lVert \begin{pmatrix} 0 \\ 5 \\ -5 \end{pmatrix} \right\rVert_\infty = 5 \leq k$$

We can obtain different relative sizes of $(k+1)\,(\sum_{i=1}^n x_i b_i)$ and k''_1 by defining

$$k''_2 = 2 \cdot k \cdot (k+1) \cdot (\sum_i |b_i|) + 1 \tag{9}$$

Now we can make sure that the last entry of a solution to the SVP problem is indeed zero. For the proof of Theorem 3 we consider the reduction given by

$$\mathcal{L} = \underbrace{\begin{pmatrix} 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \\ - & (k+1) \cdot b & - & k_2'' \end{pmatrix}}_{B} \cdot \mathbb{Z}^n \qquad k = k$$

where B denotes the basis matrix generating the lattice $\mathcal{L}$ as given above.

Consider a solution $x = (x_1, \ldots, x_{n+1})$ of the SVP with $\|Bx\|_\infty \leq k$. Then we have

$$Bx = \begin{pmatrix} 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \\ - & (k+1) \cdot b & - & k_2'' \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ (k+1)(\sum_{i=1}^n x_i b_i) + x_{n+1} \cdot k_2'' \end{pmatrix}$$

As this yields a solution to the SVP, we get:

$$|(k+1)(\sum_{i=1}^{n} x_i b_i) + x_{n+1} \cdot k_2''| \leq k \tag{10}$$

Then we calculate:

$$\begin{aligned} (k+1)(\sum_{i=1}^{n} x_i b_i) + x_{n+1} \cdot k_2'' \leq (k+1)(\sum_{i=1}^{n} |x_i||b_i|) + x_{n+1} \cdot k_2'' \leq \\ \leq (k+1)k(\sum_{i=1}^{n} |b_i|) + x_{n+1} \cdot k_2'' \end{aligned}$$

Assuming that $x_{n+1} \neq 0$, we have

$$|(k+1)k(\sum_{i=1}^{n} |b_i|)| < |2 \cdot k \cdot (k+1) \cdot (\sum_{i} |b_i|) + 1| = |k_2''| \leq |x_{n+1} \cdot k_2''|$$

Thus the two summands indeed have different relative sizes and can never cancel out the other summand. This leads to a contradiction to (10). Therefore, $x_{n+1} = 0$ must be true and $(x_1, \ldots, x_n)$ constitutes a solution to the BHLE when using k_2'' as in (9).

6 Other *p*-Norms

Up to now, we have investigated lattice problems under the infinity norm. Even though this yields nice hardness results, in practice the Euclidean norm is used more often. Unfortunately, when considering *p*-norms things do not play out as nicely. In this section, we assume $1 \leq p < \infty$ whenever we talk about a specific p.

For the CVP, there is a generalisation of the proof for every p-norm in [15, p.48, Chapter 3.2, Thm 3.1] which we also formalized. Let $a_1, \dots, a_n, s$ be an instance of Subset Sum. The reduction function maps this instance to:

$$\mathcal{L} = \begin{pmatrix} a_1 \cdots a_n \\ 2 \quad\quad 0 \\ \ddots \\ 0 \quad\quad 2 \end{pmatrix} \cdot \mathbb{Z}^n \qquad b = \begin{pmatrix} s \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad k = \sqrt[p]{n}$$

Then the following theorem holds:

Theorem 4. *The above mapping is a reduction from the Subset Sum problem to the CVP in p-norm.*

This implies that the CVP in p-norm is an NP-hard problem. The outline to the proof is given in Section 3 after Theorem 1. The important difference to the infinity norm is that the bound k scales with the dimension n of the lattice.

For the SVP, there is no known deterministic NP-hardness result in the Euclidean norm, or even any p-norm. However, Ajtai [1, 2] found an interesting alternative which is quite useful for the application in cryptography, namely randomized reductions using polynomial-time probabilistic reduction functions. In cryptography, these results guarantee the hardness of "average" cases. That is, given an average instance according to a probability distribution, it will most likely be intractable.

7 Time complexity

As stated in Section 2, time complexity of the above reduction functions has not been formalized. However, we give a short explanation why all reduction functions are indeed in polynomial time.

Subset Sum to CVP: The reduction function as given in equation (1) creates $(n + 2)(n + 1) + 1$ values using only memory access or one addition. Therefore, the time complexity in this case is $\mathcal{O}(n^2)$.

Partition to BHLE: In this case, the reduction function maps the input a of length n to b as defined in equation (7). The value $k = 1$ is fixed. Then a is mapped to a vector of length $5n - 1$. When calculating the b_i, we need to calculate the value of M as in (4). As we sum over all input values, this lies in $\mathcal{O}(n)$. Each b_i can then be calculated in $\mathcal{O}(n)$ since it only contains a constant number of additions of the input with fixed cofactors (see (5) - (6)). Putting the construction of the list and the calculation of the b_i together, we find that the whole reduction function is in $\mathcal{O}(n^2)$.

BHLE to the SVP: Consider the reduction function as given in equation (5) using the value k_2'' as in (9). Calculating k_2'' requires $n+2$ memory accesses which are processed in $n + 4$ arithmetic operations, thus having a time complexity of

$\mathcal{O}(n)$. Every other entry in the matrix is calculated on $\mathcal{O}(1)$, since they contain at most two memory accesses and at most two arithmetic operations. The input generates $(n+1)^2+1$ values, of which $(n+1)(n+1)$ are in $\mathcal{O}(1)$ (namely all the zeros and ones, the vector $(k+1)\cdot a$ and the constraint k) and one is calculated in $\mathcal{O}(n)$ (namely k_2''). Thus, the whole reduction function lies in $\mathcal{O}(n^2)$.

8 Outlook

With this paper, we now have a formal proof for NP-hardness of the CVP and SVP in the infinity norm, as well as a formal proof of the CVP in p-norm (for $1 \leq p < \infty$). In the formalization process, many gaps and imprecisions in the pen-and-paper proofs were fixed. The changes to the original proofs have been elaborated with explanations and examples. Unfortunately, giving a deterministic reduction proof of the SVP in p norm for $p < \infty$ is still an open problem. Under probabilistic assumptions, Ajtai showed NP-hardness of the SVP in Euclidean norm in [2].

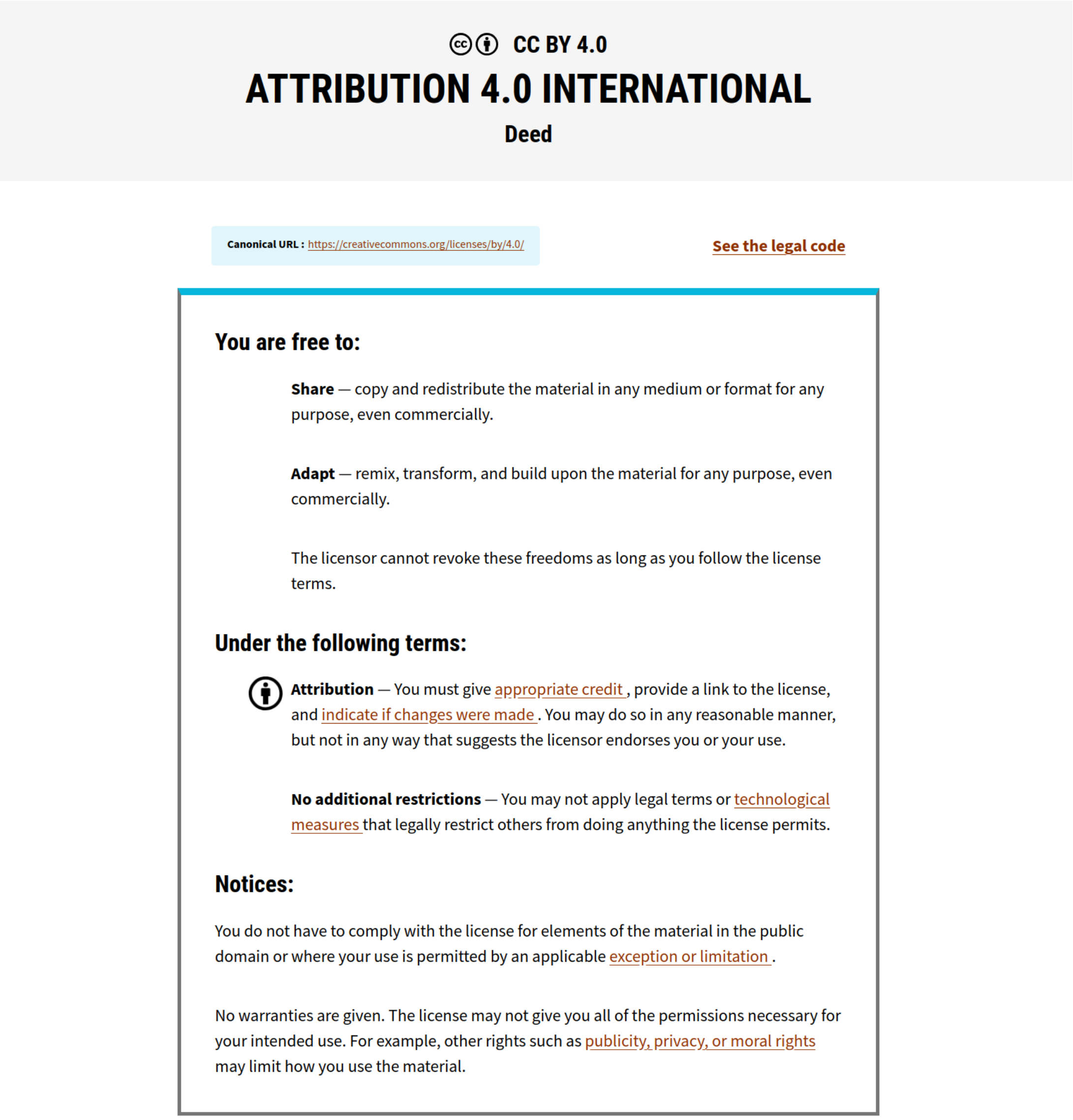
An interesting topic for future work is to develop a framework for probabilistic reductions such as in [2]. This will give the foundation to extend formalization of hardness proofs to other problems in lattice theory, especially those used in lattice-based cryptography, such as the Learning with Errors (LWE) Problem, Ring-LWE and Module-LWE. This will underline the security of many lattice-based crypto systems. Another topic for future work is to formalize the hardness proofs for approximate versions of the CVP and SVP.

Acknowledgements We thank Manuel Eberl for continuous support and fruitful discussions. The first author gratefully acknowledges the financial support of this work by the research training group ConVeY funded by the German Research Foundation under grant GRK 2428.

References

1. Ajtai, M.: Generating hard instances of lattice problems. Electron. Colloquium Comput. Complex. **3** (1996)
2. Ajtai, M.: The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In: Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98. pp. 10–19. ACM Press, Dallas, Texas, United States (1998)
3. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**, 1–13 (1986)
4. Balbach, F.J.: The cook-levin theorem. Archive of Formal Proofs (January 2023), `https://isa-afp.org/entries/Cook_Levin.html`, Formal proof development
5. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups. Springer New York (1999). `https://doi.org/10.1007/978-1-4757-6568-7`, `https://doi.org/10.1007/978-1-4757-6568-7`

6. Dinur, I., Kindler, G., Raz, R., Safra, S.: Approximating CVP to within almost-polynomial factors is NP-hard. Combinatorica **23**, 205–243 (04 2003). https://doi.org/10.1007/s00493-003-0019-y
7. van Emde Boas, P.: Another NP-Complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. tech. report 81-04. Tech. rep., Mathematisch Instituut, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands (1981)
8. Gäher, L., Kunze, F.: Mechanising complexity theory: The cook-levin theorem in coq. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPICS.ITP.2021.20, https://drops.dagstuhl.de/opus/volltexte/2021/13915/
9. Haviv, I., Regev, O.: Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing. p. 469–477. STOC '07, Association for Computing Machinery, New York, NY, USA (2007)
10. Khot, S.: Hardness of approximating the shortest vector problem in lattices. J. ACM **52**(5), 789–808 (sep 2005)
11. Kreuzer, K.: Hardness of lattice problems. Archive of Formal Proofs (February 2023), https://isa-afp.org/entries/CVP_Hardness.html, Formal proof development
12. Lenstra, A.K., Lenstra, H., Lovasz, L.: Factoring polynomials with rational coefficients. MATH. ANN **261**, 515–534 (1982)
13. Liu, Y., Collins, R.: Frieze and wallpaper symmetry groups classification under affine and perspective distortion. Tech. Rep. CMU-RI-TR-98-37, Carnegie Mellon University, Pittsburgh, PA (July 1998)
14. Micciancio, D.: The shortest vector in a lattice is hard to approximate to within some constant. In: Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280). pp. 92–98 (1998). https://doi.org/10.1109/SFCS.1998.743432
15. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems. Springer US, Boston, MA (2002)
16. Micciancio, D., Regev, O.: Lattice-based Cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5, https://doi.org/10.1007/978-3-540-88702-7_5
17. Nipkow, T., Klein, G.: Concrete Semantics with Isabelle/HOL. Springer (2014), http://concrete-semantics.org
18. Nipkow, T., Paulson, L., Wenzel, M.: Isabelle/HOL — A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002)
19. Rothvoss, T., Venzin, M.: Approximate CVP in time $2^{0.802\ n}$ – now in any norm! arXiv:2110.02387 [cs] (Oct 2021)



CC BY 4.0

ATTRIBUTION 4.0 INTERNATIONAL

Deed

Canonical URL : https://creativecommons.org/licenses/by/4.0/

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Figure A.1: Image source: CreativeCommons. `https://creativecommons.org/licenses/by/4.0/`, accessed on 2025-01-14

B Paper 2: Verification of Corretness and Security Properties for CRYSTALS-KYBER

Reference. Katharina Kreuzer. *Verification of Correctness and Security Properties for CRYSTALS-KYBER.* In 2024 IEEE 37th Computer Security Foundations Symposium (CSF), volume 2283 of LNCS, page 511–526. IEEE, July 2024. DOI: `10.1109/csf61375.2024.00016`

Synopsis. This paper describes the formalization of the Kyber public key encryption scheme, its δ-correctness and classical security against the indistinguishability under chosen plaintext attack in Isabelle. The estimation of the correctness error bound is shown to be deficient by a concrete counter-example in a small parameter sets, as well as statistical experiments in slightly larger parameter sets. An alternative error bound is found that still suffices the δ-correctness proof. A full summary of the paper is discussed in Section 6.4.

Contributions. I am the sole author of this paper. Therefore, all contributions are mine.

Verification of Correctness and Security Properties for CRYSTALS-KYBER

Katharina Kreuzer
School of Computation, Information and Technology
Technical University of Munich
Munich, Germany
0000-0002-4621-734X

***Abstract*—Since the post-quantum crypto system CRYSTALS-KYBER has been chosen for standardization by the National Institute for Standards and Technology (US), a formal verification of its correctness and security properties becomes even more relevant. Using the automated theorem prover Isabelle, we are able to formalize the algorithm specifications and parameter sets of Kyber's public key encryption scheme and verify the δ-correctness and indistinguishability under chosen plaintext attack property. However, during the formalization process, several gaps in the pen-and-paper proofs were discovered. All but one gap concerning the error bound δ could be filled. Calculations in smaller dimensions give examples where the bound δ is less than the actual error term, violating the correctness property. Since the correctness proof could be formalized up to an application of the module-Learning-with-Errors assumption, we believe that the discrepancy of the original error bound and the formalized version is relatively small. Thus the correctness could be formalized up to a minimal change to the error bound.**

***Index Terms*—post-quantum cryptography, CRYSTALS-KYBER, number theoretic transform, security, verification, Isabelle.**

I. Introduction

With large-scale quantum computers all crypto systems based on RSA and Diffie-Hellman can be broken using Shor's algorithm. Since recent developments in quantum computing lead to believe that these feasible quantum computers are not too far off in the future, methods for cryptography which are resistant even to attacks by quantum computers are hot research topics. In the course of the standardization process initialized by the National Institute of Standards and Technology (NIST) of the US, a variety of post-quantum crypto systems have been designed [33]. Most prominent are the so-called lattice-based crypto schemes.

The winner of the NIST standardization process for public key encryption (PKE) and key encapsulation methods (KEM) was announced in July 2022. It is the KEM CRYSTALS-KYBER (abbreviated as Kyber throughout this presentation) which was originally developed by Bos *et al.* [11]. In the first submission to the NIST standardization process [6], the algorithms from the original paper are extended by sampling methods using pseudorandom functions and an encoding and decoding function for mapping bits to polynomials and vice versa. A main change to the submission in the second round [5] was excluding the compression and decompression functions in the key generation and encryption functions. The reason is that a problem in the security proof for the indistinguishability under chosen plaintext attack (IND-CPA) was found by D'Anvers [11, footnote 6]. Furthermore the use of a slightly different algorithm for fast multiplication allowed the use of a smaller prime for the finite field. For the last submission in round three [4] in October 2020, some parameter changes have been made. Most notable is the change of splitting the variances of the centred binomial distribution for the error terms in the encryption. This could not be formalized since the underlying hardness assumption requires the errors to be of the same distribution. However, this only affects the proofs for Kyber512, since in Kyber 768 and Kyber1024 there is no such split. Throughout this paper, we focus on the formalization of the most recent version (namely 3rd round with security levels Kyber768 and Kyber1024) for Kyber's PKE scheme which we refer to as Kyber if not stated otherwise.

The underlying hard problem for Kyber is the module-Learning-with-Errors (module-LWE) problem. It states that it is hard to recompute a small vector when given a matrix and the matrix-vector-product perturbed by additional small errors. Without the error term, this problem can be solved by Gaussian elimination, but with the error it becomes NP-hard under certain conditions [25].

Since Kyber's key generation and encryption are based on masking the output with an error using module-LWE instances, this may result in a positive probability that the errors get too large so that we cannot decrypt correctly. We therefore need to consider δ-correctness, where δ bounds the correctness error. The correctness error is defined as the probability of an incorrect decryption in the worst case over all messages and in the mean over the generated public and secret key pairs.

As cryptography is used in many safety critical areas, security of the schemes and correctness of their mathematical proofs is crucial. The standard to ensure correctness of proofs for many years was to check and recheck proofs manually. However since humans are inevitably prone to errors, flaws in proofs may go unnoticed for years. Formalization in automated theorem provers can help to uncover such flaws, inconsistencies or simple calculus errors. Especially in cryptography, formal analysis and verification can uncover a number of

This work was supported by the Research Training Group GRK 2428 CONVEY of the German Research Council (DFG).

vulnerabilities of crypto systems or protocols. Some examples are vulnerabilities found in the Matrix messenger [1] and the Jitsi video conference tool [29] during a formalization. In recent years, formalizing proofs in cryptography has gained more and more attention. This motivates checking correctness and security properties also for post-quantum crypto systems like Kyber.

A. *Our Contribution*

With this paper, we introduce a formalization of Kyber's PKE scheme, its correctness and IND-CPA security property in Isabelle. The formalization includes the algorithms for key generation, encryption and decryption of both the original [6], [11] and the latest versions [4], [5]. Using minimal assumptions in the formalization, we allow for instantiations with various parameter sets.

During the formalization of the correctness of Kyber, we encountered two problems: Firstly, we could only verify the δ-correctness for a modified δ'. We give a counter-example for a small parameter set where the originally claimed δ [11] violates the δ-correctness property. Further experiments with different parameter sets result in similar findings. This issue has been acknowledged by Kyber authors in private communication. Secondly, we notice that the function $|| \cdot ||_\infty$ as defined in [11] is not the usual maximum norm, but only a pseudo-norm. This results in a failing proof step which can be resolved by adding an assumption on the modulus q. The additional assumption is fulfilled by all Kyber parameter sets. Overall, the correctness of Kyber could be formally proved with only a small change on the error bound.

Kyber uses the number theoretic transform (NTT) for fast multiplication. The aforementioned additional assumption is implied by assumptions on Kyber for the NTT. The NTT in the case of Kyber, as well as its convolution theorem have also been formalized for this article. However, we will not go into detail in this presentation, but include a short overview in the Appendix for the readers convenience.

The formalization is foundational, i.e. that everything is proven with respect to the higher order logic (HOL) kernel of Isabelle/HOL. The only computational assumption we make is that the underlying hardness assumption of the module-Learning-with-Errors problem (module-LWE) holds.

B. *Related Work*

A short version on the formalization of the δ-correctness of the original version of Kyber can be found in [24]. Meijers *et al.* [9], [10] announced a formalization of Kyber in EasyCrypt [14]. Furthermore, a post-quantum version of EasyCrypt called EasyPQC is being developed [8]. Recently, Almeida *et al.* [2] introduced a formalization of the implementation code to the specification in the frameworks EasyCrypt and Jasmin. The formalization in EasyCrypt/Jasmin is complementary to this presentation, since it does not verify the mathematical proofs of correctness or security properties of the specifications. To the best of the authors' knowledge, there is, up to now, no publication or publicly accessible formalization of Kybers correctness proof or the IND-CPA security proof. Private conversation with Kyber authors showed that the flaw uncovered by this formalization effort in the correctness proof was known, but a solution was not yet found.

In 2022, the NTT was verified in CryptoLine by Hwang *et al.* in [18]. CryptoLine is a tool for low-level verification of implementations which stands in contrast to our high level verification of the mathematics behind Kyber.

C. *Isabelle*

All formalizations and verifications were implemented in the theorem prover Isabelle. An introduction to Isabelle can be found in [32] and [31]. In contrast to other cryptographic verification tools, Isabelle is foundational meaning everything is proved from the axioms of higher order logic. The formalizations for this work are performed on the specification level of Kyber and are not restricted to an implementation.

Two main features in Isabelle support abstraction over a context of assumptions: The type class constraints (introduced in [15]) and explicit assumptions summarized in a context called locale (introduced in [7]). These abstractions allow instantiations with several parameter sets, making changes for example on the underlying prime (e.g. [6] to [5]) easy.

For our formalizations, we make extensive use of several libraries for Isabelle, including algebra, analysis, probability theory and CryptHOL [26] (a library for cryptography). Tutorials on the latter can be found in [27].

D. *Structure*

In this paper, we discuss the formalization and verification of Kyber and its δ-correctness proof, as well as the game-based IND-CPA security proof for Kyber. First, we have a look at the specifications and parameters of Kyber in Section II-A. We elaborate on the representation of the ring $\mathbb{Z}_q[x]/(x^n+1)$ as a type class in Isabelle. Since the formalization is independent from the actual parameters, in Section II-B we look at the instantiation of our formalization for some parameter sets. Next, we describe the formalization of the algorithms for compression, decompression, key generation, encryption and decryption of Kyber in Section III. In Section IV, we proceed with the verification of the δ-correctness proof of Kyber. Here, we recognize two problems in the proof: On the one hand, we can only show δ-correctness for a modified δ' as described in Section IV-C. We analyse why the original proof could not be formalized and how a modification on δ can fix this issue. Indeed, we showcase small dimensional examples where the proof fails for the original δ. On the other hand, we inspect a problem with the inequalities in the proof which we can solve by adding an assumption on the modulus q. This is discussed in Section IV-F. This newly found assumption is already fulfilled when working in the NTT domain. More on the formalization of the NTT on polynomials and its convolution theorem can be found in the Appendix X-D. In Section V, we give a short introduction to game-based cryptography and define the game versions of the IND-CPA security game and the module-LWE problem. As the security proof was formalized using the framework CryptHOL [26], we point out important concepts for formalizing cryptographic security proofs in Isabelle in Section VI. The formalization of the game-based security proof of Kyber against IND-CPA follows in Section VII. In the end, we give a short outlook on further research questions. The full formalization can be found in [21] and [22].

Throughout this paper, we will use bold font to highlight vectors and matrices (e.g. $\mathbf{v}$, $\mathbf{A}$) and roman font for polynomials (e.g. x).

II. Formalizing the Context of Kyber

Starting a formalization of the specification of Kyber requires a framework to state and calculate with Kyber's polynomial quotient ring. Isabelle offers possibilities to implement the framework and parameter set in a flexible way using type classes and locales.

A. Formalizing the Polynomial Quotient Ring

Let q be a prime and n a power of two, i.e., there is an n' such that $n = 2^{n'}$. Let R_q denote the ring $\mathbb{Z}_q[x]/(x^n + 1)$. This is the space where the Kyber algorithms work in. Note that $x^n + 1$ is the $2^{n'+1}$-th cyclotomic polynomial which is irreducible over the integers $\mathbb{Z}$, but reducible over the finite field $\mathbb{Z}_q$.

There are various concepts behind this construct which are not easy to formalize in Isabelle. To still be able to work over these complicated spaces without too many premises, we chose to use type class constructs.

First of all, the existing formalization of the finite field uses the type class *mod_ring* over a finite type. The modulus prime is encoded as the cardinality of the finite type. It represents the residue classes of the ring $\mathbb{Z}_q$ where q is the cardinality of the finite type.

Polynomials can be easily constructed using the *poly* type constructor. The *poly* constructor defines a polynomial to be a function from the natural numbers to the coefficient space which is 0 almost everywhere. A polynomial p in $R[x]$ is thus represented by the function of coefficients $f : \mathbb{N} \longrightarrow R$ such that $\mathrm{p} = \sum_{i=0}^{\infty} f(i)x^i$. Since p has only finitely many non-zero coefficients, f is 0 almost everywhere. For example the polynomial $\mathrm{p} = x^2 + 2$ is represented as the function f with:

$$f(i) = \begin{cases} \text{if } i = 0 \text{ then } 2 \\ \text{if } i = 2 \text{ then } 1 \\ \text{else } 0 \end{cases}$$

The most difficult part is to construct the quotient ring R_q. First, an equivalence relation needs to be established for residue classes modulo $x^n + 1$. Then, one can factor out the equivalence relation using the command *quotient_type* [19]. The concrete Isabelle formalization is explained in Appendix X-A1. The resulting structure inherits basic properties like the zero element, addition, subtraction and multiplication from the original polynomial ring through lifting and transfer [17].

Vectors are implemented using a fixed finite type as an index set. Since Isabelle does not allow dependent types, a separate finite type for indexing is used to encode the length of a vector. This idea was introduced by Harrison [16]. For example, when working with vectors in $\mathbb{Z}^k$, we use the type *(int, 'k) vec*. Here, *'k* is a finite type with cardinality exactly k used for indexing the integer coefficients.

An important fact to note when dealing with formalizations is that the functions translating between the different types always need to be stated explicitly. In the mathematical literature, this distinction is often abstracted away to enable a shorter presentation.

B. Formalizing the Parameters of Kyber

Kyber depends on a number of parameters defining the module, the compression and decompression. These are:

- $n = 2^{n'}$, the degree of the cyclotomic polynomial
- q, the prime number and modulus
- k, the dimension of vectors in the module
- d_u and d_v, the number of digits for compression and decompression of u and v, respectively

Since the framework for the context of Kyber is formalized independently from the actual parameters, we can instantiate the formalization with any parameters sufficing all required properties:

- n, n', q, k, d_u, d_v are positive integers
- $n = 2^{n'}$ is a power of 2
- $q > 2$ is a prime with $q \mod 4 = 1$ (the latter is an additional assumption and will be discussed in Section IV-F)

This is especially of interest for eventual changes in the parameter set. Furthermore, different security level implementations use different parameters. For example, the initial parameter of the modulus q in [11] is 7681, but since round two of the NIST standardization process [4], [5], Kyber uses the modulus 3329 and adapted d_u and d_v. Furthermore, different sizes k of vectors (and adapted d_u and d_v) define different security levels. The parameter sets for different security levels from the second (and third) round specification of Kyber [4], [5] can be found in Table I.

The Isabelle formalization of the parameter set can be found in Appendix X-A2. In our formalization, we instantiate the locale containing the Kyber algorithm and proof of δ-correctness with the parameter set given in Table I for Kyber768.

III. Formalizing the Kyber Algorithm

The PKE scheme Kyber consists of three algorithms: the key generation, the encryption and the decryption. The key generation produces a public and secret key pair given a random input. The keys are then applied in the en- and decryption. In order to discard some lower order bits to make the ciphertext smaller, a compression and decompression function is added. The compression function is also used to extract the message in the decryption. In the first versions of Kyber, the compression of the public key invalidates the IND-CPA security proof. Therefore, since the submission to round

Table I: Parameter set of round two and three Kyber [4], [5]

	n	n'	q	k	d_u	d_v
Kyber512 (round 2)	256	8	3329	2	10	3
Kyber768 (round 2 & 3)	256	8	3329	3	10	4
Kyber1024 (round 2 & 3)	256	8	3329	4	11	5

two of the NIST standardization process, this compression of the public key was left out. We focus on the newer versions in this presentation.

For a clearer presentation, we omit explicit type casts when they are unambiguous. For example, the embedding of integers in the reals or vice versa has an explicit type cast. An important type cast that we will state explicitly is the cast from an integer to the module R_q which we denote as the function *to_module*. In the actual formalization, all type casts are stated.

A. Input to the Algorithm

The key generation requires the inputs $\mathbf{A} \in R_q^{k\times k}$, $\mathbf{s} \in R_q^k$ and $\mathbf{e} \in R_q^k$ which are chosen randomly. $\mathbf{A}$ is chosen uniformly at random from the finite set $R_q^{k\times k}$. In the implementation, the matrix $\mathbf{A}$ is generated from a uniformly random seed via an XOF. This expansion has not been formalized. Instead, we require that $\mathbf{A}$ itself is uniformly random. $\mathbf{A}$ is also part of the public key. For elements of the secret key $\mathbf{s}$ and the error term $\mathbf{e}$, we define the centred binomial distribution β_η.

Choose η values c_i with $P(c_i = -1) = P(c_i = 1) = 1/4$ and $P(c_i = 0) = 1/2$ and return the value $x = \sum_{i=1}^{\eta} c_i$. For generating a polynomial in R_q according to β_η, every coefficient is chosen independently from β_η. Similarly, a vector in R_q^k is generated according to β_η^k by independently choosing all entries according to β_η. Both $\mathbf{s}$ and $\mathbf{e}$ are generated according to β_η^k.

For our formalization of Kyber, we use $\eta = 2$ [4], [5]. Note that in the more recent submissions of Kyber, the value η determining the variance of the centred binomial distribution was changed as well. Again, the formalization in locales allows us to easily change these values of η. However, for Kyber 512 in the third submission round [4], two separate values η_1 and η_2 have been introduced. This distinction has not been formalized. The reason is that the following definition of the module-LWE problem only allows the distribution on the elements of the error vector to be the same. Since the security proofs reduces a module-LWE instance where e_1 and e_2 appear in one vector, the formalization does not allow the splitting of η_1 and η_2.

The sampled values $\mathbf{A}$, $\mathbf{s}$ and $\mathbf{e}$ constitute an instance of the module-LWE problem which is defined in the following.

Definition 1 (Module-LWE)**.** Given a uniformly random $\mathbf{A} \in R_q^{k\times k}$ and $\mathbf{s}, \mathbf{e} \in R_q^k$ chosen randomly according to the distribution β_η^k. Let $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$, then the (decision) module-LWE problem asks to distinguish $(\mathbf{A}, \mathbf{t})$ from uniformly random $(\mathbf{A}', \mathbf{t}') \in R_q^{k\times k} \times R_q^k$.

There is a probabilistic reduction proof for the average-case NP-hardness of the module-LWE by Langlois and Stehlé [25]. Therefore, the key generation of Kyber returns a public key and secret key pair where it is (in average) NP-hard to recover the secret key from the public key alone. This property is also called the module-LWE hardness assumption.

Note that the module-LWE problem without the error term would be easy to solve using the Euclidean Algorithm. Thus, the error term cannot be reused but has to be chosen according to the distribution β_η^k again. The random choices and the reduction to the module-LWE have been formalized in the IND-CPA security proof for Kyber's PKE scheme. The NP-hardness proof of the module-LWE has not been formalized.

B. Compression and Decompression

The compression and decompression functions in Kyber help to reduce ciphertext size and obscure the message. In the decryption, the message is also extracted by a compression to one bit. In order to define these functions, we introduce a positive integer d with $2^d < q$. Thus, we have $d < \lceil log_2(q) \rceil$. In this section, we write "mod 2^d" to denote the modulo operation with modulus 2^d, yielding the unique representative in $\{0, \dots, 2^d - 1\}$.

When compressing a value x, we omit the least important bits and reduce the representation of x to d bits. Decompression rescales to the modulus q. Compression and decompression functions are defined for integers in the following way.

$$comp_d\ x = \left\lceil \frac{2^d \cdot x}{q} \right\rfloor \mod 2^d$$

$$decomp_d\ x = \left\lceil \frac{q \cdot x}{2^d} \right\rfloor$$

Note that the round function is defined as $\lceil x \rfloor = \lfloor x + \frac{1}{2} \rfloor$. The compression and decompression functions are extended to functions over $\mathbb{Z}_q$ by working with the unique representative in $\{0, \dots, q-1\}$. We denote compression and decompression over polynomials as comp and decomp and over vectors as $\mathbf{comp}$ and $\mathbf{decomp}$. They are defined to perform the compression or decompression coefficient- and index-wise, respectively.

We call the value $decomp_d\ (comp_d\ x) - x$ the compression error c_x. The rounding in the compression and decompression may introduce such a compression error. For example, consider the values $d = 2$ and $q = 5$. Then, the compression of 2 is $comp_2\ 2 = \lceil 1.6 \rfloor \mod 4 = 2$ and $decomp_2\ 2 = \lceil 2.5 \rfloor = 3$. Here, the compression error is $decomp_2\ (comp_2\ 2) - 2 = 3 - 2 = 1$. Another reason for a compression error is the modulo operation in the compression function. For example consider $d = 2$ and $q = 11$. Then the compression of 10 is $comp_2\ 10 = \lceil 3.\overline{63} \rfloor \mod 4 = 0$ and $decomp_2\ 0 = 0$. Here, the compression error for integers is $decomp_2\ (comp_2\ 10) - 10 = -10$. Interpreting this as a number over $\mathbb{Z}_{11}$, we get a compression error of 1.

In the following, for a value x, we will denote the compression of x by x^* and the decompression of the compression as x' to avoid overly lengthy expressions.

C. Key Generation, Encryption and Decryption

We now want to state the actual algorithms. For the convenience of the reader, we append the formal definitions of key generation, encryption and decryption in Isabelle in the Appendix X-A3. The calculation of the key generation is defined in the following way:

$$key\_gen\ \mathbf{A}\ \mathrm{s}\ \mathbf{e} = \mathbf{A} \cdot \mathrm{s} + \mathbf{e}$$

We denote by $\mathbf{t} = key\_gen\ \mathbf{A}\ \mathrm{s}\ \mathbf{e}$ the output of the key generation. Together, the matrix $\mathbf{A}$ and the vector $\mathbf{t}$ constitute the public key, whereas the vector s is the secret key. When we say that the public and secret key pair $(\mathbf{A}, \mathbf{t})$ and s are generated by the key generation algorithm, we mean the probabilistic program where $\mathbf{A}$, s and $\mathbf{e}$ are chosen according to their distributions, $\mathbf{t}$ is calculated by key_gen and $(\mathbf{A}, \mathbf{t})$ and s are the output.

Note that in the original version of Kyber [11], the key generation included a compression of $\mathbf{t}$. However, this resulted in a major flaw of the IND-CPA proof. Thus, since the second round of NIST's standardization [5], the compression in the key generation was omitted.

The pair $(\mathbf{A}, \mathbf{t})$ also forms an instance of the module-LWE problem. The module-LWE hardness assumption states that in average cases it is hard to recuperate the secret key s from the pair $(\mathbf{A}, \mathbf{t})$.

To encrypt a bit-string $\bar{m}$ with at most n bits, we consider the message polynomial $\mathrm{m} \in R_q$ obtained by $\mathrm{m} = \sum_{i=0}^{n-1} \bar{m}(i) x^i$. Thus, the message polynomial m only has coefficients in $\{0, 1\}$. For the encryption, we also need to generate another secret $\mathbf{r} \in R_q^k$ together with errors $\mathbf{e_1} \in R_q^k$ and $\mathrm{e}_2 \in R_q$ according to the distribution β_η^k and β_η. We then calculate the encryption:

$$\begin{aligned} &encrypt\ \mathbf{t}\ \mathbf{A}\ \mathbf{r}\ \mathbf{e_1}\ \mathrm{e}_2\ du\ dv\ \mathrm{m} = \\ &\quad (\mathbf{comp}_{du}\ (\mathbf{A}^T \cdot \mathbf{r} + \mathbf{e_1}), \\ &\quad\ \mathrm{comp}_{dv}\ (\mathbf{t}^T \mathbf{r} + \mathrm{e}_2 + \\ &\quad\quad + to\_module(\lceil q/2 \rfloor) \cdot \mathrm{m})) \end{aligned}$$

Let $\mathbf{u} = \mathbf{A}^T \cdot \mathbf{r} + \mathbf{e_1}$ and $\mathrm{v} = \mathbf{t}^T \mathbf{r} + \mathrm{e}_2 + to\_module(\lceil q/2 \rfloor) \cdot \mathrm{m}$. Then, the encryption outputs the compressed values $\mathbf{u}^*$ and v^* in a pair $(\mathbf{u}^*, \mathrm{v}^*)$. When referring to the encryption without the input of $\mathbf{r}$, $\mathbf{e_1}$ and e_2, we mean the probabilistic program that first generates $\mathbf{r}$, $\mathbf{e_1}$ and e_2 according to their distributions and then calculates the encryption function as stated above.

Using the secret key s, we can recover the message m from $\mathbf{u}^*$ and v^* in the decryption function. We extract the message as the highest bit in $\mathrm{v}' - \mathbf{s}^T \mathbf{u}'$ using the compression function with depth 1.

$$\begin{aligned} &decrypt\ \mathbf{u}^*\ \mathrm{v}^*\ \mathbf{s}\ du\ dv = \\ &\quad \mathrm{comp}_1\ ((\mathrm{decomp}_{dv}\ \mathrm{v}^*) - \\ &\quad \mathbf{s}^T (\mathbf{decomp}_{du}\ \mathbf{u}^*)) \end{aligned}$$

During the algorithms, the compression and decompression induce errors which should not affect the correctness of the decryption result. This problem is investigated in the δ-correctness proof of Kyber. The following section describes a verification of this proof in Isabelle.

IV. Verifying the δ-Correctness Proof of Kyber

To verify the δ-correctness of the specification of Kyber in Isabelle, we look at the pen-and-paper proof from [11, Theorem 1]. This proof shows the correctness of the original version of Kyber, but can also be easily adapted to the recent versions omitting the compression of the public key. Formalizations can be found in [21] and [22].

A. $\|\cdot\|_\infty$ – a Wolf in Sheep's Clothing

In order to estimate values, the authors of Kyber [11] use a function $\|\cdot\|_\infty$. However, it is defined slightly differently from what one would expect: Instead of using a regular modulo operation, the re-centred operation $\mathrm{mod}^\pm$ is defined as the representative with smallest norm. That means $\bar{a} := (a \mod^\pm q)$ is the unique element with $-q/2 < \bar{a} \le q/2$ such that $\bar{a} \equiv a \mod q$. As q is an odd number in Kyber, we get that $a \mod^\pm q \in \{\frac{-q+1}{2}, \ldots, \frac{q-1}{2}\}$. Using this re-centred modulo operation, we define the function $\|\cdot\|_\infty$ on polynomials as:

$$\mathrm{p} = \sum_{i=1}^{\deg p} p_i \cdot x^i \longmapsto \|\mathrm{p}\|_\infty = \max_{i \in \{0, \ldots, \deg p\}} |p_i \mod^\pm q|$$

Analogously, for vectors $\mathbf{v} \in R_q^k$ we define:

$$\|\mathbf{v}\|_\infty = \max_{i \in \{1, \ldots, k\}} \|\mathrm{v}_i\|_\infty$$

Unfortunately with the re-centring one loses the absolute homogeneity, i.e., for a scalar s and vector $\mathbf{v}$ only $\|s \cdot \mathbf{v}\|_\infty \le |s| \cdot \|\mathbf{v}\|_\infty$ holds with an inequality instead of equality. For example consider the case $q = 3$, $s = 2$ and $\mathbf{v} = (2)$. We then have the strict inequality:

$$\begin{gathered} \|2 \cdot (2)\|_\infty = |2 \cdot 2 \mod^\pm 3| = 1 < \\ < 2 = |2| \cdot |2 \mod^\pm 3| = |2| \cdot \|(2)\|_\infty \end{gathered}$$

Therefore, the $\|\cdot\|_\infty$ function is not a norm, but a pseudonorm. It is positive definite and fulfils the triangle inequality. This is not explicitly mentioned in [11] and indeed poses a problem in the proof of the following correctness theorem.

B. Correctness of the Kyber Algorithms

A crypto system is correct, if it always returns the original message. However, since Kyber uses errors to mask the ciphertext, there is a chance that the error may be too large to decipher correctly. Thus, we need to consider a failure probability and can only state the δ-correctness. This is defined in the following:

Definition 2 (δ-correct PKE)**.** Let *key_gen*, *encrypt* and *decrypt* constitute a public key encryption scheme $\mathcal{A}$ where *key_gen* outputs a public key pk and a secret key sk. Let $\mathcal{M}$ be the space of all possible messages. Then the public key encryption scheme is δ-correct, if and only if:

$$\mathbb{E}[\max_{m \in \mathcal{M}} \mathbb{P}[decrypt(sk, encrypt(pk, m)) \neq m]] \leq \delta$$

where the expectation is taken over (pk, sk) generated by *key_gen*.

The intuition is that the probability of a decryption failure in the worst-case scenario over the message space and in a mean over the secret and public key pair should be bounded by a constant δ.

The δ-correctness of the PKE is a necessary requirement for the Fujisaki-Okamoto transform (verified by Unruh [37] in qrhl-tool). When connected with Unruh's formalization, the formalization presented in this paper results in a formal verification of Kyber's KEM with a verified indistinguishability under chosen ciphertext attack security property. A connection to Unruh's formalization was out of scope.

For the Kyber algorithms [11, Theorem 1], the δ-correctness theorem is proved in two steps:

1) An assumption sufficient for the correct decryption can be calculated deterministically. This is the main argument of the proof. The assumption is incorporated in the definition of δ.
2) The distributions in the compression errors are claimed to be uniformly random due to a reduction using the module-LWE problem.

The first, deterministic part is stated in the following theorem. Its formalization can be found in the Appendix X-A4.

Theorem IV.1. *Let* $\mathbf{A} \in R_q^{k \times k}$*,* $\mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{e_1} \in R_q^k$*,* $\mathrm{e}_2 \in R_q$ *and let the message* $\mathrm{m} \in R_q$ *with coefficients in* $\{0, 1\}$*. Define:*

- $\mathbf{t} = $ *key_gen* $\mathbf{A}\ \mathbf{s}\ \mathbf{e}$*, the output of the key generation*
- $(\mathbf{u}^*, \mathrm{v}^*) = $ *encrypt* $\mathbf{t}\ \mathbf{A}\ \mathbf{r}\ \mathbf{e1}\ \mathrm{e2}\ du\ dv\ \mathrm{m}$*, the output of the encryption*
- $\mathbf{c_u}$ *and* $\mathrm{c_v}$*, the compression errors of* $\mathbf{u}$ *and* v*, respectively*

If $\|\mathbf{e}^T\mathbf{r} + \mathrm{e}_2 + \mathrm{c_v} - \mathbf{s}^T\mathbf{e_1} - \mathbf{s}^T\mathbf{c_u}\|_\infty < \lceil q/4 \rfloor$*, then the decryption algorithm returns the original message* m*:*

$$decrypt\ \mathbf{u}^*\ \mathrm{v}^*\ \mathbf{s}\ du\ dv = \mathrm{m}$$

We have that Kyber is correct when assuming the inequality:

$$\|\mathbf{e}^T\mathbf{r} + \mathrm{e}_2 + \mathrm{c_v} - \mathbf{s}^T\mathbf{e_1} - \mathbf{s}^T\mathbf{c_u}\|_\infty < \lceil q/4 \rfloor \tag{1}$$

C. Modifying the Error Bound

Using Theorem IV.1 and the definition of δ-correctness, we deduce the following.

Corollary. *Let:*

$$\delta' = \mathbb{E}\left[\max_{\mathrm{m} \in \mathcal{M}} \mathbb{P}\left[\left[\begin{cases} \mathbf{e}, \mathbf{r}, \mathbf{e_1} \leftarrow \beta_\eta^k; \quad \mathrm{e}_2 \leftarrow \beta_\eta; \\ \mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e_1}; \\ \mathrm{v} = \mathbf{t}^T\mathbf{r} + \mathrm{e}_2 + \lceil \frac{q}{2} \rfloor \mathrm{m}; \\ \|\mathbf{e}^T\mathbf{r} + \mathrm{e}_2 + \mathrm{c_v} - \mathbf{s}^T\mathbf{e_1} - \\ \quad -\mathbf{s}^T\mathbf{c_u}\|_\infty \geq \lceil q/4 \rfloor \end{cases}\right]\right]\right] \tag{2}$$

where the expectation is taken over $((\mathbf{A}, \mathbf{t}), \mathbf{s})$ *generated by key_gen. Then Kyber is* δ'*-correct.*

Note that in this proposition, the δ' is not the same as in [11, Theorem 1]. Using the second proving step, [11] claims δ-correctness for:

$$\delta = \mathbb{P}\left[\begin{cases} \mathbf{s}, \mathbf{e}, \mathbf{r}, \mathbf{e_1} \leftarrow \beta_\eta^k; \quad \mathrm{e}_2 \leftarrow \beta_\eta; \\ \mathbf{c_u} \leftarrow \Psi_{du}^k, \mathrm{c_v} \leftarrow \Psi_{dv} \\ \|\mathbf{e}^T\mathbf{r} + \mathrm{e}_2 + \mathrm{c_v} - \mathbf{s}^T\mathbf{e_1} + \\ \quad -\mathbf{s}^T\mathbf{c_u}\|_\infty \geq \lceil q/4 \rfloor \end{cases}\right] \tag{3}$$

Here, Ψ_d is the distribution of the compression error of x to d bits for a uniformly generated $x \leftarrow R_q$.

The main difference between δ' and δ is that in δ' the values of $\mathbf{c_u}$ and $\mathrm{c_v}$ are calculated as the correct compression errors, whereas in δ they are the compression errors of uniformly random values $\mathbf{u}$ and v. The intuitive idea given in [11, Proof of Theorem 1] is that this change is negligible since its value can be bounded by the advantage against module-LWE problems. A detailed, formal proof is missing at this point.

Despite this idea making sense intuitively, we were unable to formalize this reduction. Indeed, we claim that this reduction is incorrect in our general framework. The reason is that the change from a module-LWE instance to a uniformly random instance loses all information about the secret key. However, in the definition of δ-correctness, we cannot omit the information about the secret key during the encryption since we need it for the decryption. Therefore, we cannot separate the module-LWE instance from $\mathbb{P}[decrypt(sk, encrypt(pk, m)) \neq m]$ in order to bound this value with the advantage against the module-LWE.

To substantiate the claim that this reduction to δ does not respect the inequality of Definition 2, we perform a comparative analysis of δ' (eq. (2)), δ (eq. (3)) and the actual correctness error:

$$\mathbb{E}[\max_{m \in \mathcal{M}} \mathbb{P}[decrypt(sk, encrypt(pk, m)) \neq m]] \tag{4}$$

In the following, the value δ is calculated using a Python script by Léo Ducas [12] that was also used for the evaluation in [11].

We showcase two comparisons: First, we calculate the exact values of δ, δ' and the correctness error for very small parameters. We set $n = 2$, $q = 17$, $k = \eta = d_v = 1$ and $d_u = 3$. The expectation, maximum and probabilities can be computed by considering all possible values. Using a simple Python script [23], the outcomes are the following:

$$\delta = 0.211$$
$$corr\_error = 0.223$$
$$\delta' = 0.267$$

The experiment shows, that for these small parameters, the inequality between the correctness error and δ is violated. This is a counterexample to the claimed proof in [11] since it should hold for any parameters sufficing Kyber's assumptions.

Second, we substantiate our claim also for (slightly) bigger parameter sets. In this experiment, we approximate δ' and the correctness error using Monte-Carlo sampling. Python scripts for the calculation of δ' and the correctness error can be found in [23]. For all the parameter sets that we tested, the inequality

$$\mathbb{E}[\max_{m \in \mathcal{M}} \mathbb{P}[decrypt(sk, encrypt(pk, m)) \neq m]] \leq \delta$$

is violated. Results are shown in Figure 1 (and Appendix X-B). As parameters, we consider n between 5 and 16 and choose q to be a prime with approximatively the same ratio to n as the original parameters for Kyber ($q/n = 3329/256 \approx 13$) and $q \equiv 1 \mod 4$. Furthermore, we set $k = \eta = d_v = 2$ and $d_u = 5$. In Figure 1, we see that the correctness error (green) consistently lies below our proposed δ' (blue), but violates the relation to the calculated δ (red) from [11].

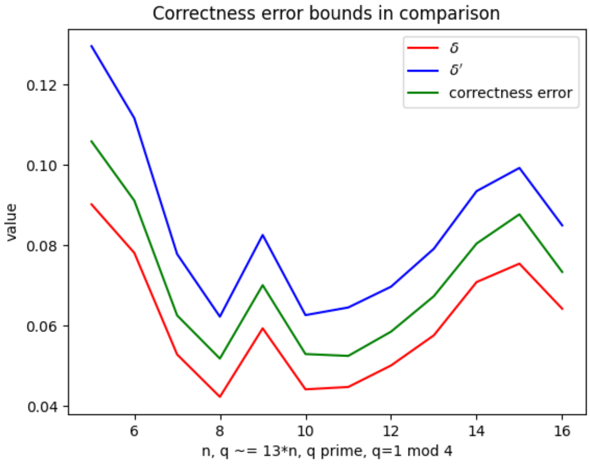


Figure 1: Comparison of absolute values of δ, δ' and the correctness error for small dimensional examples over variation on n

Private correspondence with an author of Kyber confirmed that this problem with δ was known. Indeed, they explained that the module-LWE reduction was more of a heuristic nature. With this heuristic module-LWE reduction, the error terms can be easily approximated using union bounds, as in [12]. However, since this proof was not formalized in detail, the dependency relations with the secret key may have been overlooked. The above calculations give a counter-example disproving this reduction via module-LWE for our general setting. It may be the case that additional assumptions for the Kyber parameters make the module-LWE reduction valid. An interesting future research question is to find suitable hypotheses to allow the module-LWE reduction or find a counter-example with the actual Kyber parameters.

Fortunately, our findings do not invalidate the correctness of the scheme itself since we could prove correctness with the bound δ'. Still, this issue may affect the level of security of Kyber. This leads us to two more important research questions:

1) Can we estimate/approximate δ' or the relation between δ and δ'?
2) Can we find another more easily calculable bound on the correctness error?

For this paper, our main focus was a foundational formalization of Kyber: indeed, we succeeded to show δ'-correctness.

D. Auxiliary Lemma

Before we can start the proof of Theorem IV.1, we need to show an auxiliary lemma on the estimation of the compression error.

Lemma IV.2. *Let x be an element of $\mathbb{Z}_q$ and $x' = decomp_d\ (comp_d\ x)$ its image under compression and decompression with $2^d < q$. Then we have:*

$$|x' - x \mod^{\pm} q| \leq \lceil q/2^{d+1} \rfloor$$

The proof of the auxiliary lemma can be found in Appendix X-C.

A non-trivial step in the formalization of the proof was to ensure that all calculations are conform with the residue classes modulo the polynomial $x^n + 1$. Indeed, in Isabelle the type casting is explicit, so one always has to channel through all type casts. Especially, one always has to show that the implications hold independently from the representative chosen from a residue class. In some cases, we also presume natural embeddings and isomorphisms to hold in pen-and-paper proofs which have to be stated explicitly in Isabelle (for example the *to_module* function mentioned in the previous section). Thus, formalizations are much more verbose.

E. Proof of Correctness

The formalization of the proof of Theorem IV.1 can be found in [22] (and the version with compression of the public key in [21]). One problem encountered during the formalization was that $\|\cdot\|_\infty$ is only a pseudo-norm (recall Section IV-A). This is not explicitly mentioned in [11] and indeed poses a problem in the proof which we will discuss in greater detail in the next section. In short: We cannot conclude a correct decryption in the last step of the correctness proof unless $q \equiv 1 \mod 4$.

The proof of Theorem IV.1 proceeds as follows. Given $\mathbf{A}$, $\mathbf{s}$, $\mathbf{r}$, $\mathbf{e}$, $\mathbf{e_1}$, e_2 and the message m, we calculate $\mathbf{t}$, $\mathbf{u}^*$ and v^* using the key generation and encryption algorithm. We define $\mathbf{u}'$ and v' to be the decompressed values of $\mathbf{u}^*$ and

v^*, respectively. With the compression errors $\mathbf{c_u}$ and $\mathrm{c_v}$, we get the equations:

$$\mathbf{u}' = \mathbf{A}^T\mathbf{r} + \mathbf{e_1} + \mathbf{c_u}$$
$$\mathrm{v}' = \mathbf{t}'^T\mathbf{r} + \mathrm{e}_2 + \lceil q/2 \rfloor \cdot \mathrm{m} + \mathrm{c_v}$$

This leads to the calculation in the decryption:

$$\mathrm{v}' - \mathbf{s}^T\mathbf{u}' = \mathbf{e}^T\mathbf{r} + \mathrm{e}_2 + \mathrm{c_v} - \mathbf{s}^T\mathbf{e_1} - \mathbf{s}^T\mathbf{c_u} + \lceil q/2 \rfloor \cdot \mathrm{m}$$

We accumulate all error terms in a new variable w:

$$\mathrm{w} := \mathbf{e}^T\mathbf{r} + \mathrm{e}_2 + \mathrm{c_v} - \mathbf{s}^T\mathbf{e_1} - \mathbf{s}^T\mathbf{c_u}$$

and get $\|\mathrm{w}\|_\infty < \lceil q/4 \rfloor$ from the assumptions of Theorem IV.1.

Now, we need to show that $\mathrm{m}' := \mathit{decrypt}(\mathbf{u}^*, \mathrm{v}^*, \mathbf{s})$ is indeed the original message m. We consider the value of $\mathrm{v}' - \mathbf{s}^T\mathbf{u}'$, its compression with $d = 1$, namely m', and the decompressed value $\mathrm{decomp}_1\ \mathrm{m}'$. Since the compression depth is 1, we get $\mathrm{m}' \in \{0, 1\}$. Thus:

$$\mathrm{decomp}_1\ \mathrm{m}' = \lceil q/2 \cdot \mathrm{m}' \rfloor = \lceil q/2 \rfloor \cdot \mathrm{m}'$$

Using Lemma IV.2, it follows that:

$$\begin{aligned}&\|\mathrm{w} + \lceil q/2 \rfloor(\mathrm{m} - \mathrm{m}')\|_\infty \\ =&\|\mathrm{v}' - \mathbf{s}^T\mathbf{u}' - \mathit{decomp}_1\ (\mathit{comp}_1\ (\mathrm{v}' - \mathbf{s}^T\mathbf{u}'))\|_\infty \\ \leq&\lceil q/4 \rfloor\end{aligned}$$

Using the triangle inequality on $\|\cdot\|_\infty$, we calculate

$$\begin{aligned}&\|\lceil q/2 \rfloor(\mathrm{m} - \mathrm{m}')\|_\infty = \|\mathrm{w} + \lceil q/2 \rfloor(\mathrm{m} - \mathrm{m}') - \mathrm{w}\|_\infty \\ \leq&\|\mathrm{w} + \lceil q/2 \rfloor(\mathrm{m} - \mathrm{m}')\|_\infty + \|\mathrm{w}\|_\infty \\ <&\lceil q/4 \rfloor + \lceil q/4 \rfloor = 2\lceil q/4 \rfloor\end{aligned}$$

It remains to show that we can indeed deduce $\mathrm{m} = \mathrm{m}'$ which concludes the proof of Theorems IV.1. According to the last step of [11, Proof Thm 1], this follows directly for any odd q. However, therein lies a hidden problem. [11, Proof Thm 1] makes use of the homogeneity of $\|\cdot\|_\infty$. Since $\|\cdot\|_\infty$ is only a pseudo-norm and not a norm, we needed to find an alternative proof in the formalization. Interestingly enough, in the case of $q \equiv 3 \mod 4$, we cannot conclude the proof. In the next section, we discuss why we can only deduce this step under the assumption that $q \equiv 1 \mod 4$ and give a counterexample for the case $q \equiv 3 \mod 4$.

F. Additional Assumption $q \equiv 1 \mod 4$

The following remains to be shown for the proof of Theorem IV.1: Given the inequality

$$\|\lceil q/2 \rfloor \cdot (\mathrm{m} - \mathrm{m}')\|_\infty < 2 \cdot \lceil q/4 \rfloor$$

we need to deduce that indeed $\mathrm{m} = \mathrm{m}'$.

We prove this statement by contradiction. Assume that m is not equal m', i.e., there exists a coefficient of $\mathrm{m} - \mathrm{m}'$ that is different from zero. Since m and m' are polynomials with coefficients in $\{0, 1\}$, a non-zero coefficient can either be 1 or -1. Then we get

$$\|\lceil q/2 \rfloor \cdot (\mathrm{m} - \mathrm{m}')\|_\infty = |\lceil q/2 \rfloor \cdot (\pm 1) \mod^{\pm} q| = \dots$$

Since we cannot use the homogeneity of $\|\cdot\|_\infty$ to pull out the absolute value of ± 1, we need to find a different proof. We break down the formula to find the remaining problems. All primes q greater than two are odd. Thus we have $\lceil q/2 \rfloor = (q + 1)/2$. We continue our calculation:

$$\dots = \left|\frac{q+1}{2} \mod^{\pm} q\right| = \left|\frac{-q+1}{2}\right| = \frac{q-1}{2} = 2 \cdot \frac{q-1}{4} = \dots$$

since the $\mathrm{mod}^{\pm}$ operation reduces $\frac{q+1}{2}$ to the representative $\frac{-q+1}{2}$. Now we need to relate $\frac{q-1}{4}$ to $\lceil q/4 \rfloor$. We have two cases:

Case 1: For $q \equiv 1 \mod 4$ we indeed get the equality $\frac{q-1}{4} = \lceil q/4 \rfloor$ that we need. In this case we have

$$\|\lceil q/2 \rfloor \cdot (\mathrm{m} - \mathrm{m}')\|_\infty = 2 \cdot \lceil q/4 \rfloor$$

which is a contradiction to our assumption. In this case, the proof of Theorems IV.1 is completed.

Case 2: For $q \equiv 3 \mod 4$ we get the strict inequality $\frac{q-1}{4} < \frac{q+1}{4} = \lceil q/4 \rfloor$ resulting in

$$\|\lceil q/2 \rfloor \cdot (\mathrm{m} - \mathrm{m}')\|_\infty < 2 \cdot \lceil q/4 \rfloor$$

which is no contradiction to the assumption. Indeed in this case we cannot deduce $\mathrm{m} = \mathrm{m}'$, since it is possible that a coefficient of $\mathrm{m} - \mathrm{m}'$ is non-zero.

Example IV.1. Consider this short example: Let $q = 7$ ($\equiv 3 \mod 4$, thus we are in case 2), $\mathrm{m} = 0$ and $\mathrm{m}' = 1$. In this case, the inequality of the assumption holds

$$\|\lceil q/2 \rfloor \cdot (\mathrm{m} - \mathrm{m}')\|_\infty = 3 < 4 = 2 \cdot \lceil q/4 \rfloor$$

but $\mathrm{m} \neq \mathrm{m}'$. This is a counterexample for the correctness of the proof of Theorem IV.1 in the case $q \equiv 3 \mod 4$.

In conclusion, Theorem IV.1 only holds if the modulus q fulfils the assumption $q \equiv 1 \mod 4$. □

In the specification of Kyber, concrete values for the parameters of the system are given (see Section II-B). For example in the recent version of Kyber [4], [5], the modulus q is chosen to be 3329, whereas in early versions [6], [11], the modulus was chosen as 7681. Considering possible changes to these variables (for different versions or security levels), it is important to enable the verified proof to cover all possible cases. Therefore, the implementation of the formalization was chosen to be as adaptive and flexible as possible. This resulted in the discovery of the additional assumption $q \equiv 1 \mod 4$.

Indeed, the modulus q is chosen according to a much more rigid scheme: In order to implement the multiplication to compute faster, the Number Theoretic Transform (NTT) is used. In the case of Kyber, the NTT is computed on $R_q = \mathbb{Z}_q[x]/(x^n + 1)$. The requirement for NTT on the modulus q is:

$$q \equiv 1 \mod n$$

For $n = 256$ and $q = 7681$ we have $7681 = 30 \cdot 256 + 1$, whereas for $q = 3329$ we get $3329 = 13 \cdot 256 + 1$. Since n is a power of 2^2, we can automatically infer the property $q \equiv 1 \mod 4$.

The NTT is analysed in more detail in Appendix X-D. A formalization for the NTT in the case of Kyber was included in this project. There is also a formalization of the NTT for the third round Kyber by Hwang et al. [18] in the low-level tool CryptoLine.

V. Game-Based Cryptography

An important cryptographic property of public key encryption schemes is IND-CPA security. This attack describes a game where an adversary tries to gain information about self-chosen plaintexts.

More formally, the IND-CPA game for a PKE (given by the key generation, encryption and decryption algorithms) is defined as follows.

Definition 3 (IND-CPA game)**.** Two parties, the challenger and the adversary, play the following game.

1) The challenger generates a public and secret key pair using the key generation algorithm and publishes the public key.
2) The adversary sends the challenger two messages m_0 and m_1 with the same length.
3) The challenger chooses uniformly at random a bit b. He encrypts the message m_b with the encryption algorithm and sends the ciphertext to the adversary.
4) The adversary returns a guess b' which of the two given messages m_0 and m_1 the challenger has encrypted. He wins if $b = b'$.

The advantage $Adv^{IND-CPA}$ of the adversary $\mathcal{A}$ is defined as $Adv^{IND-CPA}(\mathcal{A}) = |\mathbb{P}[b' = b] - \frac{1}{2}|$. A PKE scheme is IND-CPA secure if and only if the advantage of the adversary is negligible, that means sufficiently small.

Figure 2 depicts the IND-CPA game.

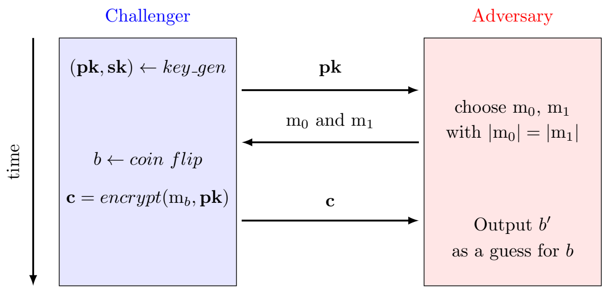

Challenger
Adversary
time
$(\mathbf{pk}, \mathbf{sk}) \leftarrow key\_gen$
$\mathbf{pk}$
m_0 and m_1
choose m_0, m_1
with $|\mathrm{m}_0| = |\mathrm{m}_1|$
$b \leftarrow coin\ flip$
$\mathbf{c} = encrypt(\mathrm{m}_b, \mathbf{pk})$
$\mathbf{c}$
Output b'
as a guess for b

Figure 2: A diagram of the IND-CPA game.

The formalization of the IND-CPA game was taken from the CryptHOL Tutorial [27]. The flexible formalization in an Isabelle locale allows the user to instantiate this concept in any context fulfilling the properties of the locale. In this way, the IND-CPA game definition could easily be applied to the case of Kyber by instantiating with the Kyber algorithms for key generation, encryption and decryption.

We can also state the module-LWE from Definition 1 in game form.

Definition 4 (module-LWE game)**.** Two parties, called the challenger and the (module-LWE) adversary, play the following game.

1) The challenger chooses $\mathbf{A_0} \in R_q^{m \times k}$ uniformly at random, $\mathbf{s}$ according to β_η^k and $\mathbf{e}$ according to β_η^m. He then computes $\mathbf{t_0} = \mathbf{A_0}\mathbf{s} + \mathbf{e}$. ($(\mathbf{A_0}, \mathbf{t_0})$ is an instance of the module-LWE problem.)
2) The challenger chooses $\mathbf{A_1} \in R_q^{m \times k}$ and $\mathbf{t_1} \in R_q^m$ uniformly at random. ($(\mathbf{A_1}, \mathbf{t_1})$ is a random instance.)
3) The challenger chooses a random bit b and sends the adversary the value of $(\mathbf{A_b}, \mathbf{t_b})$.
4) The adversary returns a guess b' whether the tuple $(\mathbf{A_b}, \mathbf{t_b})$ was generated as a module-LWE instance or is uniformly random. He wins, if his guess is correct.

The advantage Adv_m^{mLWE} of the module-LWE adversary $\mathcal{A}$ is defined as

$$Adv_m^{mLWE}(\mathcal{A}) = |\mathbb{P}[b' = 0 \land b = 0] - \mathbb{P}[b' = 0 \land b = 1]|$$

The module-LWE hardness assumption states that the advantage of an adversary in the module-LWE game is negligible.

Figure 3 depicts the module-LWE problem in game form.

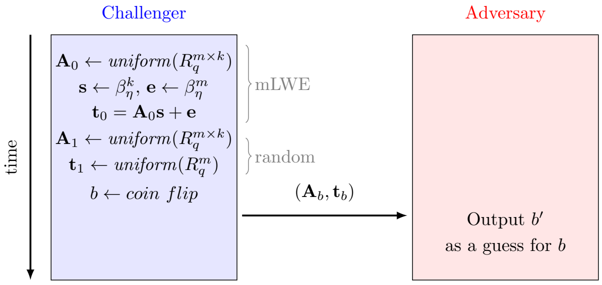

Challenger
Adversary
time
$\mathbf{A}_0 \leftarrow uniform(R_q^{m \times k})$
$\mathbf{s} \leftarrow \beta_\eta^k$, $\mathbf{e} \leftarrow \beta_\eta^m$
$\mathbf{t}_0 = \mathbf{A}_0\mathbf{s} + \mathbf{e}$
mLWE
$\mathbf{A}_1 \leftarrow uniform(R_q^{m \times k})$
$\mathbf{t}_1 \leftarrow uniform(R_q^m)$
random
$b \leftarrow coin\ flip$
$(\mathbf{A}_b, \mathbf{t}_b)$
Output b'
as a guess for b

Figure 3: A diagram of the module-LWE game.

In the proof of the IND-CPA security property for Kyber, the advantage of a module-LWE adversary is used twice, but with different dimensions m. The key generation corresponds to a module-LWE with $m = k$ such that $\mathbf{A}$ is a quadratic matrix. However, in the encryption, the matrix $\mathbf{A}$ is extended by the vector $\mathbf{t}$, resulting in a $(k+1) \times k$ matrix. This corresponds to the module-LWE with $m = k + 1$.

The module-LWE was again formalized in an Isabelle locale in order to allow for two separate instantiations (once with $m = k$ and once with $m = k+1$). However, the instantiations needed an additional twist. Since the vector type in Isabelle has a fixed dimension implemented as a finite type (in our case type $'k$ of cardinality k), it is more difficult to work over vectors whose dimension is a function over k. In our case, this could be solved using the option type. The option type $'k$ *option* embeds elements a of type $'k$ as *Some* a and adds the element *None*. Thus $'k$ *option* has exactly $k+1$ elements. This solves our problem.

VI. Using CryptHOL in Isabelle

CryptHOL [26] is a library for game-based security proofs in cryptography. It is based on the extensive libraries for probability theory in Isabelle. Its main contributions are sub-probability mass function as the type class *spmf* and generative probabilistic values as the type class *gpv*. We give a short intuitive understanding of these type classes.

A. Sub-probability Mass Functions

The *spmf* type class is a superclass of probability mass functions. We consider a finite set S. A probability mass function $f : S \longmapsto [0,1]$ is the probability distribution of a discrete random variable X, i.e., $f(x) = \mathbb{P}[X = x]$ such that the weight equals one:

$$\sum_{x \in S} f(x) = 1$$

For sub-probability mass functions, we allow the weight to be less than one:

$$\sum_{x \in S} f(x) \leq 1$$

A sub-probability mass function is called lossless, if it has weight equal to one. Indeed, in our setting we need to model the probability that a security game is compromised by intentional malicious input and may not terminate. For example in the IND-CPA game, the adversary can intentionally input two messages of different length and thus gain information about the ciphertext or simply not answer at all.

B. Generative Probabilistic Values

To model cryptographic primitives such as hash functions, we need a method to generate and store random values. This idea is developed in the *gpv* type class which describes probabilistic algorithms. The type class *gpv* depends on three input types: the type of the algorithm, the input state type and the output state type.

When running a *gpv*, we connect it with a random oracle (that models for example a hash function) and hand through the current state. Whenever we query the oracle, we generate a new state. It needs to be included in the input for the next call to the oracle using a *gpv*.

The Kyber public key encryption does not use hash functions. Thus we could model the security proof with sub-probability mass functions only. However, to stay consistent with the CryptHOL library, we generalized the formalization of the security proof to use generative probabilistic values whenever we query the adversary or the encryption algorithm. The proofs do not get significantly harder and the automation can handle this generalization step most of the time.

C. Using Monads for Describing Probabilistic Algorithms

Functional programming hands us tools to easily define probabilistic algorithms and distributions. The concept of choice is the Giry-monad. Monads are a concept from category theory applied to functional programming. We give a short introduction to monads in general and the Giry-monad in particular. More about monads can be found in [34] and the introduction of monads into functional programming in [30]. A good introduction to the Giry-monad in the context of Isabelle is given in [13].

Monads give a pattern to design type classes. They consist of a type constructor M and two operations:

- *return*: receives a value $\mathbf{A}$ and hands back a monadic value $M\ a$
- *bind*: receives a monadic value Ma and a function $f : a \longrightarrow M\ b$ and returns the application of f to the unwrapped value $\mathbf{A}$, yielding an element $M\ b$

Monads need to fulfil three laws: the left and right identities and associativity. Let us look at a short example.

Example VI.1. The $option$ type class is a monad. As described at the end of Section V, a type $'a\ option$ takes the values $Some\ a$ or $None$. In this case, the $option$ monad is defined over the type $'a$. The *return* function takes an element a of type $'a$ and returns an element $Some\ a$ of type $'a\ option$. The *bind* function on a function f is defined by:

$$bind\ None\ f = None$$
$$bind\ (Some\ a)\ f = f(a)$$

Another notation for the *bind* function is:

$$bind\ a\ f \equiv a \gg\!= f$$

Another example is the Giry-monad. It assigns to each measurable space the space of probability measures over it (see [35]).

Example VI.2. The type class of probability mass functions *pmf* for discrete distributions is a monad, called the Giry-monad. The *return* function for an element $\mathbf{A}$ is defined as the Dirac measure on a. The *bind* function on an probability mass function p_X using a function f is defined as:

$$(bind\ p_X\ \ f)(y) = \sum_x p_X(f(x)(y))$$

Thus, the Giry-monad can model successive execution of random experiments and probabilistic algorithms using the *bind* and *return* functions.

Both the type class *spmf* and *gpv* are monads with respective *return* and *bind* functions. This gives us a tool to model probabilistic algorithms in Isabelle.

VII. IND-CPA Security Proof for Kyber

Since round two of the NIST standardization process [5], the compression of the public key in Kyber has been omitted. The reason was that otherwise the IND-CPA security proof [11, Theorem 2] does not hold. The problem lies in the second reduction step where the decompression of the compression of the public key is not distributed uniformly at random any more. This entails that we cannot apply the reduction from the module-LWE. The security of Kyber without compression under IND-CPA is stated in the following theorem. Its formalization can be found in [22]

Theorem VII.1. *Given any adversary $\mathcal{A}$ to the IND-CPA game of Kyber and assuming that $\mathcal{A}$ is lossless, the advantage of $\mathcal{A}$ in the IND-CPA game can be bounded by twice the advantage in the module-LWE game.*

Loosely speaking: the public key encryption scheme Kyber without compression of the public key is IND-CPA secure against the module-LWE hardness assumption. The formalization in Isabelle can be found in Appendix X-A5.

Proof. Let Adv^{Kyber} be the advantage in the IND-CPA game instantiated with the Kyber algorithms key_gen, $encrypt$ and $decrypt$. Let f_1 be the reduction function from $\mathcal{A}$ to the first module-LWE instance and f_2 the reduction function from $\mathcal{A}$ to the second module-LWE instance. Then the exact formula of the theorem above reads:

$$Adv^{Kyber}(\mathcal{A}) \leq Adv_k^{mLWE}(f_1(\mathcal{A})) + Adv_{k+1}^{mLWE}(f_2(\mathcal{A})) \tag{5}$$

Note that in the formalization we state the reduction functions for the adversary precisely. They need to have a polynomial running time. Since a formal framework for analysing the running time is out of scope for this project, we assume the running time hypothesis to be correct. The reason is that the reduction functions use only one call to the given adversary and (for f_1) one to the Kyber encryption algorithm. Otherwise, the functions are non-recursive, polynomial time probabilistic algorithms.

The proof of equation (5) proceeds in three steps (also called game-hops).

1) Reduction of key generation from the first module-LWE instance with $m = k$
2) Reduction of encryption from the second module-LWE instance with $m = k + 1$
3) Interpretation of the rest as a coin flip

In every game-hop, we define an intermediate game and analyse the difference in the advantage. The initial game $game_0$ is exactly the IND-CPA game. That implies:

$$\mathbb{P}[b = b'] = \mathbb{P}[game_0 = true]$$

The first intermediate game $game_1$ is defined by the following steps:

1) The challenger generates a public key $(\mathbf{A}, \mathbf{t})$ uniformly at random and publishes the public key.
2) The adversary sends the challenger two messages m_0 and m_1 with the same length.
3) The challenger chooses a bit b uniformly at random. He encrypts the message m_b with the encryption algorithm and sends the ciphertext to the adversary.
4) The adversary returns a guess b' for which of the two given messages m_0 and m_1 the challenger has encrypted. He wins if $b = b'$.

Figure 4 illustrates $game_1$. The change to the initial game $game_0$ (marked in green) is in the first step where the public and secret key pair is now generated uniformly at random instead of being created by the key generation algorithm.

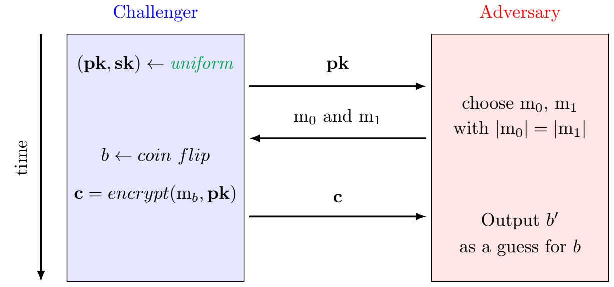

Challenger
Adversary
time
$(\mathbf{pk}, \mathbf{sk}) \leftarrow uniform$
pk
$b \leftarrow coin\ flip$
$\mathbf{c} = encrypt(\mathrm{m}_b, \mathbf{pk})$
m_0 and m_1
c
choose m_0, m_1 with $|\mathrm{m}_0| = |\mathrm{m}_1|$
Output b' as a guess for b

Figure 4: A diagram of $game_1$.

The key generation algorithm creates a module-LWE instance. Distinguishing a module-LWE instance from a uniformly random instance is exactly the module-LWE game. Hence, for a suitable reduction function f_1 have:

$$|\mathbb{P}[game_0 = true] - \mathbb{P}[game_1 = true]| = Adv_k^{mLWE}(f_1(\mathcal{A}))$$

The second intermediate game $game_2$ is defined by the following steps:

1) The challenger generates a public key $(\mathbf{A}, \mathbf{t})$ uniformly at random and publishes the public key.
2) The adversary sends the challenger two messages m_0 and m_1 with the same length.
3) The challenger chooses a bit b uniformly at random. He chooses a ciphertext uniformly at random from $R_q^k \times R_q$ and sends the ciphertext to the adversary.
4) The adversary returns a guess b' for b. He wins if $b = b'$.

Figure 5 illustrates $game_2$. The change to $game_1$ (marked in green) is that the ciphertext is not generated by the encryption but chosen uniformly at random.

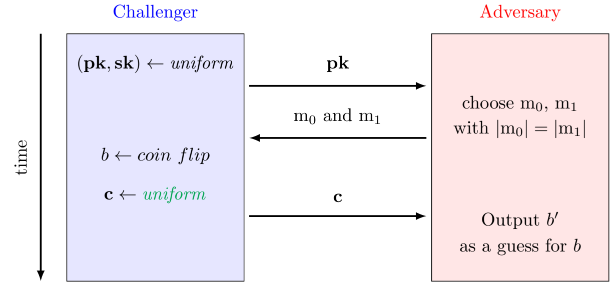

Challenger
Adversary
time
$(\mathbf{pk}, \mathbf{sk}) \leftarrow uniform$
pk
$b \leftarrow coin\ flip$
$\mathbf{c} \leftarrow uniform$
m_0 and m_1
c
choose m_0, m_1 with $|\mathrm{m}_0| = |\mathrm{m}_1|$
Output b' as a guess for b

Figure 5: A diagram of $game_2$.

In the encryption, the reduction to the module-LWE is not as straightforward as for the key generation. This is caused by the addition of the message m to the module-LWE instance. Indeed, in the formalization, we need to make two separate steps.

First, we show that the probability of distinguishing an instance of the form

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{t} \end{pmatrix} \mathbf{r} + \begin{pmatrix} \mathbf{e_1} \\ \mathrm{e}_2 \end{pmatrix}$$

and a uniformly random instance $(\mathbf{u} \ \ \mathrm{v}')^T$ is exactly the module-LWE advantage for $m = k + 1$. Note that it is important to look at $(k+1)$-dimensional vectors instead of splitting the instance in k- and 1-dimensional parts because $\mathbf{r}$ is chosen to be the same for the multiplication with both $\mathbf{A}$ and $\mathbf{t}$. This is also the reason, why we cannot split the variance for the centred binomial distribution into η_1 and η_2, since e_1 and e_2 together form the error vector of the module-LWE instance, thus needing the same distribution.

Second, we need to show that $\mathrm{v}' + \lceil q/2 \rfloor \cdot \mathrm{m}$ is also distributed uniformly. That is, we cannot distinguish between the probabilities of the value $\mathrm{v}' + \lceil q/2 \rfloor \cdot \mathrm{m}$ for a uniformly random v' and a uniformly random v. Since we are working over a finite field and v' and m are independent, we can show this property using the law of total probability.

For a suitable reduction function f_2, we deduce:

$$\Big|\mathbb{P}[game_1 = true] - \mathbb{P}[game_2 = true]\Big| = Adv_{k+1}^{mLWE}(f_2(\mathcal{A}))$$

In the last step, we have a closer look at $game_2$. Since the ciphertext sent to the adversary is now independent from the chosen message, the guess of the adversary is a coin flip. Thus the probability of guessing correctly is exactly $1/2$. We get

$$\mathbb{P}[game_2 = true] = 1/2$$

Finally, we can put together all the previous steps.

$$Adv^{Kyber}(\mathcal{A}) = \left|\mathbb{P}[b = b'] - \frac{1}{2}\right| = \left|\mathbb{P}[game_0 = true] - \frac{1}{2}\right|$$

This equality is inferred from the definition of the adversary for the IND-CPA game for Kyber. The $game_0$ is the initial IND-CPA game. We continue by applying the triangle inequality.

$$\begin{aligned}&\left|\mathbb{P}[game_0 = true] - \frac{1}{2}\right| \\ \leq &\Big|\mathbb{P}[game_0 = true] - \mathbb{P}[game_1 = true]\Big| \\ &+ \left|\mathbb{P}[game_1 = true] - \frac{1}{2}\right| \\ = &\ Adv_k^{mLWE}(f_1(\mathcal{A})) + \left|\mathbb{P}[game_1 = true] - \frac{1}{2}\right|\end{aligned}$$

The last equality is deduced from the reduction of $game_0$ to $game_1$ as a module-LWE instance. We proceed by applying the triangle inequality again on the second part.

$$\begin{aligned}&\left|\mathbb{P}[game_1 = true] - \frac{1}{2}\right| \\ \leq &\Big|\mathbb{P}[game_1 = true] - \mathbb{P}[game_2 = true]\Big| \\ &+ \left|\mathbb{P}[game_2 = true] - \frac{1}{2}\right| \\ = &\ Adv_{k+1}^{mLWE}(f_2(\mathcal{A})) + \left|\mathbb{P}[game_2 = true] - \frac{1}{2}\right|\end{aligned}$$

Here, the last equality is deduced from the reduction of $game_1$ to $game_2$ as a module-LWE instance with $m = k+1$. Finally, we have $\left|\mathbb{P}[game_2 = true] - \frac{1}{2}\right| = 0$ as $game_2$ behaves like a coin flip. In total, the claim is proven as we have shown the formula:

$$Adv^{Kyber}(\mathcal{A}) \leq Adv_k^{mLWE}(f_1(\mathcal{A})) + Adv_{k+1}^{mLWE}(f_2(\mathcal{A}))$$

□

During the formalization process, it became clear that this proof does not work for the first version of Kyber as remarked by the authors of Kyber [11, Sec. Security of the real scheme]. The proof for the current scheme could be formalized analogously to the pen-and-paper proof. The most time-consuming parts were getting familiar with the CryptHOL library environment and working out the details of the pen-and-paper proof which was extremely short.

CryptHOL works with sub-probability mass functions and generative probabilistic values and supplies a huge library of fundamental lemmas. Since the example game-based proof of the CryptHOL Tutorial [27] is based mainly on the automation, understanding the formal proof and rewriting steps is not straightforward. However, once the necessary lemmas are located and added to the automation, the automatic proof finder can solve most rewriting steps.

Some steps where the automation fails are for example when commutativity laws need to be applied in both directions. Then the simplifier runs in loops and cannot terminate. Making smaller proof steps or explicitly initializing the commutativity laws solves these issues.

VIII. Implementation Details

The implementation in Isabelle comprises about $6.7k$ lines of code. The proportions on the topics is depicted in Figure 6. Since many concepts from algebra, analysis, probability theory and cryptographic primitives could be reused, the authors could focus solely on the formalization of Kyber. Furthermore, the automation greatly helped shortening the proofs.

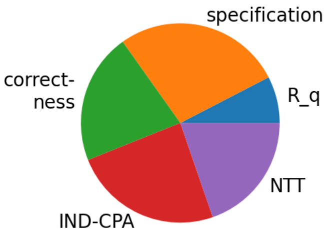


Figure 6: Distribution of lines of code on different topics

Due to the various dependencies and invocations of the library, loading the formalization theories might take some time, especially when the required theories on analysis and probability need to be built for the first time.

IX. Conclusion

In this presentation, we described the formalization of key-generation, encryption and decryption algorithms of CRYSTAL-KYBER's public key encryption scheme.

During the formalization of the δ-correctness proof two problems were uncovered: One could be solved by modifying the value of δ, the other by adding the assumption $q \equiv 1 \mod 4$. Under these conditions, the δ'-correctness could be verified. Differences between the original proof and the formalization were discussed and counterexamples for failing proof-steps were given. The additional assumption $q \equiv 1 \mod 4$ is already fulfilled by necessary properties for the number theoretic transform. Therefore, the correctness of Kyber itself is not compromised but minimal changes to the error bound δ are needed. However, the authors of Kyber acknowledged the need for an alternative bound in private communication. Moreover, a verification of the IND-CPA security proof for Kyber was presented.

Building on these results, the Fujisaki-Okamoto transform can be applied to the current algorithm formalization to obtain a verified key encapsulation mechanism that is secure against the indistinguishability under chosen ciphertext attack (IND-CCA). However, our proposed δ' cannot be approximated as easily as the original δ. Finding a calculable bound or an approximation on δ' remains an important question. Another very interesting aspect is to formalize the hardness results of the module-LWE that Kyber is building on.

Acknowledgment

Many thanks to Tobias Nipkow and Manuel Eberl for their continuous support. Thanks also to Manuel Barbosa, Peter Schwabe and Andreas Hülsing for their insightful discussions. The author gratefully acknowledges the financial support of this work by the research training group ConVeY funded by the German Research Foundation under grant GRK 2428.

References

[1] M. R. Albrecht, S. Celi, B. Dowling, and D. Jones. Practically-exploitable Cryptographic Vulnerabilities in Matrix. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2023.

[2] J. B. Almeida, M. Barbosa, G. Barthe, B. Grégoire, V. Laporte, J.-C. Léchenet, T. Oliveira, H. Pacheco, M. Quaresma, P. Schwabe, A. Séré, and P.-Y. Strub. Formally verifying Kyber Episode IV: Implementation Correctness. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, page 164–193, June 2023.

[3] T. Ammer and K. Kreuzer. Number Theoretic Transform. *Archive of Formal Proofs*, August 2022. https://isa-afp.org/entries/Number_Theoretic_Transform.html, Formal proof development.

[4] R. M. Avanzi, J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.0). 01/10/2020.

[5] R. M. Avanzi, J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 2.0). 30/03/2019.

[6] R. M. Avanzi, J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation. 30/11/2017.

[7] C. Ballarin. Locales and Locale Expressions in Isabelle/Isar. In S. Berardi, M. Coppo, and F. Damiani, editors, *Types for Proofs and Programs*, pages 34–50, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[8] M. Barbosa, G. Barthe, X. Fan, B. Grégoire, S.-H. Hung, J. Katz, P.-Y. Strub, X. Wu, and L. Zhou. EasyPQC: Verifying Post-Quantum Cryptography. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 2564–2586, New York, NY, USA, 2021. Association for Computing Machinery.

[9] M. Barbosa, A. Hülsing, M. Meijers, and P. Schwabe. Formal Verification of Post-Quantum Cryptography. https://csrc.nist.gov/CSRC/media/Presentations/formal-verifcation-of-post-quantum-cryptography/images-media/session-2-meijers-formal-verification-pqc.pdf, accessed: 2022-09-06.

[10] M. Barbosa, A. Hülsing, M. Meijers, and P. Schwabe. Formal Verification of Post-Quantum Cryptography. https://csrc.nist.gov/CSRC/media/Events/third-pqc-standardization-conference/documents/accepted-papers/meijers-formal-verification-pqc2021.pdf, accessed: 2022-09-06.

[11] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS — Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367, 2018.

[12] L. Ducas and J. Schanck. pq-crystals/security-estimates, 2021. https://github.com/pq-crystals/security-estimates/tree/master, accessed: 2023-08-09.

[13] M. Eberl, J. Hölzl, and T. Nipkow. A Verified Compiler for Probability Density Functions. In J. Vitek, editor, *Programming Languages and Systems*, pages 80–104, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[14] GitHub. EasyCrypt, 2022. https://github.com/EasyCrypt/easycrypt, accessed: 2022-07-26.

[15] F. Haftmann and M. Wenzel. Constructive Type Classes in Isabelle. In T. Altenkirch and C. McBride, editors, *Types for Proofs and Programs*, pages 160–174, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[16] J. Harrison. A HOL Theory of Euclidean space. In J. Hurd and T. Melham, editors, *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005*, volume 3603 of *LNCS*, pages 114–129, Oxford, UK, 2005. Springer-Verlag.

[17] B. Huffman and O. Kunčar. Lifting and Transfer: A Modular Design for Quotients in Isabelle/HOL. In *Certified Programs and Proofs*, pages 131–146. Springer International Publishing, 2013.

[18] V. Hwang, J. Liu, G. Seiler, X. Shi, M.-H. Tsai, B.-Y. Wang, and B.-Y. Yang. Verified NTT Multiplications for NISTPQC KEM Lattice Finalists: Kyber, SABER, and NTRU. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):718–750, August 2022.

[19] C. Kaliszyk and C. Urban. Quotients Revisited for Isabelle/HOL. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 1639–1644, New York, NY, USA, 2011. Association for Computing Machinery.

[20] J. Klemsa. *Fast and Error-Free Negacyclic Integer Convolution Using Extended Fourier Transform*, page 282–300. Springer International Publishing, 2021.

[21] K. Kreuzer. CRYSTALS-Kyber. *Archive of Formal Proofs*, September 2022. https://isa-afp.org/entries/CRYSTALS-Kyber.html, Formal proof development.

[22] K. Kreuzer. CRYSTALS-Kyber Security. *Archive of Formal Proofs*, December 2023. https://isa-afp.org/entries/CRYSTALS-Kyber_Security.html, Formal proof development.

[23] K. Kreuzer. Kyber error bound, 2023. https://github.com/ThikaXer/Kyber_error_bound, accessed: 2023-12-19.

[24] K. Kreuzer. Verification of the (1–δ)-Correctness Proof of CRYSTALS-KYBER with Number Theoretic Transform. Cryptology ePrint Archive, Paper 2023/027, 2023. https://eprint.iacr.org/2023/027.

[25] A. Langlois and D. Stehlé. Worst-Case to Average-Case Reductions for Module Lattices. *Des. Codes Cryptogr.*, 75(3):565–599, June 2015.

[26] A. Lochbihler. CryptHOL. *Archive of Formal Proofs*, May 2017. https://isa-afp.org/entries/CryptHOL.html, Formal proof development.

[27] A. Lochbihler and S. R. Sefidgar. A tutorial introduction to CryptHOL. Cryptology ePrint Archive, Paper 2018/941, 2018. https://eprint.iacr.org/2018/941.

[28] P. Longa and M. Naehrig. Speeding up the Number Theoretic Transform for Faster Ideal Lattice-Based Cryptography. In *Lecture Notes in Computer Science*, pages 124–139. Springer International Publishing, 10 2016.

[29] R. Maleckas, K. G. Paterson, and M. R. Albrecht. Practically-exploitable Vulnerabilities in the Jitsi Video Conferencing System. Cryptology ePrint Archive, Paper 2023/1118, 2023. https://eprint.iacr.org/2023/1118.

[30] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, July 1991.

[31] T. Nipkow and G. Klein. *Concrete Semantics with Isabelle/HOL*. Springer, 2014. http://concrete-semantics.org.

[32] T. Nipkow, L. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[33] NIST. NIST - Post-Quantum Cryptography, 2023. https://csrc.nist.gov/projects/post-quantum-cryptography, accessed: 2023-05-12.

[34] nLab authors. monad. https://ncatlab.org/nlab/show/monad, Nov. 2022. Revision 97.

[35] nLab authors. monads of probability, measures, and valuations. https://ncatlab.org/nlab/show/monads%20of%20probability%2C%20measures%2C%20and%20valuations, Nov. 2022. Revision 32.

[36] A. Sprenkels. The Kyber/Dilithium NTT, 2022. https://electricdusk.com/ntt.html, accessed: 2022-10-17.

[37] D. Unruh. Post-Quantum Verification of Fujisaki-Okamoto. In S. Moriai and H. Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 321–352, Cham, 2020. Springer International Publishing.

X. Appendix

A. Isabelle Formalizations

1) Isabelle Code for the Quotient Ring R_q: The quotient ring R_q in Isabelle is defined in three steps:

1) Define a type class containing the modulus q and the polynomial $x^n + 1$ as `qr_poly′` and ascertain their compatibility.
2) Define the equivalence relation for polynomials in $\mathbb{Z}_q[x]$ modulo `qr_poly′`
3) Define the final type class `qr` of the quotient ring R_q using the constructor `quotient_type` by modding out the equivalence relation

```
class qr_spec = prime_card +
  fixes qr_poly′ :: 'a itself ⇒ int poly
  assumes ¬ int CARD('a) dvd
            lead_coeff (qr_poly′ TYPE('a))
  and degree (qr_poly′ TYPE('a)) > 0

definition qr_rel where
  qr_rel P Q ⟷ [P = Q] (mod qr_poly)

quotient_type (overloaded)
  'a qr = 'a :: qr_spec mod_ring poly / qr_rel
```

2) Isabelle Code for Parameter Sets: The parameter set of Kyber with the required properties is encoded as a locale.

```
locale kyber_spec =
fixes type_a :: ('a :: qr_spec) itself
  and type_k :: ('k ::finite) itself
  and n q::int and k n′::nat
assumes n = 2 ^ n′
    and n′ > 0
    and q > 2
    and prime q
    and int (CARD('a :: qr_spec)) = q
    and int (CARD('k :: finite)) = k
    and qr_poly′ TYPE('a) =
             Polynomial.monom 1 (nat n) + 1
    and q mod 4 = 1
```

3) Isabelle Code for Kyber Algorithms: The key generation in Isabelle is defined in two steps: First we sample the inputs A, s, e according to their distributions, and second we calculate the formula $As + e$. The second part is implemented in the function `key_gen`, whereas the sampling in the first step is implemented in the function `pmf_key_gen`. Here `pmf_of_set` return a uniform distribution on an input set and `beta_vec` samples a vector of polynomials according to the distribution β_η. The latter returns a probability mass function on the output, corresponding to the probability mass function on the key generation.

```
definition key_gen where
key_gen A s e = A ⋆ s + e

definition pmf_key_gen  where
"pmf_key_gen = do {
  A ← pmf_of_set (UNIV::
       (('a qr,'k) vec,'k) vec set);
  s ← beta_vec;
  e ← beta_vec;
  let t = key_gen A s e;
  return_pmf ((A, t),s)
}"
```

As with the key generation, the encryption is also split into the calculation and the sampling part. The calculation is implemented in the function `encrypt` and the sampling in the function `pmf_encrypt`. Again, `pmf_encrypt` returns a probability mass function on the ciphertext. One important fact on the formalization is that the types cast always have to be included, for example `to_module` casts the integer $\lfloor q/2 \rceil$ to the type of the module R_q and `bitstring_to_module` casts the bit-string of the message to an element in R_q.

```
definition encrypt where
encrypt t A r e1 e2 du dv m =
   (compress_vec du (A^T ⋆ r + e1),
    compress_poly dv (t^T ⋆ r + e2 +
       to_module (round(q/2)) ⋆
       bitstring_to_module m))

definition pmf_encrypt where
"pmf_encrypt pk m = do{
  r ← beta_vec;
  e1 ← beta_vec;
  e2 ← beta;
  let c = encrypt (snd pk) (fst pk)
                r e1 e2 dt du dv m;
  return_pmf c
}"
```

Note that the compression and decompression on R_q^k are defined as an index- and coefficient-wise application. However, we need to separate these definitions in Isabelle using the suffix `_vec` and `_poly`.

Since the decryption is purely deterministic, we only implement its calculation in `decrypt`.

```
definition decrypt where
decrypt u v s du dv =
  compress_poly 1 ((decompress_poly dv v)
    - s^T ⋆ (decompress_vec du u))
```

```
lemma kyber_correct:
  fixes A s r e e1 e2 du dv cu cv t u v
  assumes t = key_gen A s e
  and (u,v) = encrypt t A r e1 e2 du dv m
  and cu = compress_error_vec du
         ((transpose A) *v r + e1)
  and cv = compress_error_poly dv
         (scalar_product t r + e2 +
         to_module (round(q/2)) * m)"
  and abs_infty_poly (scalar_product e r
        + e2 + cv - scalar_product s e1
        - scalar_product s cu)
       < round (q / 4)
  and set ((coeffs o of_qr) m) ⊆ {0,1}
  shows decrypt u v s du dv = m
```

Then the corollary that Kyber is `delta_kyber`-correct is formalized as follows:

```
lemma
shows
  expectation pmf_key_gen (λ(pk, sk).
    MAX m ∈ Msgs. pmf (do {
      (u,v) ← pmf_encrypt pk m;
     return_pmf (decrypt u v sk du dv ≠ m)
    })) True)
  ≤ delta_kyber
```

5) Isabelle Code for IND-CPA property: The Isabelle formalization of Theorem VII.1 is the following:

```
theorem concrete_security_kyber:
assumes lossless: ind_cpa.lossless 𝒜
shows
  ind_cpa.advantage (ro.oracle, ro.initial) 𝒜 ≤
  mlwe.advantage (kyber_reduction1 𝒜) +
  mlwe.advantage (kyber_reduction2 𝒜)
```

Here `ro` initialized the random oracle that needs to be specified for the existing definition of the IND-CPA security game in CryptHOL. The lossless property states that the adversary $\mathcal{A}$ terminates.

B. Comparative results on δ, δ' and the correctness error

Figure 7 shows the relations between δ, δ' and the actual correctness error for small parameter sets. Two values are portrayed on the x- and y-axis. If the experimental results of a parameter set lies above the diagonal, then the value on the y axis is bigger. If the results lie below the diagonal, then the value on the x-axis is bigger. These plots show that the inequality between δ and the correctness error is violated, whereas with δ', the inequality is preserved.

C. Proof of the Auxiliary Lemma IV.2

Proof. Let x be the representative in $\{0, \ldots, q-1\}$. Then consider two cases, namely $x < \lceil q - \frac{q}{2^{d+1}} \rfloor$ and $x \geq \lceil q - \frac{q}{2^{d+1}} \rfloor$. These cases arise from the distinction whether the modulo reduction in the definition of the compression function is triggered or not. Indeed, we have $comp_d\ x = \lceil \frac{2^d}{q} x \rfloor \mod 2^d$ where $\frac{2^d}{q} x < 2^d$, but $\lceil \frac{2^d}{q} x \rfloor = 2^d$ if and only if $x \geq \lceil q - \frac{q}{2^{d+1}} \rfloor$. In the latter case, the modulo operation in the compression function is activated and returns $comp_d\ x = 0$. In the following, we will abbreviate $comp_d\ x$ by x^*.

Case 1: Let $x < \lceil q - \frac{q}{2^{d+1}} \rfloor$. Then the modulo reduction in the compression function $x^* = \lceil \frac{2^d}{q} x \rfloor \mod 2^d = \lceil \frac{2^d}{q} x \rfloor$ is not triggered. Thus we get:

$$\begin{aligned}
&|x' - x| = |decomp_d\ (x^*) - x| \\
&= \left| decomp_d\ (x^*) - \frac{q}{2^d} \cdot x^* + \frac{q}{2^d} \cdot x^* - \frac{q}{2^d} \cdot \frac{2^d}{q} \cdot x \right| \\
&\leq \left| decomp_d\ (x^*) - \frac{q}{2^d} \cdot x^* \right| + \frac{q}{2^d} \cdot \left| x^* - \frac{2^d}{q} \cdot x \right| \\
&= \left| \left\lceil \frac{q}{2^d} \cdot x^* \right\rfloor - \frac{q}{2^d} \cdot x^* \right| + \frac{q}{2^d} \cdot \left| \left\lceil \frac{2^d}{q} \cdot x \right\rfloor - \frac{2^d}{q} \cdot x \right| \\
&\leq \frac{1}{2} + \frac{q}{2^d} \cdot \frac{1}{2} = \frac{q}{2^{d+1}} + \frac{1}{2}
\end{aligned}$$

Since $x' - x$ is an integer, we also get:

$$|x' - x| \leq \left\lfloor \frac{q}{2^{d+1}} + \frac{1}{2} \right\rfloor = \left\lceil \frac{q}{2^{d+1}} \right\rfloor$$

Therefore also $|x' - x| \leq \lfloor q/2 \rfloor$ such that the $\mathrm{mod}^{\pm}$ operation does not change the outcome. Finally for this case, we get

$$|x' - x \mod^{\pm} q| \leq \left\lceil \frac{q}{2^{d+1}} \right\rfloor$$

Case 2: Let $x \geq \lceil q - \frac{q}{2^{d+1}} \rfloor$. Then the modulo operation in the compression results in the compression to zero, i.e., $comp_d\ x = 0$. Using the assumption on x, we get:

$$\begin{aligned}
|x' - x \mod^{\pm} q| &= |decomp_d\ 0 - x \mod^{\pm} q| \\
&= |-x \mod^{\pm} q| = |-x + q| \\
&\leq \left| \left\lceil q - \frac{q}{2^{d+1}} \right\rceil - q \right| = \left\lfloor \frac{q}{2^{d+1}} \right\rfloor \leq \left\lceil \frac{q}{2^{d+1}} \right\rfloor
\end{aligned}$$

□

D. NTT and the Convolution Theorem

The NTT is used to speed up the multiplication on $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ and is based on the concepts of the Discrete Fourier Transform. An introduction to the use of the NTT for lattice-based cryptography can be found in [28] or for the special case of the CRYSTALS suite in [36]. The NTT as a nega-cyclic convolution is described in [20]. To shorten this presentation, we omit all proofs which can be found in the aforementioned references.

The standard multiplication for $f = \sum_{k=0}^{n-1} f_k x^k$ and $g = \sum_{k=0}^{n-1} g_k x^k$ in R_q is given by:

$$f \cdot g = \sum_{k=0}^{n-1} \left(\sum_{j=0}^{n-1} (-1)^{k-j \text{ div } n} f_j g_{k-j \bmod n} \right) x^k$$

Thus, multiplication in R_q is done using $\mathcal{O}(n^2)$ multiplications on coefficients. Unlike multiplication, addition is calculated in

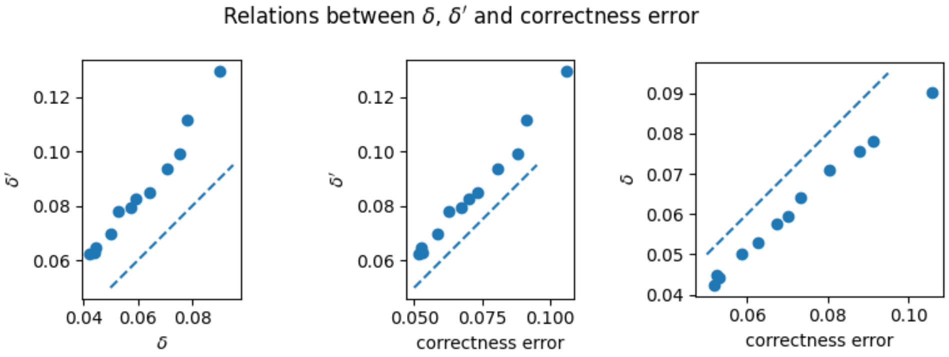


Figure 7: Relational comparisons between δ, δ' and the correctness error for small parameters

$\mathcal{O}(n)$ since addition is done entry-wise. Therefore, the most expensive part of the calculations in Kyber crypto algorithms is multiplication. Using a smarter way to multiply will make the calculations in Kyber faster.

The usual NTT requires the field $\mathbb{Z}_q$ to have a n-th root of unity, that is an element ω with $\omega^n = 1$. This can be achieved by setting $q \equiv 1 \mod n$. However, since we work over the quotient ring $Z_q[x]/(x^n+1)$, we have to consider the nega-cyclic property that $x^n \equiv -1 \mod x^n+1$ instead of the cyclic properties required by the NTT. Moreover, the original Kyber uses a "twisted" alternative which is easier to implement but requires the existence of a $2n$-th root of unity.

Considering all the constraints mentioned above, let ψ be a $2n$-th root of unity in R_q. Then we define the nega-cyclic twisted NTT on R_q for Kyber [11] as follows:

Definition 5 (NTT)**.** Let $f = \sum_{k=0}^{n-1} f_k x^k \in R_q$, then the NTT of f is defined as:

$$NTT(f) = \sum_{k=0}^{n-1} \left(\sum_{j=0}^{n-1} f_j \psi^{j(2k+1)} \right) x^k$$

The inverse transform is scaled by the factor of n^{-1} and is given by the following.

Definition 6 (inverse NTT)**.** Let $g = \sum_{k=0}^{n-1} g_k x^k \in R_q$ be in the image of the NTT, then the inverse NTT of g is defined as:

$$invNTT(g) = \sum_{k=0}^{n-1} n^{-1} \left(\sum_{j=0}^{n-1} g_j \psi^{-k(2j+1)} \right) x^k$$

We formalized a proof of correctness of the NTT and its inverse [21].

Theorem X.1. *Let f be a polynomial in R_q and g a polynomial in NTT domain. Then NTT and $invNTT$ are inverses:*

$$invNTT(NTT(f)) = f \quad and \quad NTT(invNTT(g)) = g$$

Using this transformation, we can reduce multiplications to compute within $\mathcal{O}(n\log(n))$ using a fast version of the NTT. To apply the NTT to the Kyber algorithms, we need the convolution theorem. It states that multiplication of two polynomials in R_q can be done index-wise over the NTT domain.

Theorem X.2. *Let f and g be two polynomials in R_q. Let $(\cdot)$ denote the multiplication of polynomials in R_q and $(\odot)$ the coefficient-wise multiplication of two polynomials in the NTT domain. Then the convolution theorem states:*

$$NTT(f \cdot g) = NTT(f) \odot NTT(g)$$

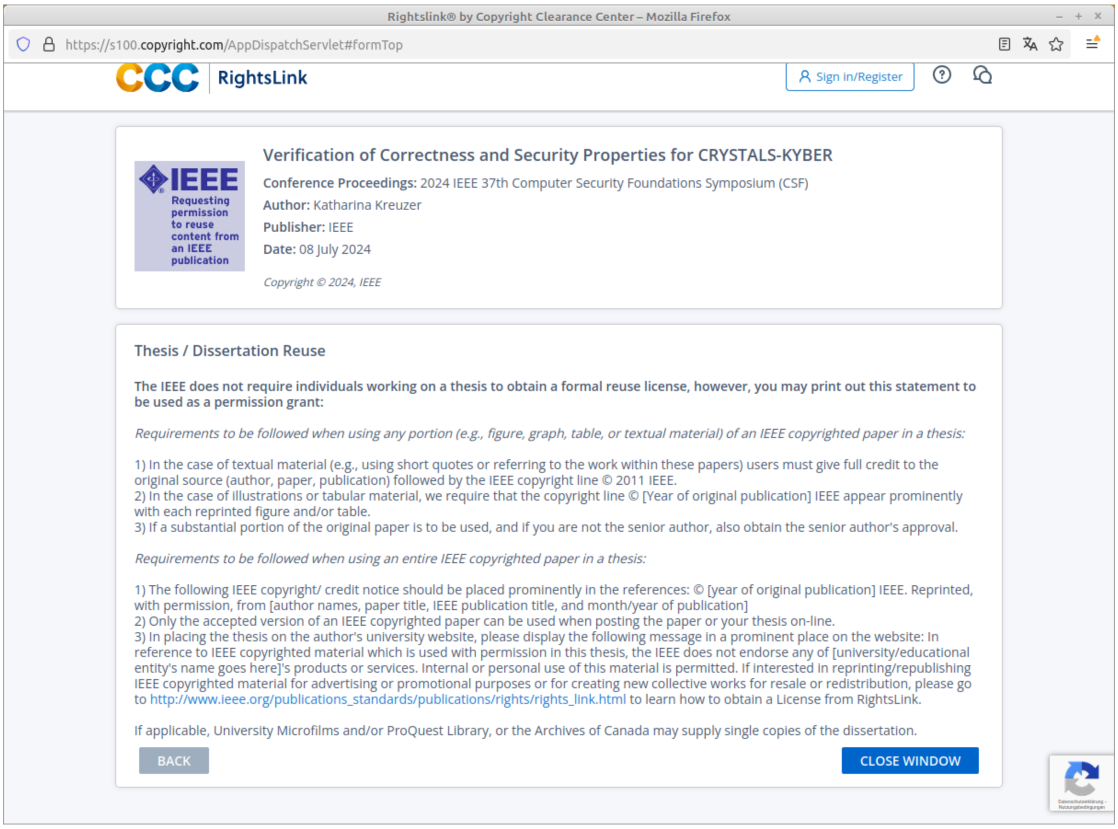
Together with Theorem X.1 this yields the fast multiplication formula.

Theorem X.3. *Let f and g be two polynomials in R_q. Let $(\cdot)$ denote the multiplication of polynomials in R_q and $\odot$ the coefficient-wise multiplication of two polynomials in the NTT domain. Then multiplication in R_q can be computed by:*

$$f \cdot g = invNTT(NTT(f) \odot NTT(g))$$

The formalization of the NTT for the original Kyber [11] was relatively straight-forward since it is based on the formalization of the standard NTT by Ammer in [3]. The only minor hindrances were the conversion between the types and working with representatives over R_q as well as the rewriting of huge sums.

Since the NTT for the recent version of Kyber [4] was also formalized in [18], we verified only the NTT for the original Kyber specifications. Note that the NTT for the latest versions of Kyber [4], [5] is a bit different, since the finite field $\mathbb{Z}_{3329}$ does not contain a $2n$-th root of unity, but only an n-th root of unity.


Rightslink® by Copyright Clearance Center – Mozilla Firefox
https://s100.copyright.com/AppDispatchServlet#formTop
CCC RightsLink
Sign in/Register
IEEE
Requesting permission to reuse content from an IEEE publication
BACK
CLOSE WINDOW

Verification of Correctness and Security Properties for CRYSTALS-KYBER

Conference Proceedings: 2024 IEEE 37th Computer Security Foundations Symposium (CSF)
Author: Katharina Kreuzer
Publisher: IEEE
Date: 08 July 2024

Copyright © 2024, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Figure B.1: Image source: RightsLink. `https://s100.copyright.com/AppDispatchServlet#formTop`, accessed on 2025-01-10

C Paper 3: Formalizing the One-Way to Hiding Theorem

Reference. Katharina Heidler and Dominique Unruh. *Formalizing the One-way to Hiding Theorem.* In Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP '25, page 243–256, New York, NY, USA, 2025. ACM. DOI: `10.1145/3703595.3705887`.

Synopsis. This paper describes the formalization of the One-way to Hiding (O2H) Theorem for quantum security proofs in Isabelle. The original semi-classical O2H Theorem (Ambainis, Hamburg and Unruh, Crypto 2019) is extended to infinite dimensional Hilbert spaces and non-terminating adversaries. A new, alternative proof for the mixed state case of the O2H Theorem is given and verified in Isabelle. A full summary of the paper can be found in Section 7.4.

Contributions. I have contributed all formalizations (apart from the formalization of Kraus maps) in Isabelle to this project. The theoretical formal model of the quantum adversary, the new proof of the mixed O2H and the extension to infinitely many dimensions and non-terminating adversaries were developed by Dominique Unruh, after I found respective gaps in the formalization. Writing the paper was a joint effort by Dominique Unruh and me.

Copyright. On the following pages, the full article is reproduced in its published form in accordance to the ACM author rights for reproduction in a dissertation. The official version in the ACM Digital Library can be found under the DOI cited above. Article published in ACM Digital Library under a CC-BY 4.0 license. To view a copy of this license, visit `https://creativecommons.org/licenses/by/4.0/`. Reproduced with friendly permission from the authors.

Formalizing the One-Way to Hiding Theorem

Katharina Heidler
TU Munich
Munich, Germany
k.kreuzer@tum.de

Dominique Unruh
RWTH Aachen
Aachen, Germany
University of Tartu
Tartu, Estonia
o2h.ap83ml@rwth.unruh.de

Abstract

As the standardization process for post-quantum cryptography progresses, the need for computer-verified security proofs against classical and quantum attackers increases. Even though some tools are already tackling this issue, none are foundational. We take a first step in this direction and present a complete formalization of the One-way to Hiding (O2H) Theorem, a central theorem for security proofs against quantum attackers. With this new formalization, we build more secure foundations for proof-checking tools in the quantum setting. Using the theorem prover Isabelle, we verify the semi-classical O2H Theorem by Ambainis, Hamburg and Unruh (Crypto 2019) in different variations. We also give a novel (and for the formalization simpler) proof to the O2H Theorem for mixed states and extend the theorem to non-terminating adversaries. This work provides a theoretical and foundational background for several verification tools and for security proofs in the quantum setting.

CCS Concepts: • **Security and privacy** → **Logic and verification**; **Mathematical foundations of cryptography**.

Keywords: Security, Verification, Quantum Adversaries, Isabelle

ACM Reference Format:
Katharina Heidler and Dominique Unruh. 2025. Formalizing the One-Way to Hiding Theorem. In *Proceedings of the 14th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '25), January 20–21, 2025, Denver, CO, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3703595.3705887

1 Introduction

With the standardization process for post-quantum cryptography led by the National Institute for Standards and Technology (NIST) of the US [19], the community makes a joint effort to find new crypto systems that are resistant against potential attacks from quantum computers. With these new systems, there is a big hurdle to overcome: showing security proofs not only against classical but also against quantum attackers.

Reasoning against quantum attackers is different than in the classical case. Let us consider the following example: In classical security proofs, we often use a "reprogramming" argument. Assume the adversary has access to some random function H (a "random oracle"). Classically, the adversary only notices reprogramming of the oracle H on some (small) input set S if they query values inside S. Therefore, we can reprogram H on the set S without the adversary noticing except with negligible probability. In the quantum setting, we cannot argue this way. A quantum attacker may query all input values in superposition in a single query, so we cannot reprogram the oracle as easily as in the classical case.

The above example shows that with quantum adversaries, we also have to rethink the mathematical toolset used in security proofs. For the reprogramming of the oracle, we can only formulate a much more elaborate statement in the quantum world: the One-way to Hiding (O2H) Theorem. This theorem bounds the distinguishing probability between a security game and its reprogramming by the probability that the adversary measures the reprogramming set.

The proof of the O2H Theorem goes deep into quantum computing theory. As quantum mechanics is no easy subject, following the proof argumentation can be quite challenging even for experts. Therefore, many people simply trust the pen-and-paper proof and assume it to be correct.

However, every pen-and-paper proof is prone to errors. For example, in the first submission of Kyber [3, 14], D'Anvers found an error in the security proof, leading to an essential change in the Kyber algorithms [2]. There have been various efforts to formalize security proofs in automated theorem provers to prevent such errors and guarantee the correctness of security proofs.

For classical adversaries, there are a lot of tools out in the world: Isabelle CryptHOL [7, 26], EasyCrypt [6, 20], CryptoVerif [10, 11] and many others. Most of these tools now strive to reason about quantum attackers as well. First and foremost, the qrhl-tool [34, 38] (based on Isabelle) uses quantum relational Hoare logic to reason with quantum attackers. Also, EasyPQC [5] (a development of EasyCrypt) tries to bridge the gap to quantum security. Recently, CryptoVerif

CPP '25, January 20–21, 2025, Denver, CO, USA

ACM ISBN 979-8-4007-1347-7/25/01
https://doi.org/10.1145/3703595.3705887

has developed strategies to handle quantum adversaries as well [12].

The main problem of most of these tools is that they often make complex mathematical assumptions and do not have foundational tool-checked proofs thereof. This implies that the security proofs are only verified up to the mathematical assumptions these tools require. If there was a bug in the implementation of one of these assumptions, it may lead to the verification of wrong results.

Our goal with this project is to start a foundational formalization for theorems needed in reasoning against quantum adversaries. Our first step in this direction is the result of this paper: a first formalization of the O2H Theorem in the theorem prover Isabelle. In the future, we aim to extend our work to more concrete bounds and a connection to other tools like qrhl-tool.

1.1 Contributions

With this work, we present a foundational formalization of the semi-classical One-way to Hiding (O2H) Theorem [1] in the theorem prover Isabelle. To our knowledge, this is the first formalization of the O2H in any theorem prover so far. Our main focus in this paper lies on the formalization process, describing the proof outline, problems we faced and how we solved them.

First, we introduce all relevant mathematical notions (Section 2). This includes the terminology of pure and mixed states and the model of quantum registers. We then state the O2H Theorem (Section 3) and give a short overview of the proof structure (Section 4). Our main design choices are discussed in Section 5.

The quantum adversary model, its execution and our formalization thereof (Section 6) include the formalism for oracle queries, the adversaries and the final states. The main difference to Ambainis, Hamburg and Unruh's work [1] is that we only allow sequential queries. The formalization of parallel queries was omitted since Isabelle does not have dependent types. Since the number of parallel queries is unknown, the type of the query would depend on a variable, needing dependent types.

Then, we can state the O2H Theorem [1] in six alternative versions. For the rest of the paper, we discuss the formalization of the proof (Section 7). For pure state adversaries, we follow the pen-and-paper proof [1] closely. However, we give an alternative proof for mixed states, foregoing the notion of Bures distances and fidelity of quantum states and using limit arguments. We follow a set of generalization steps for the proof outline, leading to a new version of the O2H including non-terminating adversaries and an extension to infinite-dimensional Hilbert spaces. We also discuss challenges during the formalization efforts and their solutions and include some technical details at the end.

1.2 Related Work

The O2H was first described by Unruh [32] when exploring revocable quantum time-release encryption. It was used to deal with the problem of rewriting the oracle in the quantum setting. As the first O2H was quite restricted, Ambainis, Hamburg and Unruh [1] generalized the results to a so-called semi-classical O2H. The semi-classical refers to the adversary measuring only whether the reprogramming set was queried and nothing else. This leads to tighter bounds in many cases. Ambainis et al. [1] then generalized the semi-classical O2H even more giving more flexibility and tighter bounds. Bindel et al. [9] give a different, tighter bound but for a specific use-case with limited applicability elsewhere. Kuchta et al. [25] used the measure-rewind-measure method to get even tighter bounds on the O2H. Other variants of the O2H were developed by Unruh [31, 33], Jiang et al. [23], Eaton [17], Czajkowski et al. [16] and many more. For our work, we chose the original semi-classical O2H [1]. This formulation gives a wide range of applicability due to its generality, but still has quite strong bounds and easier proofs for formalization.

As mentioned in the introduction, most tools for reasoning about quantum security proofs are not foundational. To our knowledge, none of these tools have fully formalized any variant of the O2H so far. The main tools for tackling quantum security proofs, we know about, are the qrhl-tool [34, 38], EasyPQC [5] and CryptoVerif [12]. However, all of these tools are still under construction as this field is still very new and evolving with new research coming out. We chose qrhl-tool as our aim for application of our formalization, as it is the most developed tool of the above.

As Isabelle underlies the qrhl-tool for the formalization of intermediate steps, our formalizations have been carried out in Isabelle. Isabelle [27, 29, 30] provides a vast library on relevant topics such as operator theory [15], quantum registers [35], Hilbert spaces, analysis and probability theory in the Archive of Formal Proofs (AFP) [18] and the distribution. Isabelle's versatile locale context [4, 24] allows instantiations in different settings.

2 Mathematical Background

We will briefly sketch the mathematical and quantum mechanical background needed for this paper. For more background on quantum computation theory (at least in the finite-dimensional case), the reader is referred to introductory books on this topic [28, 39]. When considering infinite-dimensional vector spaces, a number of extra subtleties need to be considered. We write those details in brackets ([]); those can be ignored at first reading. Readers familiar with quantum computing should still read subsection 2.4.

2.1 Miscellaneous Notation

$\mathbb{I}$ denotes the identity. $\|x\|$ the Euclidean norm of x. Given a matrix [a bounded operator] or vector A, we write A^* for its conjugate transpose [its adjoint].

2.2 Quantum Mechanics – Pure States

In its simplest form, a "pure" quantum state is a vector ψ of norm 1 in a complex Hilbert space $\mathbb{C}^n$ [in an arbitrary Hilbert space]. It can be seen as a linear combination of basis vectors, each of them corresponding to a classical state. So the quantum state represents a "superposition" of classical states. Those basis vectors / "classical states" themselves are written $|x\rangle$. E.g., an n-qubit quantum system has space $\mathbb{C}^{2^n}$ with (orthonormal) basis $\{|x\rangle\}_{x\in\{0,1\}^n}$.

An operation on a quantum state is represented by a matrix [linear operator] applied to the state. This matrix is required to be an isometry, that is, be norm-preserving. (Many presentations require the stronger notion of unitary. To avoid too unfamiliar language, we will usually talk about unitaries instead of isometries in the following.)

In addition to applying a matrix, we can perform measurements. For simplicity, we consider only binary measurements (measurements with outcomes "yes" and "no"). Such a measurement is modeled by an (orthogonal) projector P. Given a state ψ, $\|P\psi\|^2$ is the probability of getting outcome "yes", and $\|(\mathbb{I}-P)\psi\|^2$ of outcome "no". Measurements also affect the observed state; after measuring "yes", the state is $P\psi/\|P\psi\|$, after measuring "no", $(\mathbb{I}-P)\psi/\|(\mathbb{I}-P)\psi\|$.

The projector onto $|x\rangle$ is written $|x\rangle\langle x|$. So, e.g., a measurement whether a quantum system is in the classical state 0 is represented by the projector $|0\rangle\langle 0|$.

Often, we consider a quantum system composed of several systems, e.g., of $\mathbb{C}^n$ and $\mathbb{C}^m$. The resulting space consists of vectors in the space $\mathbb{C}^n \otimes \mathbb{C}^m := \mathbb{C}^{nm}$ [the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ of Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$]. A composed system that has state ψ in the first subsystem, and ϕ in the second subsystem, has state $\psi\otimes\phi$. The vector $\psi\otimes\phi$ has all the values $\psi_i\phi_j$ as coefficients [defined more abstractly in infinite-dimensions]. Similarly, we can take tensor products of operators. For example, applying U to the first subsystem and the identity to the second is described by the tensor product of matrices $U\otimes\mathbb{I}$, which is an $nm\times nm$-matrix [an operator on $\mathcal{H}_1\otimes\mathcal{H}_2$].

2.3 Quantum Mechanics – Mixed States

While many things can be described with the formalization of quantum mechanics using pure states, this formalism turns out to be limited when we want to talk, e.g., about probabilistic distributions and quantum states together (as often needed in cryptography). Because of that, there is a more general formalism that allows us to describe quantum states that are effectively probability distributions of pure states, so called "mixed states". A mixed state is described by a matrix [trace-class operator] $\rho \in \mathbb{C}^{n\times n}$, called a density operator. To represent a mixed state that is with probability 1 the pure state ψ (a vector) in this formalism, we define $\rho := \psi\psi^*$. (Notice that $\psi\psi^*$ is the product of a row and a column vector, thus a matrix.) A distribution of states ψ_i with corresponding probabilities p_i is described by $\rho := \sum p_i\psi_i\psi_i^*$ [with convergence in the trace-norm]. Applying a unitary U to a mixed state ρ results in $U\rho U^*$. Measuring with projector P gives "yes" with probability $\operatorname{tr} P\rho$ and "no" with $\operatorname{tr}(\mathbb{I}-P)\rho$; the state after the measurement is $P\rho P/\operatorname{tr} P\rho$ or $(\mathbb{I}-P)\rho(\mathbb{I}-P)/\operatorname{tr}(\mathbb{I}-P)\rho$, respectively.

The reader may verify that when the state ρ is "pure", i.e., of the form $\psi\psi^*$, these definitions simplify to the ones from the previous section.

The valid quantum states are exactly the positive operators of trace 1, corresponding to probability distributions with total probability 1. Often, we may also consider density operators with trace ≤ 1, called subdensity operators. These correspond to subprobability distributions with total probability ≤ 1. (Useful for modeling programs that terminate with probability ≤ 1.)

In addition to these operations, we can also define more general operations. For example, applying U with probability $1/2$ would map $\rho \mapsto \frac{1}{2}\rho + \frac{1}{2}U\rho U^*$. The general form for such mixed probabilistic and quantum operations is $\rho \mapsto \sum E_i\rho E_i^*$ for some E_i with $\sum E_i^*E_i = \mathbb{I}$ (or $\le \mathbb{I}$ if we allow subprobability distributions; we call this the normalization condition) [convergence of the first sum is with respect to the trace norm, and for the second with respect to the weak operator topology]. We call such maps "Kraus maps" (and the E_i Kraus operators). For example, applying a unitary U can be represented as the Kraus map with single Kraus operator $E_1 := U$. Given two Kraus maps $\mathcal{E}$ and $\mathcal{F}$ represented by $\{E_i\}_{i\in I}$ and $\{F_j\}_{j\in J}$, respectively, their functional composition $\mathcal{E}\circ\mathcal{F}$ is a Kraus map with operators $\{E_i\cdot F_j\}_{(i,j)\in I\times J}$. The normalization condition is also preserved. In the mixed state formalism, we can also compose systems. If subsystems A, B are in states ρ_A, ρ_B, respectively, the joint state of the composed system AB is in state $\rho_A\otimes\rho_B$ (tensor product of matrices [of bounded operators]). Applying a Kraus map $\{E_i\}_i$ to A corresponds to applying the Kraus map $\{E_i\otimes\mathbb{I}\}_i$ to AB.

2.4 Quantum Registers

As mentioned in the previous two sections, we can model systems consisting of several parts by composing them. E.g., if a program performs a unitary U on the first part of some system, we can describe the effect on the whole system as $U\otimes\mathbb{I}$, and similar for measurements, Kraus maps, and many more. However, doing this explicitly gets unwieldy quickly. For example, applying an operation to the fifth out of 25 subsystems becomes $\mathbb{I}\otimes\cdots\otimes\mathbb{I}\otimes U\otimes\mathbb{I}\otimes\cdots\otimes\mathbb{I}$. And when formalizing, we cannot even conveniently write "$\cdots$". Furthermore, such a rigid structure of tensor factors makes it hard to write formal statements in a general way (without imposing some arbitrary assumptions about, e.g., the number of subsystems). In informal practice, we therefore simply talk about named "registers". E.g., we call the fifth subsystem the register X, and say "we apply U to X". It is then assumed to be understood from context how a given operation is applied to a register and what it means for the overall system. A formal definition of a register (that encompasses a tensor factor as above but can also point to more general subsystems such as, e.g., "the first and last tensor factors together") was presented by Unruh [36] under the name of quantum references and formalized in Isabelle/HOL in [35]. We follow their treatment and use their Isabelle formalization. A register F with content type $\mathcal{H}_c$ and memory type $\mathcal{H}_m$ is then a mathematical object that points to a subsystem described by a Hilbert space $\mathcal{H}_c$ within a larger system $\mathcal{H}_m$.[1]

It supports various lifting operations, e.g., given a unitary U on $\mathcal{H}_c$, we get $F(U)$ on $\mathcal{H}_m$ that describes what happens to the overall system when U is applied to the subsystem. We can also combine registers in various ways, e.g., if F, G represent two disjoint subsystems (formalized as registers), we can construct the register FG consisting of the content of both these registers. We will heavily make use of this formalism to distinguish between the adversary registers, registers of the random oracle, auxiliary registers, etc.

3 The O2H Theorem

In cryptography, security properties are often written as games against an adversary. Classically, we understand quite well how the adversary should be modelled and have a large toolbox for game reductions against classical adversaries.

One particularly important model is the random oracle model [8] which is an idealization of cryptographic hash functions. In this model, all honest parties and the adversary have access to a uniformly randomly chosen function (called the random oracle) that they can query (i.e., evaluate in a single time step) at their leisure. The random oracle model is very powerful because we have many strong proof techniques, for example the following reasoning: Classically, an adversary can only query a finite number of values from a random oracle (e.g. a hash function). This guarantees we can change values that have yet to be queried without the adversary noticing. Many cryptographic proofs rely on this property.

Quantumly, however, this is not the case. In the quantum random oracle model (first explicitly studied in [13]), a quantum adversary may query all values of an oracle in superposition. That raises the question: Can we still change values in the oracle (for security proofs) without the adversary noticing (or at least with a relatively small probability that the adversary notices)?

The One-way to Hiding (O2H) Theorem tries to solve this question in the quantum case. Informally, it states: Assume we have two oracles H and G that agree everywhere but on a set S. We define two games: in the first game the adversary has access to H, in the second the adversary has access to G. In both, the adversary can query the oracle at most d times. Then, the probability with which the adversary can distinguish the two games can be bounded by $4\sqrt{(d+1)P_{\mathit{find}}}$ where P_{find} is the probability that the adversary queries values in S.

The full statement of the theorem is as follows:

Theorem 3.1 (O2H Theorem [1, Theorem 1][2]). *Let $S \subseteq X$, $G, H : X \to Y$ with $\forall x \notin S.\ G(x) = H(x)$ and z a bitstring. S, G, H, z are chosen randomly according to an arbitrary joint distribution. Let $\mathcal{A}$ be a mixed, terminating adversary with access to an oracle and query depth d. Define the following:*

$$P_{\mathit{left}} := Pr[b = 1\ :\ b \leftarrow \mathcal{A}^H(z)] \tag{1}$$

$$P^{(1)}_{\mathit{right}} := Pr[b = 1\ :\ b \leftarrow \mathcal{A}^G(z)] \tag{2}$$

$$P_{\mathit{find}} := Pr[Find\ :\ \mathcal{A}^{H\setminus S}] \tag{3}$$

Then the following inequalities hold:

$$\left|P_{\mathit{left}} - P^{(1)}_{\mathit{right}}\right| \leq 4\sqrt{(d+1)\cdot P_{\mathit{find}}} \tag{4}$$

$$\left|\sqrt{P_{\mathit{left}}} - \sqrt{P^{(1)}_{\mathit{right}}}\right| \leq 2\sqrt{(d+1)\cdot P_{\mathit{find}}} \tag{5}$$

Let us first explain the notation:

$$P_{\mathit{left}} := \Pr[b = 1 : b \leftarrow \mathcal{A}^H(z)]$$

denotes the probability that $b = 1$ when b is the return value of running adversary $\mathcal{A}$ with oracle access (superposition queries) to H and input z. Analogously for $P^{(1)}_{\mathit{right}}$. So $\left|P_{\mathit{left}} - P^{(1)}_{\mathit{right}}\right|$ is the probability with which the adversary distinguishes H and G, and (4) bounds that probability. (And so does (5), in a somewhat less intuitive formulation that turns out to give better bounds in many cryptographic proofs.) $\mathcal{A}^{H\setminus S}$ denotes the adversary running with a so-called semi-classical oracle: Whenever the adversary queries the oracle, the oracle evaluates H like the normal quantum random oracle, but additionally measures whether the input is in S or not. It does not measure anything beyond this, in particular not the concrete input! (Recall that measurements influence the quantum state. So measuring more information changes the effect of the oracle, even if that additional information is discarded afterwards.) Then **Find** denotes the event that in one of the queries, it is measured that the input is in S. Thus P_{find} is the probability that the adversary queries a value in S. Hence Theorem 3.1 indeed matches the informal description above.

[1] The formal definition of a register is a [weak*-continuous bounded] unital *-homomorphism. But it is easiest not to think of that and to only use the abstract properties of registers.

[2] The theorem stated here is a slight weakening of [1, Theorem 1]: The bounds are worse by a factor of 2 in the bounds (cf. Section 7.1).

In the following, let S always denote the change set of the oracle, d the adversary's query depth, and H and G oracles that the adversary may have access to. If not stated otherwise, we assume H and G to be the same up to the change set, i.e. $\forall x \notin S.\ H(x) = G(x)$. (As in Theorem 3.1 above.)

4 Proof Overview

The basic proof follows [1]. However, after showing the theorem for pure states, we deviate strongly from the original proof. This is because we wished to avoid formalizing the Bures distance and fidelity (and their properties), and also did not have the theorem available that any adversary can w.l.o.g. be represented as a unitary adversary on a larger space. Instead, our proof went through the following stages of greater and greater generality:

1. Proof of the O2H Theorem for a pure adversary (not using measurements etc.) and fixed H, S and z. This allows us to use the simpler pure state formalism (Section 2.2). (This proof follows closely [1, Lemma 1], except for infinite dimensions).
2. Proof of the O2H theorem for pure adversaries, but stated in the mixed state formalism. (That is, all states are formalized as mixed states but happen to be of the form $\psi\psi^*$.) This serves as glue between the previous and the following steps.
3. Proof for O2H with pure adversaries and expectation over random H, S and z. For this step, we already need mixed states because the expectation over pure states cannot be expressed as a pure state.
4. Proof of the O2H Theorem for mixed adversaries (that can do measurements etc.), but only those that are described by finite Kraus maps. (That is, a Kraus map $\{E_i\}_i$ that consists of only finitely many E_i.) The crucial point here is that an adversary represented as a finite Kraus map can be seen as a linear combination of finitely many pure adversaries, so we can lift the result from the previous step by linearity.
5. Proof of the O2H theorem with mixed adversaries represented by arbitrary (possibly infinite) Kraus maps.
6. Proof of all the different variants of the O2H theorem (as in [1]). These differ in the definition of P_{right} and are corollaries of the main variant.

5 Design Choices

Infinite-dimensional quantum mechanics. Many results (both pen-and-paper and formalized) prefer to only consider finite-dimensional vector spaces. This is roughly analogous to modeling all variables in a classical program to have a finite type, and disallowing, e.g., arbitrary-length integers and lists. While the mathematics behind finite-dimensional spaces is considerably simpler, and more familiar to many people, we chose to formalize the infinite-dimensional case for the following reasons: (a) While real-world computers are always finite (the number of bits in their memory is fixed), when defining the semantics abstractly, we often allow unbounded datatypes. (E.g., integers in Python are unbounded. Lists and set datastructures are unbounded in most languages.) Similarly, in cryptographic proofs, we usually consider semantics for programs or adversaries that allow unbounded types. (We avoid saying. e.g., that a counter is a 64-bit integer and deal with special cases such as overflows.) And then we simply *posthoc* add the condition that the adversary takes only a certain number of steps. (b) We strive for eventual integration with qrhl-tool which also uses infinite-dimensional spaces. (c) As it turns out, we need to allow infinite-dimensional quantum registers anyway for our proof: For technical reasons related to the lack of dependent types in Isabelle/HOL, we introduce a "counting register" in an intermediate adversary that is allowed to contain unbounded lists of bits. To model this intermediate adversary, we need infinite-dimensions anyway. (d) Having the results for infinite-dimensions is more general; the finite-dimensional results follow as an immediate special case.

Kraus maps vs. CPTPMs. Often, operations performed by adversaries or programs are, semantically, modeled as completely positive trace-preserving maps (CPTPMs). Yet, we choose to use another formalism, called Kraus map (or operator-sum representations), see Section 2.3. Kraus maps and CPTPMs are known to be equivalent in finite and countably dimensional spaces (implicit in [22, Section 3.1]), but it is unknown whether they represent the same class of functions in higher dimensions. Why did we choose Kraus maps? (a) It is not clear whether the proof of the O2H Theorem actually works when adversaries are arbitrary CPTPMs. The original proof [1] is in finite dimensions only (in which case CPTPMs and Kraus maps coincide anyway) and uses the assumption that any CPTPM can be represented as a unitary adversary in a larger space. This fact is quite commonly used in quantum cryptographic proofs but, to the best of our knowledge, not known to hold in infinite dimensions for CPTPMs. Yet for Kraus maps it holds. Therefore we consider Kraus maps to be a more reasonable model for quantum adversaries and programs when not restricted to finite or countable dimensions. (b) A formalization of Kraus maps already exists in Isabelle/HOL (as part of qrhl-tool [38, Kraus_Maps.thy]), while it is unclear how difficult the mathematics behind CPTPMs would be.

Isabelle/HOL vs. other theorem provers. Why did we chose Isabelle/HOL? (a) Again, the desired compatibility with qrhl-tool mandates it. (b) An extensive formalization of the required mathematical background exists [15, 35, 37].(c) The only restriction is that Isabelle does not allow dependent types. This is a limitation in some places but did not limit the development too much.

6 Modeling the Adversary and Execution

6.1 Oracle Queries

Normal queries. Before we can start formulating the adversary, we need to formulate the oracle.

Let $H : X \rightarrow Y$ be the oracle function with input state X and output space Y (as a classical function). First, we introduce quantum registers X and Y for storing the input and output of H when invoking it as a quantum oracle. The register formalism from Section 2.4 allows us to just declare two disjoint registers with content spaces $\mathbb{C}^X$ and $\mathbb{C}^Y$ [$\ell_2(X)$, $\ell_2(Y)$ in infinite dimensions]. (We use Isabelle-locales [4, 24] to collect all declarations and assumptions as a nice "package".) Recall that we write $|x\rangle$ for a classical bitstring x to denote the stored value of x in the quantum register X.

Then an oracle query is usually defined as the unitary mapping $U : |x, y\rangle \mapsto |x, y + H(x)\rangle$ where $+$ is a group operation (in typical settings: XOR on bitstrings). This operation, however, cannot be directly applied to the overall state of the system (it does not know that it should be applied to the registers X, Y). The register formalism allows us to define $U^H_{query} := XY(U)$. This means that U^H_{query} operates on the combined register XY.[3]

In our formalization [21], U is written `Uquery H`, and U^H_{query} as `(X;Y) (Uquery H)`.

Punctured oracle. Yet, for the proof we will need several variants of the above oracle, corresponding to punctured oracles. These do not just perform U^H_{query}, but additionally measure whether the input (in register X) is in the set S. A measurement whether X contains a value in S is modeled by the projector onto the span of all $|x\rangle$ with $x \in S$. We write span$|S\rangle$ for that span (in slight abuse of notation) and $Proj_{classical}(S)$ for the projector onto these $|x\rangle$. (Written `proj_classical_set S` in our formalization [21].) And thus the projector defining the measurement whether X contains a value in S is represented as $S_{embed} := X(Proj_{classical}(S))$ since the register X lifts operators on the content space to the overall memory space. S_{embed} is realised as the operator `S_embed` in our formalization [21]. In [1], the punctured oracle is implemented in several different ways. The default one is to simply perform the measurement described by S_{embed}.

Another variant does not measure, but keeps a log (in superposition) of queries in S. That is, before each query to U^H_{query}, we apply the following counting operator:

Definition 6.1 (Counting Operator)**.** Let $S \subseteq X$ be the change set. The counting operator for a counting function c with operator U_c is then defined as:

$$U_{count} = S_{embed} \otimes U_c + (\mathbb{I} - S_{embed}) \otimes \mathbb{I}$$

Intuitively, U_{count} applies U_c to the second part of the memory whenever X contains a value in S. U_{count} is unitary/an isometry if U_c is.

We distinguish two kinds of counting oracles. One that increases a counter mod q (U_c maps $|i\rangle \mapsto |i + 1 \bmod q\rangle$), and one that flips a bit in a list (U_c maps $|l\rangle$ to the bit list l with the i-th bit flipped where i is the number of the query).[4]

6.2 The Adversary

Pure states. For the simplest version of the O2H Theorem (for pure states), we formalize what an adversary on pure states is and how it is executed. Essentially, an adversary is nothing but the repeated application of a unitary operation (or more generally, any operator of norm ≤ 1 for supporting non-terminating adversaries), interleaved with invocations of the oracle.

Definition 6.2 (Pure Adversary)**.** Let ϕ_{init} be the initial (pure) state and H the oracle with unitary U^H_{query}. The pure adversary $\mathcal{A}$ of query depth d is given by $d + 1$ unitaries/operators $\{U^{\mathcal{A}}_i\}_{i\in\{0,\ldots,d\}}$ such that the adversary run can be represented by

$$\mathcal{A}(\phi_{init}) = U^{\mathcal{A}}_d \cdot U^H_{query} \cdot U^{\mathcal{A}}_{d-1} \cdots U^{\mathcal{A}}_1 \cdot U^H_{query} \cdot U^{\mathcal{A}}_0 \cdot \phi_{init}$$

In our formalisation [21], the pure adversary is formalised as the function `run_pure_adv`. When including an additional counting operator as input to `run_pure_adv`, this also formalises the execution of an adversary with a punctured oracle (an execution of $\mathcal{A}^{H\setminus S}$, i.e., where each U^H_{query} is preceded by U_{count} for suitable counting functions U_c). For abbreviation, we introduce the Isabelle type `pure_adv` for pure adversaries.

Mixed states. For the more general versions of the O2H theorem, we need to express adversaries that work on mixed

[3] The main drawback of our current formalisation is that we do not allow parallel queries by the adversary as opposed to [1]. Again, the problem is the dependent type problem: Assuming the adversary performs q parallel queries of H, the type of the input/output registers would depend on q, which is impossible in Isabelle. Since implementing a solution to this problem was too time-consuming, we leave this generalisation for future work.

[4] Here we encounter a technical challenge: In [1], the counter is mod q, and the list has length q, so the corresponding quantum spaces would need to have dimension q and 2^q, respectively. However, in Isabelle/HOL, we do not have dependent types, so we cannot easily have vector spaces of dimension depending on q. It is possible to simply make the theorem parametric in a type `'c` and to add the assumption that is has, e.g., size 2^q. However, that only pushes the difficulty in instantiating such a type to the user of the theorem. Instead, we used a more flexible, but also a bit more complex approach: We do parametrize our theorems over `'c`, but we do not assume that it is the type of length-q lists, but instead that length-q lists can be embedded in it. We then add assumptions to our theorem about the existence of constants representing the empty list, and bit flips, and accessors. This allows to instantiate the theorem later both with a finite type of size q (for those preferring finite types and willing to fix q as a constant), or as the type of bit lists (for those wanting to keep q as a parameter and accepting infinite types). All these parameters and assumptions are encapsulated in a locale, called `o2h_setting`.

states. Recall (Section 2.3) that we can describe the computation performed by an arbitrary adversary (or any physical process) by a Kraus map. To model oracle queries, we intersperse the Kraus maps describing the adversary by oracle queries. This leads to the following definition of a mixed adversary:

Definition 6.3 (Mixed Adversary)**.** Let ρ_{init} be the initial (mixed) state and H the oracle with unitary U^H_{query}. Let $\mathcal{E}^H_{query}$ be the oracle query applied as a Kraus map, i.e. $\mathcal{E}^H_{query}(\rho) = U^H_{query} \cdot \rho \cdot (U^H_{query})^*$. The mixed adversary $\mathcal{A}$ of query depth d is given by $d+1$ Kraus maps $\{\mathcal{E}^{\mathcal{A}}_i\}_{i\in\{0,\dots,d\}}$ such that the adversary run can be represented by

$$\mathcal{A}(\rho_{init}) = (\mathcal{E}^{\mathcal{A}}_d \circ \mathcal{E}^H_{query} \circ \mathcal{E}^{\mathcal{A}}_{d-1} \circ \cdots \circ \mathcal{E}^{\mathcal{A}}_1 \circ \mathcal{E}^H_{query} \circ \mathcal{E}^{\mathcal{A}}_0)(\rho_{init})$$

where "∘" is the functional composition of the Kraus maps.

Mixed adversaries are formalised as the Isabelle function run_mixed_adv. Again, with an additional input of a counting operator, this formalises the execution of mixed adversaries for punctured oracles as well. For brevity, we also introduce the type kraus_adv for mixed adversaries.

6.3 Final States and Probabilities

In the statement of the O2H, the adversary always outputs a single bit in the end. For example, in the security property for indistinguishability under a chosen plaintext attack, the adversary must guess which one of two messages was encrypted under certain conditions and outputs a bit. Many such security properties can be formulated such that the adversary only outputs a boolean.

Knowing that the adversary will return a bit, the final measurement M can be represented by a projection P and its complement $\mathbb{I} - P$. That is $M = \{P, \mathbb{I} - P\}$. Since we are interested in the probability of success of the adversary $\mathcal{A}$, this can be calculated as $Pr(b = 1, b \leftarrow \mathcal{A}) = \mathrm{tr}(P\rho)$ where ρ is the final state after running the adversary. We denote by P_M the function $\rho \mapsto \mathrm{tr}(P\rho)$.

For the statement of the O2H and its proof, there are several states to consider:

- the run of the adversary $\mathcal{A}$ without any changes in generalization steps:
  - ψ_{left}: the pure state for fixed oracle H and change set S
  - ρ_{left}: the mixed state with expectation over H, S
  - P_{left}: the measurement of ρ_{left} with P_M
- the run of the adversary $\mathcal{A}$ with counting the number of queries on a change set S in generalization steps:
  - ψ_{count}: the pure state for fixed oracle H and change set S
  - ρ_{count}: the mixed state with expectation over H, S
- the run of the adversary $\mathcal{A}$ with counting and remembering the placements of the queries on a change set S in generalisation steps:
  - ψ_{right}: the pure state for fixed oracle H and change set S
  - ρ_{right}: the mixed state with expectation over H, S
  - P_{right}: the measurement of ρ_{right} with some P_M (according to the different definitions of P_{right} in [1, Thm 1], the definition of the final measurement M on the counting register changes)
- P_{find} is the probability that the adversary queried a value in S, i.e. measuring the state ρ_{right} with the measurement $M = \{Q, \mathbb{I} - Q\}$ for the projection $Q = \mathbb{I} \otimes (\mathbb{I} - |0\rangle\langle 0|)$ (which measures if there is a non-zero element in the counting register)
- $P_{nonterm}$ is the additional non-termination part defined as $P_{nonterm} = \|\rho_{count}\|^2 - \|\rho_{right}\|^2$. This term only comes into play for non-terminating adversaries.

In the case above, P_{left} corresponds to the probability P_{left} from the O2H [1, Thm 1] defined as $P_{left} = Pr[b = 1 : b \leftarrow \mathcal{A}^H(z)]$. Similarly, P_{find} also corresponds to the definition $P_{find} = Pr[Find : \mathcal{A}^{H\setminus S}]$ from [1, Thm 1]. Here, $Find$ is the event that the adversary queries a value in S. For P_{right}, there are various alternative definitions.

Ambainis, Hamburg and Unruh [1, Theorem 1] consider six variations of the O2H by giving six definitions of P_{right}. We list them here:

Definition 6.4 (Definitions of P_{right})**.**

1. $P^{(1)}_{right} = Pr[b = 1 : b \leftarrow \mathcal{A}^G(z)]$
2. $P^{(2)}_{right} = Pr[b = 1 : b \leftarrow \mathcal{A}^{H\setminus S}(z)]$
3. $P^{(3)}_{right} = Pr[b = 1 \wedge \neg Find : b \leftarrow \mathcal{A}^{H\setminus S}(z)]$
4. $P^{(4)}_{right} = Pr[b = 1 \wedge \neg Find : b \leftarrow \mathcal{A}^{G\setminus S}(z)]$
5. $P^{(5)}_{right} = Pr[b = 1 \vee Find : b \leftarrow \mathcal{A}^{H\setminus S}(z)]$
6. $P^{(6)}_{right} = Pr[b = 1 \vee Find : b \leftarrow \mathcal{A}^{G\setminus S}(z)]$

We formalise these definitions using different measurement projections. Let P denote the final measurement of the binary outcome of $\mathcal{A}^H$. Then, the binary outcome of $\mathcal{A}^G$ can also be measured by P. For the punctuations $\mathcal{A}^{H\setminus S}$ and $\mathcal{A}^{G\setminus S}$, we have an additional counting register.

Therefore, we need to extend the final projection. For $P^{(2)}_{right}$, the measurement $P \otimes \mathbb{I}$ yields the full probability that the punctured adversary $\mathcal{A}^{H\setminus S}$ returns 1. For the rest of the definitions of P_{right}, things get more complicated.

In the case of $P^{(3)}_{right}$, we consider the event that the punctured adversary returns 1 and the event that the adversary did not query a value in S (i.e. the counting register only contains zero). These two events operate on separate registers by the projections $P \otimes \mathbb{I}$ and $\mathbb{I} \otimes |0\rangle\langle 0|$. As the two projective spaces are orthogonal, we may use the joint projection $(P \otimes \mathbb{I}) \circ (\mathbb{I} \otimes |0\rangle\langle 0|) = P \otimes |0\rangle\langle 0|$. Similarly for $P^{(4)}_{right}$, we use the same projection on the punctured adversary $\mathcal{A}^{G\setminus S}$.

For $P^{(5)}_{right}$, the probability describes the event that the punctured adversary returns 1 or the adversary queried a value in S (so the counting register is non-empty). Again, these two events operate on separate registers by the projections $P \otimes \mathbb{I}$ and $\mathbb{I} \otimes (\mathbb{I} - |0\rangle\langle 0|)$. Therefore, we get the measurement projector $P \otimes \mathbb{I} + \mathbb{I} \otimes (\mathbb{I} - |0\rangle\langle 0|)$. However, in this case, if we want to describe this as a single projection, this is the projection on the join of the projective spaces generated by $P \otimes \mathbb{I}$ and $\mathbb{I} \otimes (\mathbb{I} - |0\rangle\langle 0|)$. Again, for $P^{(6)}_{right}$, we use the same projection on the punctured adversary $\mathcal{A}^{G\setminus S}$.

7 Formalizing the Proof of the O2H

Recall Theorem 3.1 from Section 3. It corresponds to the statement from [1, Thm. 1], except that the bound differs by a factor of 2 (i.e., the version we prove here is slightly weaker, but to a degree not practically relevant).

We will address this issue in the next Section 7.1.

Note that the O2H also holds for the other definitions of P_{right} from Definition 6.4, but with tighter scalar factors. For $i \in \{2, \dots, 6\}$, we have:

$$|P_{left} - P^{(i)}_{right}| \leq 2\sqrt{(d+1) \cdot P_{find}} \tag{6}$$

$$|\sqrt{P_{left}} - \sqrt{P^{(i)}_{right}}| \leq \sqrt{(d+1) \cdot P_{find}} \tag{7}$$

7.1 Changes from Pen-and-Paper to Formalization

During the formalisation effort, we encountered several problems requiring a different approach to the original pen-and-paper proof. In this section, we give an overview of these changes.

First, we present a more general formulation of the O2H with possibly non-terminating adversaries. That is, the pure adversary given by a set $\{U_i\}_{i \in I}$ of updates must all satisfy $\|U_i\| \leq 1$ (before, the U_i were unitaries). For mixed adversaries given by a set of Kraus maps $\{\mathcal{E}_i\}_{i \in I}$, every Kraus map $\mathcal{E}_i$ given by $\{U^{(i)}_j\}_{j \in I_j}$ must suffice $\sum_{j \in I_i} (U^{(i)}_j)^* U^{(i)}_j \leq \mathbb{I}$ instead of equality. This version allows adversaries to be non-terminating. (By using $\leq \mathbb{I}$ instead of $= \mathbb{I}$.)

However, for non-terminating adversaries, the final bound needs to be adapted to include a non-termination factor. The inequality (4) changes to:

$$|P_{left} - P_{right}| \leq 4\sqrt{(d+1) \cdot P_{find} + d \cdot P_{nonterm}} \tag{8}$$

And the inequality (5) changes to:

$$|\sqrt{P_{left}} - \sqrt{P_{right}}| \leq 2\sqrt{(d+1) \cdot P_{find} + d \cdot P_{nonterm}} \tag{9}$$

We also show the alternative versions with non-terminating adversaries (all but the square-root version of $P^{(6)}_{right}$, where we cannot pull out the termination part from the projection).

Unfortunately, the Bures distance and the fidelity as used in the original proof [1] are not yet formalised in Isabelle. Since this would entail formalising a separate (and quite extensive) library, we opted for an alternative proof without the Bures distance and fidelity. However, this changes the final factors in the O2H theorem. Using the Bures distance and fidelity, it is possible to show the bound $2\sqrt{(d+1) \cdot P_{find}}$ for both the version with square-roots (5) and the version without (4). For our alternative proof, we can show (5) as it is, but get the following bound:

$$\begin{aligned} \left|P_{left} - P_{right}\right| &= \left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \cdot \left|\sqrt{P_{left}} + \sqrt{P_{right}}\right| \\ &\leq 2 \cdot \left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \end{aligned}$$

This adds a factor of two in the final inequality (4).

To finish the proof without the Bures distance or fidelity, we make use of the following lemma over real numbers:

Lemma 7.1. *Let M be a finite set of indices. Let t, u, v and a be functions indexed by M into the reals. Assume that $t(x) \geq 0$, $u(x) \geq 0$, $v(x) \geq 0$, $a(x) \geq 0$ and that:*

$$\forall x \in M. \left|\sqrt{t(x)} - \sqrt{u(x)}\right| \leq \sqrt{v(x)}$$

Then the following holds:

$$\left|\sqrt{\sum_{x \in M} a(x)t(x)} - \sqrt{\sum_{x \in M} a(x)u(x)}\right| \leq \sqrt{\sum_{x \in M} a(x)v(x)}$$

The proof of Lemma 7.1 can be found in the appendix.

7.2 Proof

Recall the overall proof structure by successive generalizations (Section 4). We will go through each step in the following and explain the proof steps in more detail.

Step 1: We first prove the O2H for a pure, punctured adversary with fixed H, S and z.

Lemma 7.2 (Pure O2H with punctured oracles)**.** *Fix $H : X \to Y$, $S \subseteq X$ and a random input z. Let $\mathcal{A}^H$ be a pure adversary with access to H that operates on a register M and has query depth d. Let $\mathcal{B}^{H,S}$ be an adversary that operates like $\mathcal{A}^H$ on M but has an additional counting register L. The counting register L has d bit values and is initialised by the empty state $|0\rangle$. The counting operator U_S in the i-th query is defined by:*

$$U_S(\psi \otimes |l\rangle) = \begin{cases} \psi \otimes |l\rangle & \text{if } \psi \text{ orthogonal to } \operatorname{span}|S\rangle \\ \psi \otimes |flip_i(l)\rangle & \text{if } \psi \in \operatorname{span}|S\rangle \end{cases}$$

Let ψ_{left} be the final state of $\mathcal{A}^H$ and ψ_{right} the final state for $\mathcal{B}^{H,S}$. Let $\tilde{P}_{find}$ be the probability that an element in S was queried, i.e. $\tilde{P}_{find} = \|(\mathbb{I} \otimes (\mathbb{I} - |0\rangle\langle 0|))\psi_{right}\|^2$.

Let $\tilde{P}_{nonterm}$ be the difference between the probabilities of counting with and without remembering the query placements, i.e. $\tilde{P}_{nonterm} = \|\psi_{count}\|^2 - \|\psi_{right}\|^2$ where ψ_{count} is the final state of an adversary $\mathcal{B}^{H,S}_{count}$ that simply counts the number of queries to S. Then, the pure O2H states:

$$\|\psi_{left} \otimes |0\rangle - \psi_{right}\|^2 \leq (d+1)\tilde{P}_{find} + d\tilde{P}_{nonterm} \tag{10}$$

Note that the non-termination part $\tilde{P}_{nonterm}$ is zero if the adversary is terminating. The formal proof closely follows the pen-and-paper version [1, Lemma 5, p.18] but extends the proof to add the non-termination part.

Proof. We give a short idea of the proof. The most important trick is to insert an intermediate adversary $\mathcal{B}^{H,S}_{count}$ that counts only the number of queries to S. We define $\mathcal{B}^{H,S}_{count}$ to be similar to $\mathcal{B}^{H,S}$ but with a different counting operator U'_S on a counting register C (where C can be represented by the space $\mathbb{C}^{\{0,\dots,d\}}$). Then the counting operator U'_S is defined by:

$$U'_S\big(\psi \otimes |c\rangle\big) = \begin{cases} \psi \otimes |c\rangle & \text{if } \psi \text{ orthogonal to } \operatorname{span}|S\rangle \\ \psi \otimes |c+1 \bmod d+1\rangle & \text{if } \psi \in \operatorname{span}|S\rangle \end{cases}$$

Let ψ_{count} be the final state of $B^{H,S}_{count}$. Then we can split up:

$$\psi_{count} = \sum_{i\in\{0,\dots,d\}} \psi'_i \otimes |i\rangle_C$$

Using the linear map N' defined by $N'(|x\rangle \otimes |y\rangle_C) = |x\rangle \otimes |0\rangle_C$, we get:

$$\psi_{left} = \sum_{i\in\{0,\dots,d\}} \psi'_i$$

Similarly, we split the right state:

$$\psi_{right} = \sum_{l\in\{0,1\}^d} \psi_l \otimes |l\rangle_L$$

With the projection on S, we get that $\psi_0 = \psi'_0$. Then:

$$\|\psi_0\|^2 = \|\psi_{right}\|^2 - \tilde{P}_{find}$$

$$\|\psi'_0\|^2 = \|\psi_{count}\|^2 - \tilde{P}_{find}$$

We then have:

$$\sum_{\substack{l\in\{0,1\}^d\\ l\neq 0}} |\psi_l \otimes |l\rangle_L\|^2 = \tilde{P}_{find} \tag{11}$$

We set $\tilde{P}_{nonterm} = \|\psi_{count}\|^2 - \|\psi_{right}\|^2$ and get:

$$\sum_{i\in\{1,\dots,d\}} \|\psi'_i \otimes |i\rangle_C\|^2 = \tilde{P}_{nonterm} + \tilde{P}_{find} \tag{12}$$

In the final calculation, we get:

$$\begin{aligned}
&\|\psi_{left} \otimes |0\rangle_L - \psi_{right}\|^2 = \\
=&\left\|(\psi_{left} - \psi_0) \otimes |0\rangle_L - \sum_{\substack{l\in\{0,1\}^d\\ l\neq 0}} \psi_l \otimes |l\rangle_L\right\|^2 = \\
=&\|(\psi_{left} - \psi_0)\|^2 + \sum_{\substack{l\in\{0,1\}^d\\ l\neq 0}} \|\psi_l \otimes |l\rangle_L\|^2 = \\
=&\left\|\sum_{i\in\{1,\dots,d\}} \psi'_i \otimes |i\rangle_C\right\|^2 + \tilde{P}_{find} \\
\leq& d \cdot \sum_{i\in\{1,\dots,d\}} \|\psi'_i \otimes |i\rangle_C\|^2 + \tilde{P}_{find} \\
=& d(\tilde{P}_{nonterm} + \tilde{P}_{find}) + \tilde{P}_{find}
\end{aligned}$$

In the first equation, we split the ψ_{right} into ψ_0 and the rest and rearrange the terms. In the second equation, we split up the norms into norms of orthogonal parts. Then, we use $\psi_0 = \psi'_0$ and (11) to rewrite. In the next step, we use the arithmetic-quadratic-mean inequality and finish using (12). □

Step 2: In the second step, we take the transition from Hilbert space vectors to operators (from a pure state ψ to the operator $\psi\psi^*$). Furthermore, the operators we use are trace-class (i.e., the trace of the operators converges), so we use the type `trace_class` in Isabelle.

The most important change in this step is the change from norms to traces, introducing a square root in the final inequality.

Example 7.3. Let ψ be a pure state with corresponding operator $\rho = \psi\psi^*$. Then we have $\operatorname{tr}\rho = \|\psi\|^2$.

Lemma 7.4. *In the setting of Lemma 7.2, let ρ_{left} be the operator corresponding to ψ_{left}. Similarly, ρ_{right} corresponds to ψ_{right}, P_{find} to $\tilde{P}_{find}$ and $P_{nonterm}$ to $\tilde{P}_{nonterm}$. Let P_M be the projective measurement of the adversary outcome (i.e. $P_M(\rho) = \operatorname{tr}(P\rho)$ for a projection P). Then we have:*

$$\begin{aligned}
&\left|\sqrt{P_M(\rho_{left} \otimes |0\rangle\langle 0|)} - \sqrt{P_M(\rho_{right})}\right| \\
\leq&\sqrt{(d+1)P_{find} + dP_{nonterm}}
\end{aligned}$$

Proof. We calculate:

$$\begin{aligned}
&|\sqrt{P_M(\rho_{left} \otimes |0\rangle\langle 0|)} - \sqrt{P_M(\rho_{right})}| \\
=&|\ \|P(\psi_{left} \otimes |0\rangle)\| - \|P\psi_{right}\|\ | \\
\leq&\|P(\psi_{left} \otimes |0\rangle - \psi_{right})\| \\
\leq&\|P\| \cdot \|\psi_{left} \otimes |0\rangle - \psi_{right}\| \\
\leq&\sqrt{(d+1)\tilde{P}_{find} + d\tilde{P}_{nonterm}} \\
=&\sqrt{(d+1)P_{find} + dP_{nonterm}}
\end{aligned}$$

In the first equality, we take advantage of Example 7.3. Then, we use the triangle inequality and homogeneity of the norm. In the last inequality, we use $\|P\| \leq 1$ since P is a projection and Lemma 7.2. Lastly, $\tilde{P}_{find}$ and P_{find} are the same probability (respectively also $\tilde{P}_{nonterm}$ and $P_{nonterm}$). □

Step 3: Since we worked with a fixed set S and function H so far, we extend the result to an expectation over S and H over some discrete distribution. The distribution must be provided in the context of the O2H. This is formalised in the locale `mixed_o2h` in Isabelle (see supplementary material).

Lemma 7.5. *In the setting of Lemma 7.4, we fix a (discrete) distribution D over H, S and a randomised, additional input z. We denote by $p_{(H,S,z)}$ the probability, that H, S and z are drawn from the distribution D. Furthermore, we denote by $\rho^{H,S,z}_{left}$, $\rho^{H,S,z}_{right}$, $P^{H,S,z}_{find}$ and $P^{H,S,z}_{nonterm}$ the corresponding values for fixed H, S and z. Let ρ_{left}, ρ_{right}, P_{find} or $P_{nonterm}$ be defined as the estimations of the corresponding fixed values over the distribution D. That is: $X = \sum_{(H,S,z)\in carrier(D)} p_{(H,S,z)} X^{H,S,z}$ where X stands for ρ_{left}, ρ_{right}, P_{find} or $P_{nonterm}$. Then the O2H over the estimations also holds:*

$$\left|\sqrt{P_M(\rho_{left} \otimes |0\rangle\langle 0|)} - \sqrt{P_M(\rho_{right})}\right|$$
$$\leq \sqrt{(d+1)P_{find} + dP_{nonterm}}$$

Using the Lemma 7.1, we can easily prove this generalisation step, taking the carrier set of D as our finite index set.

Step 4: This generalisation from pure to mixed states is essential, providing an alternative proof to Ambainis, Hamburg and Unruh's work [1]. This step generalizes the adversaries from a set of unitaries $\{U_i\}_{i\in I}$ to a set of Kraus maps $\{\mathcal{E}_i\}_{i\in I}$ where each $\mathcal{E}_i$ consists of finitely many operators.

Lemma 7.6 (Finite mixed O2H with punctuation)**.** *Fix a distribution D on H, S and z. Let $\mathcal{A}^H$ be a mixed adversary with access to oracles H drawn from D that operates on a register M and has query depth d. Let $\mathcal{A}^H$ be represented by the Kraus maps $\{\mathcal{E}_i\}_{i\in\{0,\ldots,n\}}$ and assume that every Kraus map $\mathcal{E}_i$ consists of finitely many $U^{(i)}_j$. Let $\mathcal{B}^{H,S}$ be an adversary that operates like $\mathcal{A}^H$ on M but has an additional counting register L. The counting register L has d bit values and is initialised by the empty state $|0\rangle$. The counting operator U_S is defined as in Lemma 7.2. Let ρ_{left} be the final state of $\mathcal{A}^H$ and ρ_{right} the final state for $\mathcal{B}^{H,S}$ (in mean over D). Let P_{find} be the probability that an element in S was queried, i.e. $P_{find} = \mathrm{tr}((\mathbb{I}\otimes (\mathbb{I} - |0\rangle\langle 0|))\rho_{right})$. Let $P_{nonterm}$ be the probability that the adversary $\mathcal{A}^H$ does not terminate, i.e. $P_{nonterm} = \|\rho_{count}\|^2 - \|\rho_{right}\|^2$ where ρ_{count} is the final state of an adversary $\mathcal{B}^{H,S}_{count}$ that simply counts the number of queries to S. Let P_M be the projective measurement of the adversary outcome. Then, the mixed O2H for finite Kraus maps states:*

$$\left|\sqrt{P_M(\rho_{left} \otimes |0\rangle\langle 0|)} - \sqrt{P_M(\rho_{right})}\right|$$
$$\leq \sqrt{(d+1)P_{find} + dP_{nonterm}}$$

Proof. The proof idea is to consider the adversary Kraus maps as linear combinations of many pure adversaries and then apply Lemma 7.1.

For this, we rewrite the adversarial run ρ_{left} and ρ_{right} as the application of one single Kraus map. We can then show that the calculations on the register M are the same and that the Kraus maps for $\mathcal{A}^H$ and $\mathcal{B}^{H,S}$ both have the index set $I' := I_0 \times \cdots \times I_n$. Since the I_i are all finite, so is I'. The operators of the rewritten $\mathcal{A}^H$ are denoted by A_i (similarly B_i for the adversary $\mathcal{B}^{H,S}$) for $i \in I'$. Therefore, we can apply the Lemma 7.1 on the index set I' to show:

$$\left|\sqrt{\sum_{i\in I'} A_i \rho_{init} A_i^*} - \sqrt{\sum_{i\in I'} B_i \rho_{init} B_i^*}\right|$$
$$\leq \sqrt{\sum_{i\in I'} (d+1)P^{(i)}_{find} + dP^{(i)}_{nonterm}}$$

As an assumption to Lemma 7.1, we use the Lemma 7.5 on every pure adversary defined by A_i and B_i. As a last step, we have to make sure that rewriting the definitions of P_{find} and $P_{nonterm}$ also yields the index set I' and that we can write it in the above form. This finishes the proof. □

Step 5: In this step, we generalise to adversaries represented by arbitrary Kraus maps. In explicit, the adversarial Kraus maps may now contain infinitely many operators.

Lemma 7.7. *The Lemma 7.6 also holds for adversaries $\mathcal{A}^H$ represented by arbitrary Kraus maps $\mathcal{E} := \{\mathcal{E}_i\}_{i\in\{0,\ldots,d\}}$.*

Proof. We need to show that the final inequality still holds when taking a limit. Indeed, we have to define a limit process to show that the adversarial run with arbitrary Kraus maps converges. As the adversary consists of $d+1$ Kraus maps, we can consecutively build the limit on each Kraus map using an induction. For one Kraus map, we consider the filter generated by all finite subsets. As the number of operators in a Kraus family must still be countable to suffice our summability notion, the set of finite subsets is dense. Therefore, we can calculate the limit on finite subsets. For a Kraus map $\mathcal{E}$ with operators $\{U_i\}_{i\in I}$ (where I is countable), we will write $\mathcal{F} \in$ subadv($\mathcal{E}$) to say that $\mathcal{F}$ is represented by $\{U_i\}_{i\in J}$ for a finite $J \subseteq I$ ("subadv" indicates that $\mathcal{F}$ is a sub-adversary of $\mathcal{E}$). Then we consider the limit on property P:

$$P(\mathcal{F}) \xrightarrow{\mathcal{F}\in\ \mathrm{subadv}(\mathcal{E})} P(\mathcal{E})$$

We extend the notion of sub-adversaries to adversaries with finitely many Kraus maps inductively. Now, we can prove that the limit also applies to the adversarial runs with the adversary $\mathcal{E} = \{\mathcal{E}_i\}_{i\in\{0,\dots,d\}}$:

$$\rho_{left}(\mathcal{F}) \xrightarrow{\mathcal{F}\in\ \mathrm{subadv}(\mathcal{E})} \rho_{left}(\mathcal{E})$$

$$\rho_{right}(\mathcal{F}) \xrightarrow{\mathcal{F}\in\ \mathrm{subadv}(\mathcal{E})} \rho_{right}(\mathcal{E})$$

By first defining a more general adversarial run that takes the adversary and its query depth as input, we then prove the limit using induction on the query depth.

In the last step, we prove that taking the limit composes with measuring and taking the trace. Finally, we have:

$$\left|\sqrt{P_M(\rho_{left}(\mathcal{F}))} - \sqrt{P_M(\rho_{right}(\mathcal{F}))}\right| \xrightarrow{\mathcal{F}\in\ \mathrm{subadv}(\mathcal{E})} \left|\sqrt{P_M(\rho_{left}(\mathcal{E}))} - \sqrt{P_M(\rho_{right}(\mathcal{E}))}\right|$$

Similarly, we also get the limit for the find and non-termination probabilities:

$$\sqrt{(d+1)P_{find}(\mathcal{F}) + dP_{nonterm}(\mathcal{F})} \xrightarrow{\mathcal{F}\in\ \mathrm{subadv}(\mathcal{E})} \sqrt{(d+1)P_{find}(\mathcal{E}) + dP_{nonterm}(\mathcal{E})}$$

Finally, we can show the inequality

$$\left|\sqrt{P_M(\rho_{left}(\mathcal{E}))} - \sqrt{P_M(\rho_{right}(\mathcal{E}))}\right| \leq \sqrt{(d+1)P_{find}(\mathcal{E}) + dP_{nonterm}(\mathcal{E})}$$

using Lemma 7.6 on adversaries with finite Kraus maps. □

Step 6: Using the punctured O2H in Lemma 7.7, we can finally prove all the different versions of the O2H given by the definitions of P_{right} in Definition 6.4.

The proof of Theorem 3.1 still requires an additional step. Up to now, we have always estimated the difference between an adversary and its punctuation. In Theorem 3.1, we consider two adversaries with two oracle functions H and G that differ only in the change set. The idea here is very simple: First, we show that the probabilities for punctuations with H and G are the same.

$$P_M(\rho_{right}(\mathcal{A}^H)) = P_M(\rho_{right}(\mathcal{A}^G)) \tag{13}$$

Second, we can split the O2H inequality into two punctuation inequalities. Formally, we have:

$$\begin{aligned}
&\left|\sqrt{P_M(\rho_{left}(\mathcal{A}^H))} - \sqrt{P_M(\rho_{left}(\mathcal{A}^G))}\right| \\
=&\Big|\sqrt{P_M(\rho_{left}(\mathcal{A}^H))} - \sqrt{P_M(\rho_{right}(\mathcal{A}^H))} \\
&\quad + \sqrt{P_M(\rho_{right}(\mathcal{A}^G))} - \sqrt{P_M(\rho_{left}(\mathcal{A}^G))}\Big| \\
\leq&\left|\sqrt{P_M(\rho_{left}(\mathcal{A}^H))} - \sqrt{P_M(\rho_{right}(\mathcal{A}^H))}\right| \\
&\quad + \left|\sqrt{P_M(\rho_{right}(\mathcal{A}^G))} - \sqrt{P_M(\rho_{left}(\mathcal{A}^G))}\right| \\
\leq&\sqrt{(d+1)P_{find}(\mathcal{A}^H) + dP_{nonterm}(\mathcal{A}^H)} \\
&\quad + \sqrt{(d+1)P_{find}(\mathcal{A}^G) + dP_{nonterm}(\mathcal{A}^G)}
\end{aligned}$$

If the adversary is terminating, we can conclude that $P_{nonterm} = 0$ and that $P_{find}(\mathcal{A}^H) = P_{find}(\mathcal{A}^G)$, yielding the additional factor of 2 in the final Theorem 3.1.

The alternative versions can be shown using different projective measurements at the end. Let P be the final measurement of the adversary $\mathcal{A}^H$ on the register M. Then, the proof of the alternative version uses the following:

- For $P^{(2)}_{right}$, we use the punctured O2H with projection $Q_2 = P \otimes \mathbb{I}$.
- For $P^{(3)}_{right}$, we use the punctured O2H with projection $Q_3 = P \otimes |0\rangle\langle 0|$.
- For $P^{(4)}_{right}$, we use the punctured O2H with projection Q_3 and equation (13).
- For $P^{(5)}_{right}$, we use the punctured O2H with projection $Q_5 = P \otimes |0\rangle\langle 0| + \mathbb{I} \otimes (\mathbb{I} - |0\rangle\langle 0|)$.
- For $P^{(6)}_{right}$, we use the punctured O2H with projection Q_5 and equation (13).

This concludes the proof of Theorem 3.1 and its alternative versions.

7.3 Challenges during Formalization

The main challenge during formalisation arises when the pen-and-paper proof uses concepts or results that still need to be formalised. For this project, the original proof [1] uses Bures distances and the fidelity of quantum states, which have yet to be formalised in Isabelle. As we would also need several lemmas and theorems on these concepts, the formalisation would have taken us too long. Therefore, we chose a different path: finding an alternative proof. Fortunately, we could reuse the formalisation of quantum registers [35], of complex bounded operators [15] and of tensor products on Hilbert spaces [37].

Another essential dependency is the development of Kraus maps in Isabelle. We build on previous work by Unruh from the qrhl-tool [34, 38]. We extend the formalisation of Kraus maps to suffice the needs of our proofs. For example, we introduce an extension of Kraus maps by tensoring the identity to all operators. This implements the need for an additional counting register where the Kraus map works as the identity. Furthermore, we show basic properties of this new notion.

There have also been smaller formalisation-specific issues. For example, having several layers of types, such as types for a Hilbert space vector, operators, and trace-class operators, we need to restate lemmas in different types and show that the type-liftings are all fulfilled. Another issue with types is the embedding of reals in the complex numbers. For example, we know that the probability outcomes are all real, but the measurement is taken by the trace function — which outputs a complex number in Isabelle. Therefore, we always need explicit type-casts, such as taking the real parts of complex numbers and showing compatibility with all calculations.

Another interesting aspect of formalisation is that we need to state definitions more clearly than pen-and-paper proofs. An example here is when reducing adversaries using Kraus maps to pure ones. On paper, we easily rearrange terms to see that the composition of Kraus maps can be represented by one single Kraus map. However, for the formalisation, we need to state the composed index set and operators precisely. This takes a little effort to juggle all indices into place. Another example is that we need to concretely define the counting functions and show that they all behave well.

A more mathematical question during formalisation was the summability properties for the adversary runs (and again lifting these properties through all types). On pen-and-paper, we simply write down the limits often without giving too much thought to the exact convergence arguments. Formally, we always have to ensure that Kraus maps over infinite index sets converge. Another mathematical challenge is ensuring that the outcome of pure adversaries stays pure, even if we formulate the result over operators as mixed states. This could be proven by induction on the adversary depth.

7.4 Technical Details

Our proof developments comprise several parts:

- additional foundational lemmas (0.6k loc)
- definitions of the O2H locale context (0.9k loc)
- the adversary runs with and without counting (1.6k loc)
- the pure O2H formalization (0.4k loc)
- the mixed O2H formalization including the reduction to pure adversaries , limit arguments and the Lemma 7.1 (2.3k loc)
- the final O2H Theorem 3.1 with its alternative versions (1.1k loc)

In total, our formalisation amounts to around 6.9k lines of code. Graphically, the proportions of the loc counts above can be shown as follows in Figure 1.

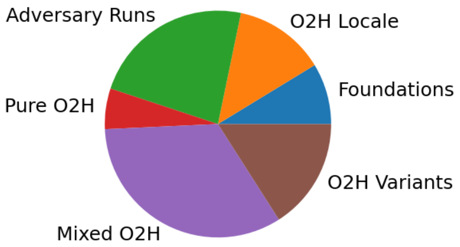


Figure 1. Distribution of lines of code on different topics

8 Outlook

To summarise our contribution, we have formalized the One-way to Hiding Theorem in Isabelle. We gave a new and alternative proof to Ambainis, Hamburg and Unruh's work [1] omitting the notions of Bures distance and fidelity and even generalized the result to possibly non-terminating adversaries. Furthermore, we described challenges, their solutions and various generalization steps during the formalization process. Moreover, we explained our approach to formalizing quantum adversaries using Kraus maps. Finally, we give the general proof ideas needed in the formalization to conclude the One-way to Hiding Theorem with several alternative formulations. We hope that this work provides essential and foundational groundwork to formalize security proofs against quantum adversaries.

In future work, we propose to continue formalizing different one-way to hiding versions, and results which establishes a concrete bound on the probability of finding a reprogrammed value, e.g. [1, Theorem 2]. The next step is to connect our formalizations with the qrhl-tool [34, 38] to back the One-way to Hiding Theorem with a complete and foundational formalization. The most challenging part here could be to align different definitions of quantum adversaries. Ultimately, the goal would be a fully verified tool to formalize and check security proofs of cryptographic primitives and protocols against quantum attackers.

A Appendix: Proof of Lemma 7.1

Here, we give the proof of Lemma 7.1.

Proof. Proof by induction on M. The base case $M = \emptyset$ is trivial. For the induction step, let $M = N \cup \{y\}$ such that $\left|\sqrt{\sum_{x\in N} a(x)t(x)} - \sqrt{\sum_{x\in N} a(x)u(x)}\right| \leq \sqrt{\sum_{x\in N} a(x)v(x)}$ holds. Let us abbreviate the sums as $t_N := \sum_{x\in N} a(x)t(x)$, $u_N := \sum_{x\in N} a(x)u(x)$ and $v_N := \sum_{x\in N} a(x)v(x)$. Then, by squaring the induction hypothesis, we have:

$$t_N + u_N \leq v_N + 2\sqrt{t_N u_N} \tag{14}$$

By squaring the assumption for y and scaling with $a(y)$, we get:

$$a(y)t(y) + a(y)u(y) \leq a(y)v(y) + 2a(y)\sqrt{t(y)u(y)} \tag{15}$$

From $\forall a, b.\ a + b \geq 2\sqrt{ab}$ and squaring, we also have

$$\begin{aligned}&\sqrt{t_N u_N} + a(y)\sqrt{t(y)u(y)}\\ \le&\sqrt{(t_N + a(y)t(y))(u_N + a(y)u(y))}\end{aligned} \tag{16}$$

Together, we calculate:

$$\begin{aligned}&\left|\sqrt{t_N + a(y)t(y)} - \sqrt{u_N + a(y)u(y)}\right|^2\\ =&\ t_N + a(y)t(y) + u_N + a(y)u(y)\\ &\quad - 2\sqrt{(t_N + a(y)t(y))(u_N a(y)u(y))}\\ \overset{(14)(15)}{\le}&\ v_N + a(y)v(y) + 2\sqrt{t_N u_N} + 2a(y)\sqrt{t(y)u(y)}\\ &\quad - 2\sqrt{(t_N + a(y)t(y))(u_N + a(y)u(y))}\\ \overset{(16)}{\le}&\ v_N + a(y)v(y)\end{aligned}$$

This proves the lemma. □

Acknowledgments

Many thanks to Tobias Nipkow for the continuous support. The first author was funded by the research training group ConVeY of the German Research Foundation under grant GRK 2428 and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – NI 491/18-1. The second author was funded by the ERC consolidator grant CerQuS (Certified Quantum Security, 819317), the Estonian Centre of Excellence in IT (EXCITE, TK148), the Estonian Centre of Excellence "Foundations of the Universe" (TK202) and the Estonian Research Council PRG grant "Secure Quantum Technology" (PRG946).

References

[1] Andris Ambainis, Mike Hamburg, and Dominique Unruh. 2019. *Quantum Security Proofs Using Semi-classical Oracles.* Springer International Publishing, 269–295. https://doi.org/10.1007/978-3-030-26951-7_10

[2] Roberto Maria Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 30/03/2019. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 2.0).

[3] Roberto Maria Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 30/11/2017. CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation.

[4] Clemens Ballarin. 2004. Locales and Locale Expressions in Isabelle/Isar. In *Types for Proofs and Programs*, Stefano Berardi, Mario Coppo, and Ferruccio Damiani (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 34–50.

[5] Manuel Barbosa, Gilles Barthe, Xiong Fan, Benjamin Grégoire, Shih-Han Hung, Jonathan Katz, Pierre-Yves Strub, Xiaodi Wu, and Li Zhou. 2021. EasyPQC: Verifying Post-Quantum Cryptography. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 2564–2586.

[6] Gilles Barthe, François Dupressoir, Benjamin Grégoire, César Kunz, Benedikt Schmidt, and Pierre-Yves Strub. 2014. *EasyCrypt: A Tutorial.* Springer International Publishing, 146–166. https://doi.org/10.1007/978-3-319-10082-1_6

[7] David A. Basin, Andreas Lochbihler, and S. Reza Sefidgar. 2020. CryptHOL: Game-Based Proofs in Higher-Order Logic. *Journal of Cryptology* 33, 2 (Jan. 2020), 494–566.

[8] Mihir Bellare and Phillip Rogaway. 1993. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (Fairfax, Virginia, USA) *(CCS '93)*. Association for Computing Machinery, New York, NY, USA, 62–73. https://doi.org/10.1145/168588.168596

[9] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. 2019. *Tighter Proofs of CCA Security in the Quantum Random Oracle Model.* Springer International Publishing, 61–90. https://doi.org/10.1007/978-3-030-36033-7_3

[10] Bruno Blanchet. 2009. A Computationally Sound Mechanized Prover for Security Protocols. *Dependable and Secure Computing, IEEE Transactions on* 5 (01 2009), 193 – 207. https://doi.org/10.1109/TDSC.2007.1005

[11] Bruno Blanchet. 2024. CryptoVerif: Cryptographic protocol verifier in the computational model. url-https://bblanche.gitlabpages.inria.fr/CryptoVerif/, accessed: 2024-07-17.

[12] Bruno Blanchet and Charlie Jacomme. 2024. Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges. In *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*. IEEE Computer Society, Los Alamitos, CA, USA, 515–528.

[13] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. 2011. *Random Oracles in a Quantum World.* Springer Berlin Heidelberg, 41–69. https://doi.org/10.1007/978-3-642-25385-0_3

[14] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 2018. CRYSTALS — Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy.* 353–367.

[15] José Manuel Rodríguez Caballero and Dominique Unruh. 2021. Complex Bounded Operators. *Archive of Formal Proofs* (September 2021). https://isa-afp.org/entries/Complex_Bounded_Operators.html, Formal proof development.

[16] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. 2019. Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability. Cryptology ePrint Archive, Paper 2019/428. https://eprint.iacr.org/2019/428

[17] Edward Eaton. 2017. *Leighton-Micali Hash-Based Signatures in the Quantum Random-Oracle Model.* Springer International Publishing, 263–280. https://doi.org/10.1007/978-3-319-72565-9_13

[18] Manuel Eberl et al. 2024. Archive of Formal Proofs. https://www.isa-afp.org/, accessed: 2024-06-21.

[19] National Institute for Standards and Technology: computer security resource center. 2024. Post-Quantum Cryptography. https://csrc.nist.gov/projects/post-quantum-cryptography, accessed: 2024-09-11.

[20] GitHub. 2022. EasyCrypt. https://github.com/EasyCrypt/easycrypt, accessed: 2024-07-26.

[21] Katharina Heidler and Dominique Unruh. 2024. One-way to Hiding Formalization – Formalizing the O2H Theorem in Isabelle. https://doi.org/10.5281/zenodo.14278513 last accessed: 2024-12-04, to be published at AFP.

[22] A. S. Holevo. 2011. Entropy gain and the Choi-Jamiolkowski correspondence for infinite-dimensional quantum evolutions. *Theoretical and Mathematical Physics* 166, 1 (Jan. 2011), 123–138. https://doi.org/10.1007/s11232-011-0010-5

[23] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. 2018. *IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited.* Springer International Publishing, 96–125. https://doi.org/10.1007/978-3-319-96878-0_4

[24] Florian Kammüller, Markus Wenzel, and Lawrence Paulson. 1999. Locales A Sectioning Concept for Isabelle. 839–839. https://doi.org/10.1007/3-540-48256-3_11

[25] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shi-Feng Sun. 2020. *Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security.* Springer International Publishing, 703–728. https://doi.org/10.1007/978-3-030-45727-3_24

[26] Andreas Lochbihler. 2017. CryptHOL. *Archive of Formal Proofs* (May 2017). https://isa-afp.org/entries/CryptHOL.html, Formal proof development.

[27] Technische Universität München and Cambridge University. 2022. Isabelle. https://isabelle.in.tum.de/index.html, accessed 2024-07-26.

[28] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press.

[29] Tobias Nipkow and Gerwin Klein. 2014. *Concrete Semantics with Isabelle/HOL*. Springer. http://concrete-semantics.org.

[30] Tobias Nipkow, Lawrence Paulson, and Markus Wenzel. 2002. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic.* LNCS, Vol. 2283. Springer.

[31] Dominique Unruh. 2014. *Quantum Position Verification in the Random Oracle Model.* Springer Berlin Heidelberg, 1–18. https://doi.org/10.1007/978-3-662-44381-1_1

[32] Dominique Unruh. 2014. *Revocable Quantum Timed-Release Encryption.* Springer Berlin Heidelberg, 129–146. https://doi.org/10.1007/978-3-642-55220-5_8

[33] Dominique Unruh. 2015. *Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model.* Springer Berlin Heidelberg, 755–784. https://doi.org/10.1007/978-3-662-46803-6_25

[34] Dominique Unruh. 2018. Quantum Relational Hoare Logic. *Proceedings of the ACM on Programming Languages* 3 (02 2018), 1–31. https://doi.org/10.1145/3290346

[35] Dominique Unruh. 2021. Quantum and Classical Registers. *Archive of Formal Proofs* (October 2021). https://isa-afp.org/entries/Registers.html, Formal proof development.

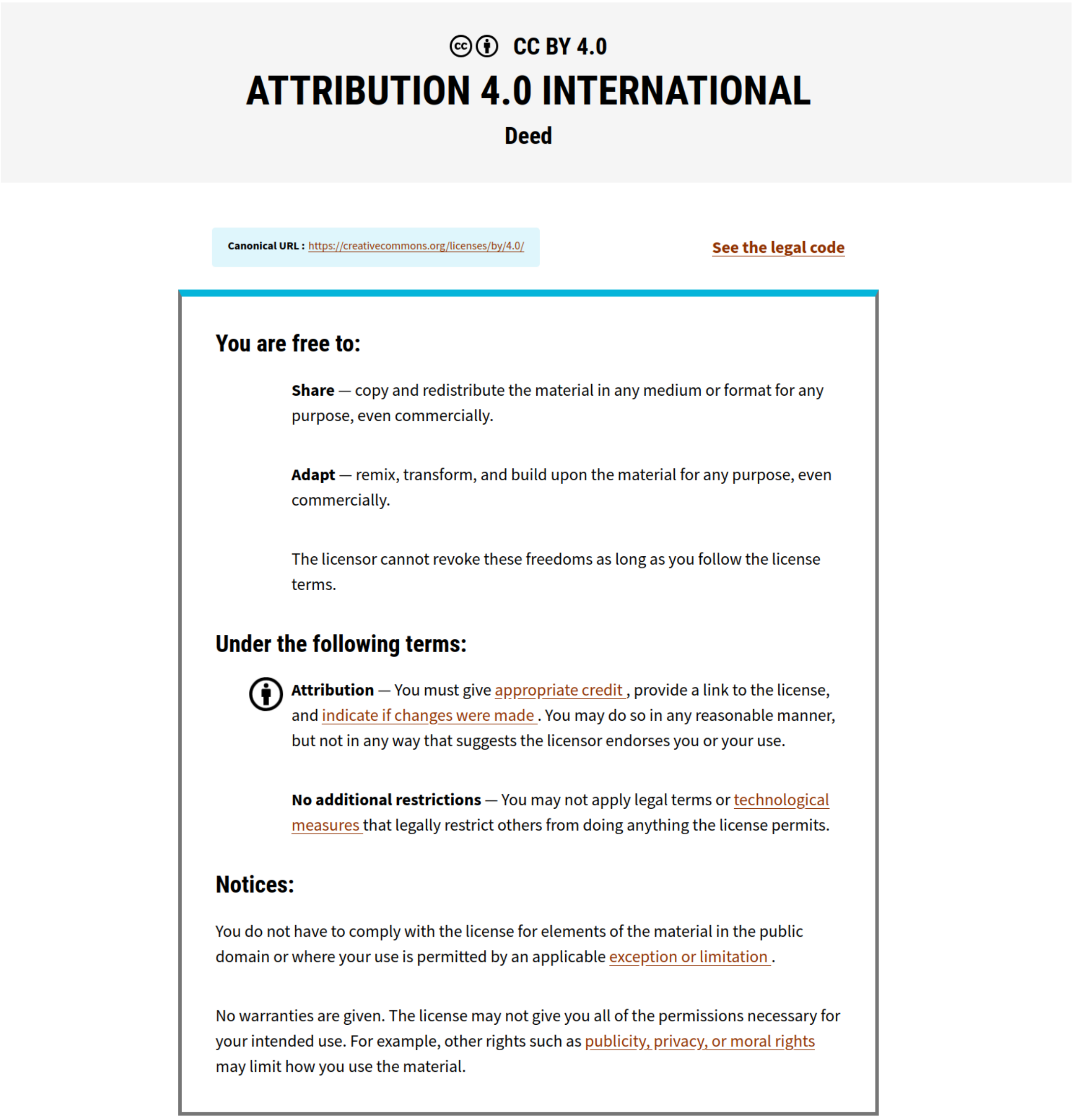
[36] Dominique Unruh. 2021. Quantum references. https://doi.org/10.48550/ARXIV.2105.10914

[37] Dominnique Unruh. 2024. Hilbert Space Tensor Product. https://github.com/dominique-unruh/afp/tree/unruh-edits/thys/Hilbert_Space_Tensor_Product, accessed: 2024-09-17, formal proof development.

[38] Dominique Unruh. 2024. qrhl-tool. https://github.com/dominique-unruh/qrhl-tool, accessed: 2024-07-16.

[39] John Watrous. 2018. *The Theory of Quantum Information.* Cambridge University Press.

Received 2024-09-10; accepted 2024-11-19



CC BY 4.0

ATTRIBUTION 4.0 INTERNATIONAL

Deed

Canonical URL : https://creativecommons.org/licenses/by/4.0/

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Figure C.1: Image source: CreativeCommons. `https://creativecommons.org/licenses/by/4.0/`, accessed on 2025-01-14