Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Concrete IND-CCA Security of NIST PQC KEMs in the ROM

Bachelor Thesis

M. Himmelberger

August 26, 2022

Advisors: Prof. Dr. Kenny Paterson, V. Maram

Applied Cryptography Group
Institute of Information Security
Department of Computer Science, ETH Zürich

Abstract

Public-Key Cryptography is at the heart of today's internet but with the advent of quantum-computing on the horizon, new encryption schemes are needed to ensure continued confidentiality of our data. The National Institute of Standards and Technology (NIST), at the time of writing, is in the process of selecting and standardizing post-quantum algorithms for this purpose. Many of the finalists employ variants of the FO transformation presented in [FO13] as their constructions. However, most proposals provide only little in terms of comprehensive proofs for the concrete classical security of their suggested schemes (e.g. their security in the ROM) and their changes compared to [FO13] are often not discussed in detail.

In this work, we provide security reductions that show how the security of the NIST round 3 finalists in the KEM category can be bounded in terms of their underlying hardness assumptions. We employ the modular framework for FO constructions from [HHK17] to describe the finalists' constructions in detail. We then provide concrete, stand-alone and tight reduction proofs in the ROM for all of the round 3 finalists NTRU, Classic McEliece, CRYSTALS-KYBER and SABER. We also give an overview of our assumptions and the obstacles we encountered in each case and we compare our results to the respective specification documents. Lastly, we slightly improve a bound in one of the theorems in [HHK17] without requiring any additional assumptions.

These proofs serve as a first step towards formally verifying the security proofs for NIST finalists. They also give us confidence in the confidentiality provided by finalists' constructions given their underlying hardness assumptions. And finally, we hope that many of our methods and results can be reused in order to analyze other KEM constructions in the ROM (such as other NIST candidates) in a similar fashion.

Acknowledgement

I would like to extend my thanks first and foremost to my supervisors Varun Maram and Prof. Dr. Kenny Paterson for helping me find and tackle a topic that I previously had no experience with. Every meeting helped me to improve my thesis and I have learned a lot in the process.

Further, I want to thank Dr. Kathrin Hövelmanns to whose input I owe the idea behind the reductions for CRYSTALS-KYBER and SABER and who has helped me understand the previous research in this space.

Last but not least, I thank my friends, my family, my girlfriend and our cat for helping me to stay motivated and for supporting me in countless ways during all of my studies so far.

Contents

Chapter 1

Introduction

Ours is a digital society: No matter whether we are browsing the web, using electronic banking, texting our friends or trading crypto currencies, most of us are able to enjoy a global and secure web. An interconnected world like this would be unthinkable without modern cryptography at its core. In all of the examples above, cryptography secures our communication against a third party modifying our messages, it hides the contents of the messages behind encryption and it is used to authenticate who we are talking to. It is safe to say that cryptography is essential to our society today.

However, as technology advances and with quantum computing on the horizon, cryptography's very foundation is being challenged. Problems that were believed to be infeasible for any realistic number of classical computers seem in reach of sufficiently large quantum computers and the assumptions underlying the "unbreakability" of modern cryptography are being called into question.

Shor's algorithm [Sho99] is a well-known quantum algorithm that poses a risk to all asymmetric cryptography schemes relying on the discrete logarithm problem or on the factorization of big integers (see [MVDJ18] for an overview). This includes the widely used RSA scheme, which underlies much of the Internet's protocols, as well as more recent schemes like ECDSA employing elliptic curves to digitally sign messages. Another quantum algorithm, Grover's algorithm [Gro96], forces users of most symmetric cryptography schemes to approximately double the size of their keys to achieve the same level of security as before.

Recently, researchers have been focusing more and more on studying the short-comings of existing algorithms and on finding alternatives that are secure even under the presence of adversaries with access to quantum computers. The National Institute of Standards and Technology (NIST) has gone on the record saying that while this is not as big of an issue for symmetric primitives like AES (with sufficiently large keys) or for the security of modern hash functions like SHA-256 or SHA-3, quantum computers pose a much more existential threat to asymmetric schemes like RSA, ECDSA, ECDH and DSA, which need to be replaced by secure schemes in the near future [CJL$^+$16].

In an effort to establish suitable so-called ”post-quantum” cryptographic primitives, NIST has announced a standardization process in late 2016 in order to find suitable algorithms for digital signatures, public-key encryption (PKE), as well as key encapsulation. The process proceeds similar to a competition where applicants submit proposals for algorithms to be standardized that can then be analyzed by the academic community before NIST selects a subset of algorithms to advance to the subsequent round.

At the time of starting this thesis, the competition had reached Round 3 [AASA$^+$20] and there were 4 finalists left in the category for public-key encryption and key encapsulation. These were Classic McEliece [ABC$^+$20], CRYSTALS-KYBER [SAB$^+$20], NTRU [CDH$^+$20] and SABER [DKR$^+$20].

These proposed algorithms in the PKE category do not directly tackle secure encryption of arbitrarily large messages as would be desirable. Instead however, confidential communication can be established using the so-called KEM-DEM composition as seen in [CS03]: One party uses a key encapsulation mechanism (KEM) to generate a fixed-length key and then securely transmits it to the other party. Using this shared secret key, these parties can now use a faster symmetric encryption scheme to encrypt variable-length messages securely. This so-called hybrid encryption is used in many modern protocols such as TLS, SSH, OpenPGP and PKCS #7.

Towards this goal, all of the finalists above specified a weakly secure PKE scheme (for fixed-length messages) and then used variants of the Fujisaki-Okamoto (FO) transformation [FO13] to create a KEM from their PKE scheme. This KEM is then shown to be strongly secure in an idealized model known as the so-called random oracle model (ROM).

1.1 Our Work

Our thesis focuses on exactly how the finalists’ transformations to their underlying PKE schemes differ from the FO transformation. It also identifies complications to the security proofs that may have been overlooked in the NIST proposals. We then proceed to re-create the exact transformations used in the proposals within the framework of the modular FO transformations provided by [HHK17] and finally, we prove the security of the NIST Round 3 finalists in the ROM under appropriate assumptions. In the process, we also slightly improve upon a bound for the security guarantees of one of the FO transformations in [HHK17].

It is worth noting that the ROM does not correspond to the security of these algorithms against quantum computers. Instead the ROM only captures the security against classical computers[1].

Nonetheless, our work is relevant to the current discussion, because we provide an overview of the exact PKE to KEM transformations used in each proposal and it is important that we can also be confident in the security of these algorithms in a world without quantum computers. The four finalists' proposals, at the time of writing, do not provide sufficiently formal proofs of their security in the ROM.

As an example, the schemes SABER and CRYSTALS-KYBER only cite theorems from [HHK17] and argue that their transformation is therefore secure in the ROM. Meanwhile, they neglect the fact that the transformations used by both schemes are not merely compositions of the FO transformations presented in [HHK17]. Extra steps and additional reductions are needed to successfully deduce the security of their schemes in the way the authors claim. For both schemes, we derive a significantly different bound which also involves the γ-spreadness of the underlying PKE schemes, essentially a measure of the entropy of ciphertexts generated by the probabilistic encryption algorithm. This parameter was absent from both bounds presented in the respective specifications.

The structure of this thesis is as follows: In Chapter 2, we define the relevant concepts and give the security definitions that are relevant for the remainder of the thesis. Chapter 3 examines the four finalists in sequence and provides reductions proving IND-CCA security for each. In Section 3.4, we summarize our findings and contrast them with the content of the respective specification documents. The improvement upon one of the theorems in [HHK17] as well as its proof can be found in Theorem 3.8 and Appendix A respectively.

1.2 Open Questions and Round 4

This thesis only focuses on the confidentiality of the schemes in the ROM, i.e. their ability to hide the content of encrypted data in a classical setting. It is important to also study other related notions such as anonymity and robustness which play a crucial role in upcoming technologies such as anonymous cryptocurrencies and electronic voting systems. Recent work in

[1] The ROM technically also allows for quantum computers which use only classical queries to the random oracle (but can still use quantum computation locally). Because, however, the hash function used to implement the random oracle in practice is public knowledge, the adversary could try to implement it using a quantum circuit. Therefore, this specific case of adversaries using quantum computation locally but not for random oracle queries is somewhat unrealistic. That is why research focuses on a less restrictive model (the so-called QROM) that also allows quantum queries to the oracle. This newer model is then used to prove the post-quantum security of these algorithms.

this direction has been conducted in [GMP22] and [Xag22]. Further, all of these notions can also be studied in the more powerful quantum random oracle model.

In addition, our goal is only to prove the security of the KEM schemes under the assumption that the underlying PKE schemes are secure. Whether or not that is justified and how secure these schemes are in concrete terms, we also leave open.

During the final two months of writing this thesis, NIST has closed the third round of the competition after very nearly two years of active research into the current proposals [AAC+22]. From the four round 3 finalists we considered, NIST has selected CRYSTALS-KYBER for standardization. Classic McEliece has advanced into the fourth round, as have three of the alternate round 3 KEM candidates that we did not study in this work. The remaining finalists were no longer considered.

Chapter 2

Preliminaries

2.1 Notation

As we borrow a lot of notation from [HHK17], we briefly re-state the relevant parts here.

For a set S, $|S|$ denotes the cardinality of S. For a finite set S, we denote the sampling of a uniform random element x by $x \leftarrow_{\$} S$, while we denote the sampling according to some distribution $\mathcal{D}$ by $x \leftarrow \mathcal{D}(S)$. By $[\![B]\!]$ we denote the bit that is 1 if the Boolean Statement B is true, and otherwise 0. If E denotes a probabilistic event, then $\overline{\mathsf{E}}$ denotes its negation.

Algorithms We denote deterministic computation of an algorithm A on input x by $y := A(x)$. We denote algorithms with access to an oracle O by A^O. Unless stated otherwise, we assume all our algorithms to be probabilistic and denote the computation by $y \leftarrow A(x)$. To denote the event of algorithm A outputting y upon input of x, we write $A(x) \Rightarrow y$.

Random Oracles We will repeatedly model hash functions as random oracles. We call this idealization the *Random Oracle Model* (ROM). To keep record of the queries issued to a hash function H, we will use a hash list $\mathfrak{L}_H$ that contains all tuples $(x, H(x))$ of arguments x in the domain of H that H was queried on and the respective answers $H(x)$. We make the convention that $H(x) = \perp$ for all x not in the domain of H.

Games Following [Sho04, BR06], we use code-based games. We implicitly assume Boolean flags to be initialized to false, numerical types to 0, sets to $\emptyset$, and strings to the empty string ϵ unless otherwise noted. We make the convention that a procedure terminates once it has returned an output. Further, procedures that abort and whose output is used yield $\perp$. We also make the following convention: Whenever we use the number of the adversary's oracle queries (such as q_H), we count the total number of times that oracle is executed in the experiment. That is, the number of explicit queries to the oracle made by the adversary plus the number of implicit queries made by the experiment (such as by another oracle defined in terms of the first one).

2.2 Public-Key Encryption

Symmetric cryptography, mentioned briefly in the introduction, captures cryptographic schemes where both parties share a common key that can be used to both encrypt and decrypt messages. In this work, we study a certain kind of asymmetric cryptography, namely *public-key encryption* (PKE). Let there be a message space $\mathcal{M}$, a space of ciphertexts $\mathcal{C}$, a public key space $\mathcal{PK}$ and a private key space $\mathcal{SK}$. A PKE scheme is defined by three procedures $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ which satisfy the following:

- KGen randomly returns a pair $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{PK} \times \mathcal{SK}$ of public and private key respectively. This procedure optionally takes a so-called security parameter (denoted $1^{\mathcal{K}}$) as argument.
- Enc takes two arguments: $\mathsf{pk} \in \mathcal{PK}$ and $m \in \mathcal{M}$. Enc returns a value $c \in \mathcal{C}$, possibly in a randomized way. If necessary, we explicitly specify the randomness used by encryption as another argument $r \in \mathcal{R}$ where $\mathcal{R}$ is an appropriate randomness space for the scheme.
- Dec is a deterministic procedure taking two arguments: $\mathsf{sk} \in \mathcal{SK}$ and $c \in \mathcal{C}$. It outputs either a message $m \in \mathcal{M}$ or $\perp \notin \mathcal{M}$, a special symbol denoting decryption failure.

Determinism A PKE scheme is called *deterministic* if Enc is a deterministic procedure, i.e. for the same pair of inputs (pk, m), we will always get the same output c. We use the abbreviation DPKE for deterministic PKE schemes.

Correctness A *perfectly correct* PKE scheme is one that satisfies

$$\forall m.\ \Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m] = 1$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$ and the probability is taken over the randomness in Enc and KGen. To define partial correctness, we follow [HHK17] and rely on two games defined in Figure 2.1. We need to differentiate between PKE schemes that do not make use of a random oracle and those that do. In the latter case, the power of an adversary to find messages violating correctness might depend on the number of queries to the random oracle.

A PKE scheme that does not use a random oracle is called *δ-correct* if

$$\Pr[\mathsf{COR}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1] \leq \delta$$

```
GAME COR
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
  $m \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
  $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
  return $[\![\mathsf{Dec}(\mathsf{sk}, c) \neq m]\!]$
```

```
GAME COR-RO
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
  $m \leftarrow \mathcal{A}^{G}(\mathsf{pk}, \mathsf{sk})$
  $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
  return $[\![\mathsf{Dec}(\mathsf{sk}, c) \neq m]\!]$
```

Figure 2.1: Correctness games COR and COR-RO for a scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

for all (possibly unbounded) adversaries $\mathcal{A}$.
A PKE scheme which does use a random oracle G is called *δ-correct* if

$$\Pr[\mathsf{COR\text{-}RO}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1] \leq \delta(q_G)$$

for all (possibly unbounded) adversaries $\mathcal{A}$ issuing at most q_G queries to G.

In the second case, δ is a function while in the first case, δ is a constant. The first case can thus be viewed as a special case of the second where $q_G = 0$.

Rigidity As in [BP18], we define a DPKE scheme to be *rigid*, if for all key pairs $(pk, sk) \leftarrow \mathsf{KGen}$, and all ciphertexts c, it holds that either $\mathsf{Dec}(\mathsf{sk}, c) = \perp$ or $\mathsf{Enc}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, c)) = c$. Intuitively, rigidity captures the property that there is only at most a single ciphertext decrypting to each message while correctness talks about encrypted messages being recovered properly after decryption.

Gamma spreadness As in [FO13, HHK17], we define the min-entropy of $\mathsf{Enc}(\mathsf{pk}, m)$ by

$$\gamma(\mathsf{pk}, m) := -\log \max_{c \in \mathcal{C}} \Pr_{r \leftarrow \mathcal{R}}[c = \mathsf{Enc}(\mathsf{pk}, m; r)]$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$ and $m \in \mathcal{M}$. We say that PKE is *γ-spread* if, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$ and every message $m \in \mathcal{M}$, $\gamma(\mathsf{pk}, m) \geq \gamma$. While we won't use this definition exactly later on, we will make use of a consequence of it, namely that in a γ-spread scheme, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$, every message $m \in \mathcal{M}$ and every ciphertext $c \in \mathcal{C}$,

$$\Pr_{r \leftarrow \mathcal{R}}[c = \mathsf{Enc}(\mathsf{pk}, m; r)] \leq 2^{-\gamma}$$

Rejection A KEM scheme's decapsulation function is often defined on at least some inputs that cannot be produced by encapsulation using the corresponding public key. In these cases, the scheme may choose to notify the user of the failure by outputting $\perp$. In this case, we say that the scheme has *explicit rejection*. Alternatively, the scheme might output some $K \in \mathcal{K}$, possibly pseudo-randomly, and the failure might not be immediately obvious. These schemes are said to have *implicit rejection*.

2.3 KEM

Intuitively, while a PKE scheme is used to encrypt and decrypt arbitrary messages, a *key encapsulation mechanism* (KEM) aims to instead generate and distribute a shared key. Formally, let there be a key space $\mathcal{K}$ and a space of ciphertexts $\mathcal{C}$ as well as public and private key spaces $\mathcal{PK}$ and $\mathcal{SK}$ respectively. A KEM scheme is defined by three procedures $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$ which satisfy the following:

- KGen randomly returns a pair $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{PK} \times \mathcal{SK}$ of public and private key respectively. This procedure optionally takes a so-called security parameter (denoted $1^{\mathcal{K}}$) as argument.
- Encaps (short for encapsulation) takes only $\mathsf{pk} \in \mathcal{PK}$ as an argument and returns a pair $(K, c) \in \mathcal{K} \times \mathcal{C}$ in a randomized way.
- Decaps is a deterministic procedure taking two arguments: $\mathsf{sk} \in \mathcal{SK}$ and $c \in \mathcal{C}$. It outputs either a key $K \in \mathcal{K}$ or $\perp \notin \mathcal{K}$, a special symbol denoting decapsulation failure.

Correctness A KEM scheme can also be *perfectly correct*, namely if $\mathsf{Decaps}(\mathsf{sk}, c) = K$ for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$ and every $(K, c) \leftarrow \mathsf{Encaps}(\mathsf{pk})$. Further, we also define δ-correctness for KEM schemes completely analogously to the definition for PKE schemes but there is no "worst input" that the adversary is allowed to pick, so we opt for a definition without games.

A KEM scheme is called *δ-correct* if

$$\Pr[\mathsf{Decaps}(\mathsf{sk}, c) \neq K \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}; (K, c) \leftarrow \mathsf{Encaps}(\mathsf{pk})] \leq \delta$$

where the probability is taken over the randomness used by KGen and Encaps. Note that, as there is no game-based definition and no adversary, we do not need to consider the amount of oracle queries possibly involved in these procedures.

2.4 Security Games

We previously talked about weakly and strongly secure schemes, two notions which we will now formalize. "Weakly secure" is what we will define as $\mathsf{OW\text{-}CPA}$ or $\mathsf{IND\text{-}CPA}$ and "strongly secure" will correspond to $\mathsf{IND\text{-}CCA}$ security. In this work, we are concerned only with the confidentiality these schemes provide and we use security and confidentiality interchangeably.

PKE Security For the security of a PKE scheme, we introduce several notions to express that given a ciphertext and public key, it is hard for an adversary to reconstruct the corresponding plaintext: One-Wayness under

Chosen Plaintext Attacks (OW-CPA), One-Wayness under Plaintext Checking Attacks (OW-PCA), One-Wayness under Validity Checking Attacks (OW-VA) and One-Wayness under Plaintext and Validity Checking Attacks (OW-PCVA).

These notions differ only in the capabilities available to the adversary. In all One-Wayness games, the adversary is given a public key and an encryption of a random message as inputs in the security game and it has access to the random oracle (if any) as part of the ROM. Depending on the security notion in question, the adversary has access to these additional oracles in the OW-ATK game:

$$O_{\mathsf{ATK}} := \begin{cases} - & \mathsf{ATK} = \mathsf{CPA} \\ \mathrm{Pco} & \mathsf{ATK} = \mathsf{PCA} \\ \mathrm{Cvo} & \mathsf{ATK} = \mathsf{VA} \\ \mathrm{Pco}, \mathrm{Cvo} & \mathsf{ATK} = \mathsf{PCVA} \end{cases}$$

where Pco is a plaintext checking oracle and Cvo is a ciphertext validity oracle. Pco checks if a given plaintext ($\perp$ is not a valid input) is indeed the decryption of a given ciphertext. Cvo checks whether a given ciphertext has any valid decryption (but it may not be called on c^*).

Following [SXY18], we define these games in terms of a distribution $\mathcal{D}$ on the message space. This yields more general definitions than in [HHK17] but we can recover the more standard notions of One-Wayness by using $\mathcal{D} := \mathcal{U}$ where $\mathcal{U}$ is the uniform distribution (on messages). If $\mathcal{D}$ is missing from the notations defined below, we implicitly mean $\mathcal{D} = \mathcal{U}$.

The games and the semantics of the oracles in question are defined formally in Figure 2.2. We also define an adversary's *OW-ATK advantage* in these games (against a scheme PKE) as

$$\mathsf{Adv}\,^{\mathsf{OW\text{-}ATK}}_{\mathsf{PKE},\,\mathcal{D}}(\mathcal{A}) := \Pr[\mathsf{OW\text{-}ATK}^{\mathcal{A}}_{\mathsf{PKE},\mathcal{D}} \Rightarrow 1]$$

Another notion we define for PKE schemes is called Indistinguishability under Chosen Plaintext Attacks (IND-CPA). This is a stronger notion than OW-CPA if the message space is large enough (see [HHK17, Lemma 2.3]). The IND-CPA security game is defined in Figure 2.3 and we define an adversary's *IND-CPA advantage* (against a scheme PKE) as

$$\mathsf{Adv}\,^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \left| \Pr[\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1] - \frac{1}{2} \right|$$

```
GAME OW-ATK
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
    $m^* \leftarrow \mathcal{D}(\mathcal{M})$
    $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
    $m' \leftarrow \mathcal{A}^{O_{\mathsf{ATK}}}(\mathsf{pk}, c^*)$
    return $[\![\mathrm{Pco}(m', c^*)]\!]$

$\mathrm{Pco}(m \in \mathcal{M}, c)$
    return $[\![\mathsf{Dec}(\mathsf{sk}, c) = m]\!]$

$\mathrm{Cvo}(c \neq c^*)$
    $m := \mathsf{Dec}(\mathsf{sk}, c)$
    return $[\![m \in \mathcal{M}]\!]$
```

Figure 2.2: One-Wayness games OW-ATK for a scheme PKE $= (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, a distribution on messages $\mathcal{D}$ and ATK $\in \{\mathsf{CPA}, \mathsf{PCA}, \mathsf{VA}, \mathsf{PCVA}\}$.

```
GAME IND-CPA
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
    $b \leftarrow_{\$} \{0, 1\}$
    $(m_0^*, m_1^*, st) \leftarrow \mathcal{A}_1(\mathsf{pk})$
    $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b^*)$
    $b' \leftarrow \mathcal{A}_2(\mathsf{pk}, c^*, st)$
    return $[\![b' = b]\!]$
```

Figure 2.3: Indistinguishability game IND-CPA for a scheme PKE $= (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

KEM Security As we are concerned with creating strongly secure KEM schemes only in this work, we consider only a single notion of security for KEM schemes, namely Indistinguishability under Chosen Ciphertext Attacks (IND-CCA). This is very analogous to IND-CPA and we could define both for PKE as well as for KEM schemes but we restrict ourselves to the needed definitions only.

In this game, the adversary has access to a decapsulation oracle Decaps which may be queried on any ciphertext except c^* and returns its decapsulation. The IND-CCA security game and the semantics of the oracle are defined formally in Figure 2.4 and we define an adversary's *IND-CCA advantage* (against a scheme KEM) as

$$\mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) := \left| \Pr[\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}} \Rightarrow 1] - \frac{1}{2} \right|$$

2.5 Lemmas

We repeatedly use the following three Lemmas throughout our work without explicitly citing them as they are well-known.

```
**GAME** IND-CCA
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
$b \leftarrow_{\$} \{0, 1\}$
$(K_0^*, c^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$
$K_1^* \leftarrow_{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\mathsf{Decaps}}(\mathsf{pk}, c^*, K_b^*)$
**return** $[\![b' = b]\!]$
```

```
$\mathsf{Decaps}(c \neq c^*)$
$K := \mathsf{Decaps}(\mathsf{sk}, c)$
**return** $K$
```

Figure 2.4: Indistinguishability game IND-CCA for a scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

The first theorem was already stated in [Sho04, Lemma 1] but we choose to re-state it again for convenience:

Lemma 2.1 (Difference Lemma) *Let* $\mathsf{A}, \mathsf{B}, \mathsf{F}$ *be events defined in some probability distribution, and suppose that* $\mathsf{A} \wedge \overline{\mathsf{F}} \iff \mathsf{B} \wedge \overline{\mathsf{F}}$. *Then* $|\Pr[\mathsf{A}] - \Pr[\mathsf{B}]| \leq \Pr[\mathsf{F}]$.

We will use this lemma to argue the following: If two games only differ if some event F takes place then the change in the adversary's success probability is at most the probability of F. As we often define the two games to use the same underlying probability space (i.e. we only change the rules for computing some random variables), F is well-defined in both games and we may compute its probability in whichever game we choose.

Lemma 2.2 (Domain Separation) *Let H be a random oracle with domain A and range B and let X, Y be two disjoint subsets of A. Further, let $H_X : X \to B$ and $H_Y : Y \to B$ be domain-restrictions of H. Then, for all $x \in X, y \in Y$ and for all $b_1, b_2 \in B$ such that* $\Pr[H_Y(y) = b_2] > 0$, *we have*

$$\Pr[H_X(x) = b_1] = \Pr[H_X(x) = b_1 \mid H_Y(y) = b_2]$$

where the probabilities are taken over the possible functions H from A to B.

This is equivalent to stating that H_X and H_Y are two independent random oracles.

For the next Lemma, we use some definitions from [Yas21]: Let P and Q be probability distributions over a finite set Ω. For a distribution P over Ω and an event $A \subseteq \Omega$, we denote by $P(A)$ the probability of event A, which is equal to $\sum_{x \in A} P(x)$.

Further, we define the *statistical distance* between two distributions P and Q as:

$$\mathsf{SD}(P, Q) = \max_{A \subseteq \Omega} |P(A) - Q(A)|$$

It follows easily that we can also calculate the statistical distance as follows:

$$\mathsf{SD}(P, Q) = \frac{1}{2} \sum_{x \in \Omega} |P(x) - Q(x)|$$

Lemma 2.3 (Distribution Replacement) *Let G_A and G_B be two games where G_A employs the distribution P and G_B employs the distribution Q. Other than that, let there be no differences in the definition of the games.*

For any adversary $\mathcal{B}$, it holds that

$$\left| \Pr[G_A^{\mathcal{B}} \Rightarrow 1] - \Pr[G_B^{\mathcal{B}} \Rightarrow 1] \right| \leq SD(P, Q)$$

Chapter 3

Analysis of NIST Finalists

3.1 NTRU

3.1.1 Modified $\mathsf{U}^{\not\perp}_m$ transformation

To begin this chapter, we will discuss a modification to the transformation $\mathsf{U}^{\not\perp}_m$ from [HHK17]. Specifically, we change the distributions used to generate both m and s, allowing for more control over the resulting scheme. This transformation could also be viewed as a modified SXY transformation from [SXY18] which uses only a single random oracle H and does not do the re-encryption check.

We call this new transformation $\mathsf{DU}^{\not\perp}_m[\mathsf{DPKE}, H, \mathcal{D}]$ and parameterize it using a random oracle H, a deterministic PKE scheme DPKE and a distribution $\mathcal{D}$ on the message space of DPKE. Implicitly, the natural number ℓ is also a parameter but we omit it in the notation so as to not overload it.

The following theorem reduces the security of a KEM scheme constructed using this transformation to a non-standard notion of OW-CPA security. Because the KEM scheme generates messages not uniformly at random but instead uses the distribution $\mathcal{D}$, its security does not depend immediately

```
$\mathsf{KGen}_{\mathsf{KEM}}(1^{\mathcal{K}})$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^{\mathcal{K}})$
    $s \leftarrow_\$ \{0,1\}^\ell$
    $\overline{\mathsf{sk}} := (\mathsf{sk}, s)$
    return $(\mathsf{pk}, \overline{\mathsf{sk}})$

$\mathsf{Encaps}(\mathsf{pk})$
    $m \leftarrow \mathcal{D}(\mathcal{M})$
    $c := \mathsf{Enc}(\mathsf{pk}, m)$
    $K := H(m)$
    return $(K, c)$

$\mathsf{Decaps}(\overline{\mathsf{sk}} = (\mathsf{sk}, s), c)$
    $m' := \mathsf{Dec}(\mathsf{sk}, c)$
    if $m' \neq \perp$ then
        return $K := H(m')$
    else
        return $K := H(s, c)$
```

Figure 3.1: The transformation $\mathsf{DU}^{\not\perp}_m[\mathsf{DPKE}, H, \mathcal{D}]$ where $\mathsf{DPKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

on the OW-CPA security of the underlying PKE scheme. More precisely, its security instead depends on the underlying scheme being OW-CPA secure in a game where messages are generated using the same distribution. This ties in to our definition of $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE},\,\mathcal{D}}(\mathcal{A})$ given in the Preliminaries.

While we will not use the entire power of this theorem to analyze NTRU (because NTRU is perfectly correct), we hope that this theorem can find applications outside of our work as well.

Theorem 3.1 (DPKE rigid, OW-CPA $\overset{\text{ROM}}{\Longrightarrow}$ $\mathsf{DU}^{\not\perp}_m[\mathsf{DPKE}, H, \mathcal{D}]$ IND-CCA) *Let $\mathsf{KEM} := \mathsf{DU}^{\not\perp}_m[\mathsf{DPKE}, H, \mathcal{D}]$ in the following and assume that $(\{0,1\}^\ell \times \mathcal{C}) \cap \mathcal{M} = \emptyset$ where $\mathcal{M}$ and $\mathcal{C}$ are the message and ciphertext spaces of DPKE respectively. If DPKE is δ-correct, then so is KEM. Furthermore, assume that DPKE is rigid and does not use any random oracle, then for any IND-CCA adversary $\mathcal{B}$ against KEM issuing at most q_H queries to the random oracle H, there exists an OW-CPA adversary $\mathcal{A}$ against DPKE, such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{B}) \leq \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{DPKE},\,\mathcal{D}}(\mathcal{A}) + \frac{q_H}{2^\ell} + 2\delta$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.

Proof The proof proceeds very similar to the proofs for [HHK17, Theorem 3.4, Theorem 3.5]. We reconstruct it here as it is omitted in [HHK17]. We call the decapsulation oracle, which $\mathcal{B}$ has access to as part of the IND-CCA game, $\mathsf{Decaps}^{\not\perp}_m$.

It is easy to verify the correctness bound. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM, issuing at most q_H queries to the random oracle H. Consider the games given in Figure 3.2.

Game G_0 Since game G_0 is the original IND-CCA game,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{B})$$

Also note that $H(m, c)$ and $H(m)$ operate on different domains (i.e. $(\{0,1\}^\ell \times \mathcal{C}) \cap \mathcal{M} = \emptyset$) and are thus independent random oracles.

Game G_1 In game G_1 we make two changes. First, we raise flag QUERY and abort if $H(s, \cdot)$ is queried (lines 35 and 36). Second, we make the pseudorandom keys output by $\mathsf{Decaps}^{\not\perp}_m$ perfectly random. That is, in $\mathsf{Decaps}^{\not\perp}_m$, we replace $K = H(s, c)$ by $K = H'(c)$ if $m' = \mathsf{Dec}(\mathsf{sk}, c) = \perp$ (line 14), where H' is an internal random oracle that cannot be accessed by $\mathcal{B}$. The latter remains unnoticed by $\mathcal{B}$ unless $H(s, c)$ is queried, in which case G_1 aborts. Since $\mathcal{B}$'s view is independent of (the uniform secret) s unless G_1 aborts,

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_0^{\mathcal{B}} \Rightarrow 1]\right| \leq \frac{q_H}{2^\ell}$$

```
GAMES $G_0$-$G_3$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
$s \leftarrow_\$ \{0,1\}^\ell$
$\mathsf{sk}' := (\mathsf{sk}, s)$
$m^* \leftarrow \mathcal{D}(\mathcal{M})$
$c^* := \mathsf{Enc}(\mathsf{pk}, m^*)$
$K_0^* := H(m^*)$
$K_1^* \leftarrow_\$ \{0,1\}^n$
$b \leftarrow_\$ \{0,1\}$
$b' \leftarrow \mathcal{B}^{\mathsf{Decaps}_m^{\not\perp}, H}(\mathsf{pk}, c^*, K_b^*)$
return $[\![b' = b]\!]$
```

```
$\mathsf{Decaps}_m^{\not\perp}(c \neq c^*)$                // $G_0$-$G_1$
$m' := \mathsf{Dec}(\mathsf{sk}, c)$
if $m' = \perp$ then
    return $K := H(s, c)$                      // $G_0$
    return $K := H'(c)$                        // $G_1$
else return $K := H(m')$
```

```
$\mathsf{Decaps}_m^{\not\perp}(c \neq c^*)$                // $G_2$-$G_3$
if $\exists K$ such that $(c, K) \in \mathfrak{L}_D$
    return $K$
$K \leftarrow_\$ \mathcal{K}$
$\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$
return $K$
```

```
$H(m)$                                          // $m \in \mathcal{M}$
if $\exists K$ s.th. $(m, K) \in \mathfrak{L}_H$
    return $K$
if $m = m^*$                                    // $G_3$
    CHAL := true; abort                         // $G_3$
$c' := \mathsf{Enc}(\mathsf{pk}, m)$                       // $G_2$-$G_3$
$K \leftarrow_\$ \mathcal{K}$
if $\exists K'$ s.th. $(c', K') \in \mathfrak{L}_D$         // $G_2$-$G_3$
    $K := K'$                                   // $G_2$-$G_3$
else                                            // $G_2$-$G_3$
    $\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c', K)\}$       // $G_2$-$G_3$
$\mathfrak{L}_H := \mathfrak{L}_H \cup \{(m, K)\}$
return $K$
```

```
$H(s', c)$                      // $(s', c) \in \{0,1\}^\ell \times \mathcal{C}$
if $\exists K$ s.th. $(s', c, K) \in \mathfrak{L}_{Hs}$
    return $K$
if $s' = s$ then                                // $G_1$-$G_3$
    QUERY := true; abort                        // $G_1$-$G_3$
$K \leftarrow_\$ \mathcal{K}$
$\mathfrak{L}_{Hs} := \mathfrak{L}_{Hs} \cup \{(s', c, K)\}$
return $K$
```

Figure 3.2: Games G_0 - G_3 for the proof of Theorem 3.1

Game G_2 In game G_2, the oracles H and $\mathsf{Decaps}^{\not\perp}_m$ are modified such that $\mathsf{Decaps}^{\not\perp}_m$ does not make use of the secret key any longer. In game G_2 we will use two lists, $\mathfrak{L}_H$ and $\mathfrak{L}_D$, for bookkeeping. $(m, K) \in \mathfrak{L}_H$ indicates that H was queried on m and $H(m) = K$ holds; $(c, K) \in \mathfrak{L}_D$ indicates that $\mathsf{Decaps}^{\not\perp}_m(c) = K$ holds and either H was queried on some message m such that $c = \mathsf{Enc}(\mathsf{pk}, m)$ or $\mathsf{Decaps}^{\not\perp}_m$ was queried on c.

Let CORERR denote the event that $\mathfrak{L}_H$ contains an entry (m, K) with $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) \neq m$. Intuitively, CORERR denotes the event that a correctness error of DPKE actually occurs. We will show that the view of B is identical in games G_1 and G_2 unless CORERR happens.

To do so, we have to examine if $\mathsf{Decaps}^{\not\perp}_m$ and H handle queries consistently in game G_2. To analyze game G_2, let c be a query to $\mathsf{Decaps}^{\not\perp}_m$, and let $m' := \mathsf{Dec}(\mathsf{sk}, c)$.

We first show that before the query to $\mathsf{Decaps}^{\not\perp}_m$ on c and the query to H on m', no entry of the form (c, K) could already exist in $\mathfrak{L}_D$ yet unless CORERR happened: since neither $\mathsf{Decaps}^{\not\perp}_m$ was yet queried on c nor H was yet queried on m', existence of an entry (c, K) in $\mathfrak{L}_D$ implies that H was queried on some message $m \neq m'$ such that $\mathsf{Enc}(\mathsf{pk}, m) = c$. Hence, $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = \mathsf{Dec}(\mathsf{sk}, c) = m' \neq m$, meaning that m induces a correctness error and CORERR happened.

We will now analyze the games' behavior in the case that CORERR did not happen.

- Case 1: $m' = \perp$. Since H cannot be queried on m', the simulation of H can never add a tuple of the form (c, K) to $\mathfrak{L}_D$. Hence, querying $\mathsf{Decaps}^{\not\perp}_m$ in game G_2 will return a uniformly random key, as in Game G_1.
- Case 2: $m' \neq \perp$. In game G_1, it holds that $\mathsf{Decaps}^{\not\perp}_m(c) = H(\mathsf{Dec}(\mathsf{sk}, c))$ for all valid ciphertexts. We will now show that H in game G_2 is "patched", meaning that it is ensured $\mathsf{Decaps}^{\not\perp}_m(c) = H(m')$, for all valid ciphertexts c assuming CORERR did not happen. We distinguish two sub-cases: $\mathcal{B}$ might either first query H on m', then $\mathsf{Decaps}^{\not\perp}_m$ on c, or the other way round.
    - If H is queried on m' first, $c = \mathsf{Enc}(\mathsf{pk}, m')$ is computed correctly (because CORERR did not happen and DPKE is rigid) and since $\mathsf{Decaps}^{\not\perp}_m$ was not yet queried on c, no entry of the form (c, K) already exists in $\mathfrak{L}_D$. Therefore, besides adding $(m', K \leftarrow_\$ \mathcal{K})$ to $\mathfrak{L}_H$, H also adds (c, K) to $\mathfrak{L}_D$ in line 30, thereby defining $\mathsf{Decaps}^{\not\perp}_m(c) := K = H(m')$.

- If $\mathsf{Decaps}_m^{\not\perp}$ is queried on c first, it adds $(c, K \leftarrow_\$ \mathcal{K})$ to $\mathfrak{L}_D$, thereby defining $\mathsf{Decaps}_m^{\not\perp}(c) := K$. When queried on m' afterwards, H computes $c = \mathsf{Enc}(\mathsf{pk}, m')$ and recognizes that an entry of the form (c, K) already exists in $\mathfrak{L}_D$ in line 27. By adding (m', K) to $\mathfrak{L}_H$ and returning K, H defines $H(m') := K = \mathsf{Decaps}_m^{\not\perp}(c)$

We have shown that $\mathcal{B}$'s view is identical in both games unless a correctness error (in the form of CORERR) occurs.

$$\left|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq \Pr[\mathsf{CORERR}]$$

We can bound $\Pr[\mathsf{CORERR}]$ with a straightforward reduction to the δ-correctness of DPKE. In this reduction, an adversary on DPKE's correctness simulates Game G_1 and additionally checks for CORERR upon every $\mathsf{Decaps}_m^{\not\perp}$ and every H query. As DPKE does not use any random oracle, δ-correctness of DPKE gives us:

$$\Pr[\mathsf{CORERR}] \leq \delta$$

Game G_3 In Game G_3, we abort (with uniformly random output) immediately on the event that $\mathcal{B}$ queries H on m^*. Denote this event as CHAL. We also define the event CORERR exactly as in Game G_2. Due to the difference lemma and the argument above that $\Pr[\mathsf{CORERR}] \leq \delta$,

$$\begin{aligned}\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right| &\leq \Pr[\mathsf{CHAL}] \\ &= \Pr[\mathsf{CHAL} \wedge \overline{\mathsf{CORERR}}] + \Pr[\mathsf{CHAL} \wedge \mathsf{CORERR}] \\ &\leq \Pr[\mathsf{CHAL} \wedge \overline{\mathsf{CORERR}}] + \Pr[\mathsf{CORERR}] \\ &\leq \Pr[\mathsf{CHAL} \wedge \overline{\mathsf{CORERR}}] + \delta\end{aligned}$$

In Game G_3, $H(m^*)$ will not be given to $\mathcal{B}$; neither through a hash nor a decapsulation query, meaning bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_3^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}$$

It remains to bound $\Pr[\mathsf{CHAL} \wedge \overline{\mathsf{CORERR}}]$. To this end, we construct an adversary $\mathcal{A}$ against the OW-CPA security of DPKE under the message distribution $\mathcal{D}$, simulating G_3 for $\mathcal{B}$ as in Figure 3.3.

Assume that CORERR does not occur and note that the simulation is perfect until CHAL occurs. Furthermore, CHAL implies that $\mathcal{B}$ queried $H(m^*)$, which implies that $(m^*, K') \in \mathfrak{L}_H$ for some K'. In this case, we have $\mathsf{Enc}(\mathsf{pk}, m^*) = c^*$ (since DPKE is deterministic). We can also be sure that there is no

```
$\mathcal{A}(\mathsf{pk}, c^*)$
$K^* \leftarrow_\$ \mathcal{K}$
$b' \leftarrow \mathcal{B}^{\mathsf{Decaps}^{\not\perp}_m(\cdot), H(\cdot)}(\mathsf{pk}, c^*, K^*)$
if $\exists (m', K') \in \mathfrak{L}_H$ s.th. $\mathsf{Enc}(\mathsf{pk}, m') = c^*$
    return $m'$
else
    abort
```

Figure 3.3: Adversary $\mathcal{A}$ against OW-CPA for the proof of Theorem 3.1, where H is defined as in Game G_2 and $\mathsf{Decaps}^{\not\perp}_m$ is defined as in Game G_3 of Figure 3.2

$m' \neq m^*$ such that $\exists K''(m', K'') \in \mathfrak{L}_H \wedge \mathsf{Enc}(\mathsf{pk}, m') = c^*$ because then either m' or m^* would have to exhibit a correctness error and CORERR would occur, contradicting our assumption. Thus $\mathcal{A}$ returns the unique $m' = m^*$ and wins the OW-CPA game (because CORERR did not occur and therefore $\mathsf{Dec}(\mathsf{sk}, c^*) = m^*$ matching $\mathcal{A}$'s output). Hence,

$$\Pr[\mathsf{CHAL} \wedge \overline{\mathsf{CORERR}}] \leq \mathsf{Adv}^{\text{OW-CPA}}_{\text{DPKE}, \mathcal{D}}(A)$$

Collecting the probabilities yields the required bound. □

3.1.2 Security of NTRU

Let $\mathcal{M}$ and $\mathcal{C}$ be the message and ciphertext spaces for NTRU-DPKE respectively and let $\mathcal{K}$ be the key space used for the NTRU KEM. Let $H : \mathcal{M} \cup (\{0,1\}^\ell \times \mathcal{C}) \to \mathcal{K}$ be a random oracle.

We can see from Figure 3.4 that the NTRU KEM can almost be interpreted as $\mathsf{U}^{\not\perp}_m[\text{NTRU-DPKE}, H]$ of [HHK17] with the only difference being that rm and s are not sampled uniformly at random from the same space $\mathcal{M}$. This difference is precisely what the $\mathsf{DU}^{\not\perp}_m$ transformation seen in Figure 3.1 does

```
$\mathsf{KGen}(1^\mathcal{K})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^\mathcal{K})$
$s \leftarrow_\$ \{0,1\}^{256}$
$\overline{\mathsf{sk}} := (\mathsf{sk}, s)$
return $(\mathsf{pk}, \overline{\mathsf{sk}})$

$\mathsf{Encaps}(\mathsf{pk} = h)$
$coins \leftarrow_\$ \{0,1\}^{256}$
$rm \leftarrow \mathsf{Sample\_rm}(coins)$
$c := \mathsf{Enc}(h, rm)$
$K := H(rm)$
return $(K, c)$

$\mathsf{Decaps}(\overline{\mathsf{sk}} = (\mathsf{sk}, s), c)$
$rm := \mathsf{Dec}(\mathsf{sk}, c)$
if $rm \neq \perp$ then
    return $K := H(rm)$
else
    return $K := H(s, c)$
```

Figure 3.4: The NTRU KEM scheme, as described in [Xag22, Figure 7] and [CDH+20] with $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}) = $ NTRU-DPKE.

as well. The paper [SXY18] presents the SXY transformation for a similar effect but the security proof of the SXY transformation in the ROM relies on [HHK17, Theorem 3.6] without arguing about the non-uniform message distribution at all. So instead of basing the security of NTRU on the SXY transformation, we use our own Theorem 3.1 above.

Corollary 3.2 (Security of NTRU in the ROM) *NTRU is perfectly correct. Furthermore, for any IND-CCA adversary $\mathcal{B}$ against NTRU issuing at most q_H queries to the random oracle H, there exists an OW-CPA adversary $\mathcal{A}$ against NTRU-DPKE, such that*

$$\mathsf{Adv}_{\mathsf{NTRU}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq \mathsf{Adv}_{\mathsf{NTRU\text{-}DPKE,\ Sample\_rm}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) + \frac{q_H}{2^{256}}$$

Proof This follows directly from Theorem 3.1 because NTRU-DPKE is perfectly correct, deterministic, rigid and does not use any random oracle. The message distribution used in NTRU is Sample_rm. And further, in NTRU-DPKE messages have a length of $8 \cdot$ dpke_plaintext_bytes bits and ciphertexts have a bit length of $8 \cdot$ dpke_ciphertext_bytes (see [CDH$^+$20]). In any reasonable parameter set, we will have dpke_plaintext_bytes $<$ dpke_ciphertext_bytes and $\ell = 256$ is fixed, so we can conclude that $(\{0,1\}^\ell \times \mathcal{C}) \cap \mathcal{M} = \emptyset$. □

As we just showed, we can reduce the IND-CCA security of NTRU to a certain non-standard OW-CPA security (using the Sample_rm distribution) of NTRU-DPKE. Typically, we would want to reduce instead to the standard notion of OW-CPA security, namely under a uniform message distribution.

It is well-known that we could achieve this last step by now arguing that the only difference is this change of distribution and therefore we can bound the OW-CPA security under the Sample_rm distribution by the standard OW-CPA security plus the statistical distance SD(Sample_rm,$\mathcal{U}$).

The following section pursues this approach.

Statistical Distance in NTRU-HRSS We require a bound on the statistical distance between a uniform distribution on $S/3$[1] and the distribution that Sample_rm produces. As the following will simply serve as an illustration of the problem, we focus on the NTRU-HRSS parameter set. In the following, let $\mathcal{D}$ be the distribution produces by Sample_rm.

Sample_rm gets as input an appropriate number of independent, uniformly random bits. These bits are grouped into bytes and interpreted as the coefficients of the polynomial $v \in \mathbb{Z}[x]$. By construction, v has degree $n-2$. Before v is returned from Sample_rm however, it is reduced modulo Φ_n and the coefficients are reduced modulo 3.

[1] For the definition of the polynomial space $S/3$, we refer the reader to [CDH$^+$20].

The former does nothing, as v's degree is already small enough. During the latter process however, we move from every coefficient being independent and identically distributed (i.i.d.) following a uniform distribution on $\{0,1,\ldots,255\}$ to every coefficient being i.i.d. using a non-uniform distribution we call $\mathcal{F}$ on $\{-1,0,1\}$. In particular:

$$\mathcal{F}(0) = \frac{258}{768} = \frac{1}{3} + \frac{2}{768}$$
$$\mathcal{F}(1) = \mathcal{F}(-1) = \frac{255}{768} = \frac{1}{3} - \frac{1}{768}$$

This affects the distribution $\mathcal{D}$, as the probability of any polynomial depends now on the number of coefficients that are 0. In particular, given any polynomial $x \in S/3$ with z coefficients equal to 0, we have:

$$\Pr_{\mathcal{D}}[x] = \mathcal{F}(0)^z \cdot \mathcal{F}(1)^{n-1-z}$$

By using the formula $\mathsf{SD}(\mathcal{D},\mathcal{U}) = \frac{1}{2}\sum_{x\in S/3} |\Pr_{\mathcal{D}}[x] - \Pr_{\mathcal{U}}[x]|$ and by grouping polynomials together by the number of coefficients equal to 0, we get:

$$\mathsf{SD}(\mathcal{D},\mathcal{U}) = \frac{1}{2}\sum_{z=0}^{n-1} \binom{n-1}{z} \cdot 2^{n-1-z} \cdot \left|\mathcal{F}(0)^z \cdot \mathcal{F}(1)^{n-1-z} - 3^{1-n}\right|$$

This can be computed for the case of $n = 701$ that NTRU recommends for NTRU-HRSS to give a statistical distance of 0.05821. It should also be noted that this statistical distance could be significant in terms of the concrete security of NTRU. In fact, the statistical distance even increases monotonically with n, though we provide no proof for that fact here.

Security of NTRU From the discussion above, it becomes clear that we cannot really reduce the security of NTRU to the OW-CPA security of NTRU-DPKE using a uniform distribution. Instead we simply stick with Corollary 3.2 above and assume the OW-CPA security of NTRU-DPKE under the message distribution Sample_rm. We have shown that NTRU is an IND-CCA secure KEM under this assumption.

From here, there are two options to fully prove the security of NTRU. There may well be another way to tightly reduce the security of NTRU-DPKE to OW-CPA security under a uniform distribution. For example, other research suggests using the Hellinger Distance instead [Yas21].

It might also be possible to instead further reduce the OW-CPA security of NTRU-DPKE using the non-uniform distribution to some reasonable hardness assumption. Note that this has to examined for a given parameter set. Both of these tasks are left open for future work.

```
$\mathsf{KGen}(1^{\mathcal{K}})$
    return $\mathsf{KGen}_B(1^{\mathcal{K}})$

$\mathsf{Enc}(\mathsf{pk}, e)$
    return $\mathsf{Enc}_B(\mathsf{pk}, e)$

$\mathsf{Dec}(\mathsf{sk}, c)$
    $e := \mathsf{Dec}_B(\mathsf{sk}, c)$
    if $\mathsf{Enc}_B(\mathsf{pk}, e) = c$ then
        return $e$
    return $\perp$
```

Figure 3.5: A DPKE scheme CM-DPKE = (KGen, Enc, Dec) incorporating a re-encryption check as part of decryption where $(\mathsf{KGen}_B, \mathsf{Enc}_B, \mathsf{Dec}_B)$ = CM-BASE.

```
$\mathsf{KGen}_I(1^{\mathcal{K}})$
    return $\mathsf{KGen}(1^{\mathcal{K}})$

$\mathsf{Enc}_I(\mathsf{pk}, e)$
    $c_0 := \mathsf{Enc}(\mathsf{pk}, e)$
    $c_1 := F(e)$
    return $(c_0, c_1)$

$\mathsf{Dec}_I(\mathsf{sk}, (c_0, c_1))$
    $e := \mathsf{Dec}(\mathsf{sk}, c_0)$
    if $e \neq \perp \wedge c_1 = F(e)$ then
        return $e$
    return $\perp$
```

Figure 3.6: An intermediate scheme CM-INTER = $(\mathsf{KGen}_I, \mathsf{Enc}_I, \mathsf{Dec}_I)$ incorporating hashing of the message as part of encryption where (KGen, Enc, Dec) = CM-DPKE.

3.2 Classic McEliece

In order to understand Classic McEliece in a more modular sense, we define multiple schemes gradually building up to the Classic McEliece KEM in an analogous way to the $\mathsf{FO}^{\not\perp}$ transformation from [HHK17].

- CM-BASE is the Niederreiter PKE using Goppa Codes which is at the core of Classic McEliece.
- CM-DPKE can be described as a T transformation of CM-BASE, and it is the scheme the Classic McEliece specification uses for the Encode and Decode procedures. It is defined in Figure 3.5 and in [ABC+20].
- CM-INTER is essentially CM-DPKE but it also incorporates a hash of the message as part of the ciphertext and checks it upon decryption. It is defined in Figure 3.6.
- CM-KEM is then finally the Classic McEliece KEM and almost a $\mathsf{DU}^{\not\perp}$ transformation of CM-INTER (a more detailed description of the transformation follows further below). It is defined in Figure 3.7.

Let $\mathcal{M}$ and $\mathcal{C}$ be the message and ciphertext spaces for CM-DPKE respectively and let $\mathcal{K}$ be the key space used for the Classic McEliece KEM. Let $F : \mathcal{M} \to \mathcal{R}$ be a random oracle and let $H_0 : \{0,1\}^{\ell} \times \mathcal{C} \times \mathcal{R} \to \mathcal{K}$ and $H_1 : \mathcal{M} \times \mathcal{C} \times \mathcal{R} \to \mathcal{K}$ be two more random oracles, all three of them independent.

```
$\mathsf{KGen}_{\mathsf{KEM}}(1^{\mathcal{K}})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}_I(1^{\mathcal{K}})$
$s \leftarrow_\$ \mathbb{F}_2^n$
$\overline{\mathsf{sk}} := (\mathsf{sk}, s)$
return $(\mathsf{pk}, \overline{\mathsf{sk}})$
```

```
$\mathsf{Encaps}(\mathsf{pk})$
$e \leftarrow \textsc{FixedWeight}()$
$c := \mathsf{Enc}_I(\mathsf{pk}, e)$
$K := H_1(e, c)$
return $(K, c)$
```

```
$\mathsf{Decaps}(\overline{\mathsf{sk}} = (\mathsf{sk}, s), c)$
$e := \mathsf{Dec}_I(\mathsf{sk}, c)$
if $e \neq \bot$ then
    $K := H_1(e, c_0, c_1)$
else
    $K := H_0(s, c_0, c_1)$
return $K$
```

Figure 3.7: The Classic McEliece KEM scheme, as described in [Xag22, Fig. 18] and [ABC$^+$20] with $(\mathsf{KGen}_I, \mathsf{Enc}_I, \mathsf{Dec}_I) = $ CM-INTER. We renamed the oracles from [ABC$^+$20] to H_0, H_1 and F (implicit in CM-INTER) depending on their first argument. FixedWeight is used to generate messages for the underlying PKE scheme.

Security Goals Notice first that the message distribution used by the KEM construction (in the specification this distributions is generated by the function FixedWeight) indeed outputs uniformly random elements of the message space of CM-BASE. This is why we can analyze this finalist without worrying about non-uniform message distributions.

We will prove the following security guarantees, assuming that CM-BASE is OW-CPA secure.

- CM-DPKE is OW-PCA secure.
- CM-INTER is OW-PCA secure.
- CM-KEM is IND-CCA secure.

CM-DPKE CM-DPKE, as depicted in Figure 3.5, can be understood as simply the output of a T transformation applied to CM-BASE. In particular, this scheme adds a "re-encryption check" to decryption. We conclude that CM-DPKE is OW-PCA secure:

Corollary 3.3 (Security of CM-DPKE in the ROM) *CM-DPKE is perfectly correct. Furthermore, for any OW-PCA adversary $\mathcal{B}$ against CM-DPKE issuing at most q_P queries to a plaintext checking oracle* Pco*, there exists an OW-CPA adversary $\mathcal{A}$ against CM-BASE, such that*

$$\mathit{Adv}\,^{\text{OW-PCA}}_{\text{CM-DPKE}}(\mathcal{B}) \leq \mathit{Adv}\,^{\text{OW-CPA}}_{\text{CM-BASE}}(\mathcal{A})$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, CM-DPKE is rigid.

Proof We carry out a direct reduction from CM-DPKE's OW-PCA security to the OW-CPA of CM-BASE. In essence, this proof is a shortened version of the proof for [HHK17, Theorem 3.1] where the PKE scheme is deterministic and perfectly correct, like CM-BASE. Under these two assumptions we

```
$\mathcal{A}(\mathsf{pk}, c^*)$
    $m' \leftarrow \mathcal{B}^{\textsc{Pco}(\cdot)}(\mathsf{pk}, c^*)$
    return $m'$

$\textsc{Pco}(m \in \mathcal{M}, c)$
    return $[\![\mathsf{Enc}_B(\mathsf{pk}, m) = c]\!]$
```

Figure 3.8: Adversary $\mathcal{A}$ against OW-CPA security of CM-BASE.

can carry out a direct reduction rather simply and without introducing any game hops.

Let $\mathcal{B}$ be some OW-PCA adversary against CM-DPKE issuing at most q_P queries to a plaintext checking oracle Pco. We construct an OW-CPA adversary $\mathcal{A}$ against CM-BASE explicitly as in Figure 3.8.

We argue first that $\mathcal{A}$'s implementation of the Pco oracle is a perfect simulation. Because $\mathsf{KGen} = \mathsf{KGen}_B$ and $\mathsf{Enc} = \mathsf{Enc}_B$, simply forwarding pk, c^* is already a perfect simulation of the setup procedure. Then, we argue about the advantage $\mathcal{A}$ achieves and finish the proof.

Let Pco be a plaintext checking oracle for CM-DPKE. Pco is characterized by the fact that, upon input of $m \in \mathcal{M}, c$, it responds with bit 1 if and only if $\mathsf{Dec}_B(\mathsf{sk}, c) \neq \perp$, $\mathsf{Enc}_B(\mathsf{pk}, \mathsf{Dec}_B(\mathsf{sk}, c)) = c$ and $\mathsf{Dec}_B(\mathsf{sk}, c) = m$.

Our implementation checks whether $\mathsf{Enc}_B(\mathsf{pk}, m) = c$. If this is the case, then by the perfect correctness of CM-BASE, $\mathsf{Dec}_B(\mathsf{sk}, c) = m \neq \perp$. This satisfies all three of the requirements above.

On the other hand, if $\mathsf{Enc}_B(\mathsf{pk}, m) \neq c$ then all of the three statements above cannot be simultaneously true as $\mathsf{Enc}_B(\mathsf{pk}, \mathsf{Dec}_B(\mathsf{sk}, c)) = c$ and $\mathsf{Dec}_B(\mathsf{sk}, c) = m$ together imply $\mathsf{Enc}_B(\mathsf{pk}, m) = c$, contradicting our assumption.

This shows that $[\![\mathsf{Enc}_B(\mathsf{pk}, m) = c]\!]$ is indeed 1 if and only if $\textsc{Pco}(m, c)$ returns 1 as well and thus that $\mathcal{A}$ perfectly simulates the OW-PCA game towards $\mathcal{B}$.

Now, note that whenever $\mathcal{B}$ wins the OW-PCA game against CM-DPKE, we have $\mathsf{Dec}(\mathsf{sk}, c^*) = m'$. By definition this implies $\mathsf{Dec}_B(\mathsf{sk}, c^*) = m'$. Since $\mathcal{A}$ returns exactly the same m', it also wins its OW-CPA game against CM-BASE. This lets us conclude

$$\mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}DPKE}}(\mathcal{B}) \leq \mathsf{Adv}\,^{\mathsf{OW\text{-}CPA}}_{\mathsf{CM\text{-}BASE}}(\mathcal{A})$$

finishing our proof.

Note that in [HHK17], there was no argument about why the T transformation adds rigidity to a non-rigid scheme, so we quickly include this for completeness. Assume that there exist some m, c such that $\mathsf{Dec}(\mathsf{sk}, c) = m \wedge \mathsf{Enc}(\mathsf{pk}, m) \neq c$, i.e. a violation of CM-DPKE's rigidity. Because Dec incorporated a re-encryption check, it can only output m after decryption if

$\mathsf{Enc}_B(\mathsf{pk}, m) = c$ (notice this is the underlying encryption function). However, this contradicts our assumption that $\mathsf{Enc}(\mathsf{pk}, m) \neq c$ since for any pk, m it is true that $\mathsf{Enc}(\mathsf{pk}, m) = \mathsf{Enc}_B(\mathsf{pk}, m)$. Thus, the T transformation has managed to provide rigidity by means of the re-encryption check. □

CM-INTER The change between CM-DPKE and CM-INTER (namely the addition of a message hash to the ciphertext) is not one usually carried out in any of the transformations in [HHK17] but it does not decrease the security of the scheme. We will prove this fact by reduction where the main technique is to patch the random oracle F to be consistent with a hash that we generate before knowing m^*.

Theorem 3.4 (Security of CM-INTER in the ROM) *CM-INTER is perfectly correct. Furthermore, for any OW-PCA adversary $\mathcal{B}$ against CM-INTER issuing at most q_P queries to a plaintext checking oracle* Pco_I*, there exists an OW-PCA adversary $\mathcal{A}$ against CM-DPKE that makes at most q_P queries to its* Pco *oracle, such that*

$$Adv\,^{OW\text{-}PCA}_{CM\text{-}INTER}(\mathcal{B}) \leq Adv\,^{OW\text{-}PCA}_{CM\text{-}DPKE}(\mathcal{A})$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, CM-INTER is rigid.

Proof Correctness and rigidity are easy to verify given correctness and rigidity of CM-DPKE.

Let $\mathcal{B}$ be an adversary against CM-INTER issuing at most q_P queries to a plaintext checking oracle Pco_I. We construct an OW-PCA adversary $\mathcal{A}$ against CM-DPKE as shown in Figure 3.9.

Using the plaintext checking oracle Pco from the OW-PCA game against CM-DPKE, $\mathcal{A}$ can perfectly simulate the oracle Pco_I for $\mathcal{B}$ as shown. The patching of $F(m^*)$ cannot be noticed by $\mathcal{B}$ as the random oracle's outputs remain uniformly random. There can also not be any $m' \neq m^*$ that F could "mistake" for m^* because if we had $\mathsf{Enc}(\mathsf{pk}, m') = \mathsf{Enc}(\mathsf{pk}, m^*)$, at least one of them would exhibit a correctness error in CM-DPKE. But that is not possible as CM-DPKE is perfectly correct.

These two observations allow us to conclude that the OW-PCA game for $\mathcal{B}$ is in fact perfectly simulated by $\mathcal{A}$.

We now finish the proof by simply noticing that if $\text{Pco}_I(m, c_0, c_1) = 1$ for some m, c_0, c_1 then it must also hold that $\text{Pco}(m, c_0) = 1$ for the same m and c_0. Thus, every time $\mathcal{B}$ wins the simulated game with $\mathcal{A}$, $\mathcal{A}$ wins the OW-PCA game against CM-DPKE:

$$\begin{aligned} \mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}INTER}}(\mathcal{B}) &= \Pr[\mathsf{OW\text{-}PCA}^{B}_{\mathsf{CM\text{-}INTER}} \Rightarrow 1] \\ &\leq \Pr[\mathsf{OW\text{-}PCA}^{A}_{\mathsf{CM\text{-}DPKE}} \Rightarrow 1] = \mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}DPKE}}(\mathcal{A}) \end{aligned}$$ □

```
$\mathcal{A}(\mathsf{pk}, c^*)$
$r \leftarrow_\$ \mathcal{R}$
$c := (c^*, r)$
$m' \leftarrow \mathcal{B}^{\textsc{Pco}_I(\cdot), F(\cdot)}(\mathsf{pk}, c)$
return $m'$

$\textsc{Pco}_I(m, c_0, c_1)$
return $\textsc{Pco}(m, c_0) \wedge F(e) = c_1$

$F(m)$
if $\mathsf{Enc}(\mathsf{pk}, e) = c^*$
    return $r$
else
    return $F'(e)$
```

Figure 3.9: Adversary $\mathcal{A}$ against OW-PCA where $\textsc{Pco}$ is defined as in the OW-PCA game against CM-DPKE and F' is an internal random oracle that $\mathcal{B}$ cannot access.

```
$\mathsf{KGen}(1^{\mathcal{K}})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^{\mathcal{K}})$
$s \leftarrow_\$ \{0,1\}^\ell$
$\overline{\mathsf{sk}} := (\mathsf{sk}, s)$
return $(\mathsf{pk}, \overline{\mathsf{sk}})$

$\mathsf{Encaps}(\mathsf{pk})$
$m \leftarrow \mathcal{D}(\mathcal{M})$
$c := \mathsf{Enc}(\mathsf{pk}, m)$
$K := H_1(m, c)$
return $(K, c)$

$\mathsf{Decaps}(\overline{\mathsf{sk}} = (\mathsf{sk}, s), c)$
$m' := \mathsf{Dec}(\mathsf{sk}, c)$
if $m' \neq \perp$ then
    return $K := H_1(m', c)$
else
    return $K := H_0(s, c)$
```

Figure 3.10: The transformation $\overline{\mathsf{DU}}^{\not\perp}[\mathsf{DPKE}, H_0, H_1, \mathcal{D}]$ where $\mathsf{DPKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

CM-KEM And finally, we can analyze CM-KEM as the result of applying a slightly modified $\mathsf{DU}^{\not\perp}$ transformation to CM-INTER where instead of using the same random oracle for both $H(m, c)$ and $H(s, c)$ queries, we use two independent random oracles. We call this transformation $\overline{\mathsf{DU}}^{\not\perp}$ and we define it explicitly in Figure 3.10 and we present a theorem about the security it provides as Theorem 3.5.

Theorem 3.5 (DPKE OW-PCA $\overset{\text{ROM}}{\Longrightarrow}$ $\overline{\mathsf{DU}}^{\not\perp}$[DPKE, $H_0, H_1, \mathcal{D}$] IND-CCA) *Let KEM := $\overline{DU}^{\not\perp}$[DPKE, $H_0, H_1, \mathcal{D}$] in the following. Let H_0, H_1 be two independent random oracles. If DPKE is δ-correct, then so is KEM. For any IND-CCA adversary $\mathcal{B}$ against KEM issuing at most q_{H_0} queries to the random oracle H_0, there exists an OW-PCA adversary $\mathcal{A}$ against DPKE, such that*

$$\mathit{Adv}\,_{\mathit{KEM}}^{\mathit{IND\text{-}CCA}}(\mathcal{B}) \leq \mathit{Adv}\,_{\mathit{DPKE},\ \mathcal{D}}^{\mathit{OW\text{-}PCA}}(\mathcal{A}) + \frac{q_{H_0}}{2^\ell}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.

Proof (Proof sketch) This proof is essentially obtained from the proof of [HHK17, Theorem 3.4]. The only differences in the proof are as follows:

We generate m^* using the distribution $\mathcal{D}$ as in the other theorems. Consequently, $\mathcal{D}$ will show up again in the advantage against DPKE but otherwise the proof is unaffected. We also generate s uniformly at random from the space $\{0,1\}^\ell$.

In Game G_1, we would further change the use of H_0 to using an internal random oracle (H' in the proofs of Theorem 3.1 and [HHK17, Theorem 3.4]) to make the output from decapsulation perfectly random upon decryption failure. So we adapt the proof and argue instead that:

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_0^{\mathcal{B}} \Rightarrow 1]\right| \leq \frac{q_{H_0}}{2^\ell}$$

We again use that 2^ℓ is the size of the space (the uniform secret) s is sampled from. We can omit the check whether $m' = s$ as there is no harm in making such queries on H_1 available to the adversary.

From this point on, H_0 is used only to check whether the adversary queries it on inputs of the form $H_0(s, \cdot)$ (called QUERY in [HHK17]) but it is not used in the game anymore and its output is either irrelevant or not available for $\mathcal{B}$ from this point onward. The rest of the proof proceeds as in [HHK17] where their H is instantiated with our H_1. Also note that we do not need to check for the QUERY event in H_1 as queries involving s produce no output of any particular relevance.

The final change is the addition of the $\mathcal{D}$ distribution into the bound in Game G_3. In particular, we conclude now that

$$\Pr[\mathsf{CHAL}] = \mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{DPKE},\,\mathcal{D}}(\mathcal{A})$$

With these modifications, the proof holds for our modified games.

We provide the game hops incorporating our changes in Figure 3.11.

Corollary 3.6 (Security of CM-KEM in the ROM) *CM-KEM is perfectly correct. For any IND-CCA adversary $\mathcal{B}$ against CM-KEM issuing at most q_{H_0} queries to the random oracle H_0 and q_{H_1} queries to the random oracle H_1, there exists an OW-PCA adversary $\mathcal{A}$ against CM-INTER, such that*

$$\mathit{Adv}\,^{\mathit{IND\text{-}CCA}}_{\mathit{CM\text{-}KEM}}(\mathcal{B}) \leq \mathit{Adv}\,^{\mathit{OW\text{-}PCA}}_{\mathit{CM\text{-}INTER}}(\mathcal{A}) + \frac{q_{H_0}}{2^\ell}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.

Proof This follows directly from Theorem 3.5 because CM-INTER is perfectly correct and deterministic. Also, H_0 and H_1 are in fact independent random oracles because of Domain Separation. In particular, the first byte of the input to the underlying random oracle is fixed and different for the two oracles. Finally, the message distribution in use (FixedWeight) is in fact the uniform distribution, hence we can omit $\mathcal{D}$ in the advantage term. □

```
**GAMES** $G_0$-$G_3$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
$s \leftarrow_\$ \{0,1\}^\ell$
$\mathsf{sk}' := (\mathsf{sk}, s)$
$m^* \leftarrow \mathcal{D}(\mathcal{M})$
$c^* := \mathsf{Enc}(\mathsf{pk}, m^*)$
$K_0^* := H_1(m^*, c^*)$
$K_1^* \leftarrow_\$ \{0,1\}^n$
$b \leftarrow_\$ \{0,1\}$
$b' \leftarrow \mathcal{B}^{\mathsf{Decaps}^{\not\perp}, H_0, H_1}(\mathsf{pk}, c^*, K_b^*)$
return $[\![b' = b]\!]$

$\mathsf{Decaps}_m^{\not\perp}(c \neq c^*)$    // $G_0$-$G_1$
$m' := \mathsf{Dec}(\mathsf{sk}, c)$
if $m' = \perp$ then
    return $K := H_0(s, c)$            // $G_0$
    return $K := H'(c)$                // $G_1$
else return $K := H_1(m', c)$

$\mathsf{Decaps}_m^{\not\perp}(c \neq c^*)$    // $G_2$-$G_3$
if $\exists K$ such that $(c, K) \in \mathfrak{L}_D$
    return $K$
$K \leftarrow_\$ \mathcal{K}$
$\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$
return $K$
```

```
$H_1(m, c)$
if $\exists K$ s.th. $(m, K) \in \mathfrak{L}_H$
    return $K$
$K \leftarrow_\$ \mathcal{K}$
if $\mathsf{Dec}(\mathsf{sk}', c) = m$                     // $G_2$-$G_3$
    if $m = m^*$                                // $G_3$
        CHAL := true; abort                     // $G_3$
    if $\exists K'$ s.th. $(c, K') \in \mathfrak{L}_D$    // $G_2$-$G_3$
        $K := K'$                               // $G_2$-$G_3$
    else                                        // $G_2$-$G_3$
        $\mathfrak{L}_D := \mathfrak{L}_D \cup \{(c, K)\}$        // $G_2$-$G_3$
$\mathfrak{L}_H := \mathfrak{L}_H \cup \{(m, c, K)\}$
return $K$

$H_0(s', c)$
if $\exists K$ s.th. $(s', c, K) \in \mathfrak{L}_{Hs}$
    return $K$
if $s' = s$ then                                // $G_1$-$G_3$
    QUERY := true; abort                        // $G_1$-$G_3$
$K \leftarrow_\$ \mathcal{K}$
$\mathfrak{L}_{Hs} := \mathfrak{L}_{Hs} \cup \{(s', c, K)\}$
return $K$                                      □
```

Figure 3.11: Games G_0 - G_3 for the proof of Theorem 3.5

As this was the last reduction needed, we can now collect all the bounds in a final corollary.

Corollary 3.7 (Security of Classic McEliece in the ROM) *CM-KEM is perfectly correct. For any IND-CCA adversary $\mathcal{B}$ against CM-KEM issuing at most q_{H_0} queries to the random oracle H_0 and q_{H_1} queries to the random oracle H_1, there exists an OW-CPA adversary $\mathcal{A}$ against CM-BASE, such that*

$$\mathit{Adv}\,^{\textit{IND-CCA}}_{\textit{CM-KEM}}(\mathcal{B}) \leq \mathit{Adv}\,^{\textit{OW-CPA}}_{\textit{CM-BASE}}(\mathcal{A}) + \frac{q_{H_0}}{2^{256}}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.

```
$\mathsf{KGen}(1^{\mathcal{K}})$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^{\mathcal{K}})$
    $z \leftarrow_\$ \{0,1\}^\ell$
    $\overline{\mathsf{sk}} := (\mathsf{sk}, z, \mathsf{pk})$
    return $(\mathsf{pk}, \overline{\mathsf{sk}})$
```

```
$\mathsf{Encaps}(\mathsf{pk})$
    $m \leftarrow_\$ \mathcal{M}$
    $m := H(m)$
    $r := G_2(m, H(\mathsf{pk}))$
    $c := \mathsf{Enc}(\mathsf{pk}, m; r)$
    $K' := G_1(m, H(\mathsf{pk}))$
    $K := KDF(K', H(c))$
    return $(K, c)$
```

```
$\mathsf{Decaps}(\overline{\mathsf{sk}} = (\mathsf{sk}, z, \mathsf{pk}), c)$
    $m' := \mathsf{Dec}(\mathsf{sk}, c)$
    $r := G_2(m', H(\mathsf{pk}))$
    if $c \neq \mathsf{Enc}(\mathsf{pk}, m'; r)$ then
        return $K := KDF(z, H(c))$
    else
        $K' := G_1(m', H(\mathsf{pk}))$
        return $K := KDF(K', H(c))$
```

Figure 3.12: The abstract KS transformation where H, G_1, G_2 and KDF are random oracles, ℓ is a parameter of the transformation defaulting to $\ell = 256$ and the underlying PKE scheme is $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

Proof This follows from Theorems and Corollaries 3.3, 3.4 and 3.6 as well as the fact that $\ell = 256$ is fixed in [ABC$^+$20].

We construct an OW-PCA adversary $\mathcal{A}'$ against CM-INTER analogous to [HHK17, Figure 14], with the same minor changes as in the game hops in the proof of Theorem 3.5 (we omit the construction here). Notice that this adversary now makes $q_P = q_{H_1}$ queries to the Pco oracle.

We then construct another OW-PCA adversary $\mathcal{A}''$ against CM-DPKE as in Figure 3.9. And finally, we employ Theorem 3.3 to get a third adversary $\mathcal{A}$ against CM-BASE.

From our constructions as well as Theorem 3.3, Theorem 3.4 and Theorem 3.6 respectively, we conclude:

$$\begin{aligned}
\mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}DPKE}}(\mathcal{A}'') &\leq \mathsf{Adv}\,^{\mathsf{OW\text{-}CPA}}_{\mathsf{CM\text{-}BASE}}(\mathcal{A}) \\
\mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}INTER}}(\mathcal{A}') &\leq \mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}DPKE}}(\mathcal{A}'') \\
\mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{CM\text{-}KEM}}(\mathcal{B}) &\leq \mathsf{Adv}\,^{\mathsf{OW\text{-}PCA}}_{\mathsf{CM\text{-}INTER}}(\mathcal{A}') + \frac{q_{H_0}}{2^\ell}
\end{aligned}$$

Combining these inequalities yields our claimed bound. □

This concludes our discussion of Classic McEliece as we have now successfully reduced the IND-CCA security of CM-KEM to the OW-CPA security of CM-BASE, i.e. the Niederreiter PKE using Goppa Codes.

3.3 CRYSTALS-KYBER and SABER

We will discuss the last two remaining finalists in one section because we can easily abstract them to the same type of transformation which we will call KS. Loosely, this transformation is somewhat similar to $\mathsf{FO}^{\not\perp}$ in [HHK17] but there a couple of details that complicate this approach. We define it in Figure 3.12.

3.3.1 Assumptions

Let $\mathcal{M}$ and $\mathcal{C}$ be the message and ciphertext space for PKE respectively. Let $\mathcal{PK}$ be the public key space of PKE and let $\mathcal{K}$ be the key space used for the final KEM. Let $\mathcal{R}$ be a space used for fixing the randomness in PKE's encryption. Let $\mathcal{S}$ be the output space of the random oracle H defined below. Let $\mathcal{K}' = \{0,1\}^{\ell}$.

Let $H : (\mathcal{M} \cup \mathcal{C} \cup \mathcal{PK}) \to \mathcal{S}$, $G_1 : \mathcal{M} \times \mathcal{S} \to \mathcal{K}'$ and $G_2 : \mathcal{M} \times \mathcal{S} \to \mathcal{R}$ be independent random oracles. $KDF : \mathcal{K}' \times \mathcal{S} \to \mathcal{K}$ is either another independent random oracle or it may resolve to the same underlying function as H. Any instantiation needs to define these random oracles as well as the underlying PKE scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and the parameter ℓ.

Let PKE be δ-correct.

To ensure that the random oracles never interfere with each other in the analysis and modifications to parts of the oracles do not affect other parts, we require the following:

- If H and KDF are instantiated using the same underlying hash function (i.e. they denote the same random oracle in our model) then we require all of the following to be true
  - $(\mathcal{K}' \times \mathcal{S}) \cap \mathcal{PK} = \emptyset$
  - $(\mathcal{K}' \times \mathcal{S}) \cap \mathcal{C} = \emptyset$
  - $(\mathcal{K}' \times \mathcal{S}) \cap \mathcal{M} = \emptyset$
  - $\mathcal{PK} \cap \mathcal{M} = \emptyset$
  - $\mathcal{C} \cap \mathcal{M} = \emptyset$
- If, on the other hand, H and KDF are independent random oracles then we only require that the following holds
  - $\mathcal{PK} \cap \mathcal{M} = \emptyset$
  - $\mathcal{C} \cap \mathcal{M} = \emptyset$

3.3.2 Instantiations

To get CRYSTALS-KYBER, we instantiate:

- H with SHA3-256

- G_1, G_2 with SHA3-512.
  In CRYSTALS-KYBER, only one random oracle G with longer output was used for generating both intermediate keys (the K' in Figure 3.12) as well as for fixing PKE's random coins. In other words, lines 7 and 9 in Figure 3.12 would be replaced by a single line $(K', r) := G(m, H(\mathsf{pk}))$. As the outputs of G were split exactly in half and used in distinct contexts in [SAB$^+$20], we decided to instead regard these outputs as the outputs of two independent random oracles (the output of G_1 comes from the first half of G's output and the output of G_2 comes from the second half). This modelling is equivalent in the ROM because we assume G_1 and G_2 to be independent.
- *KDF* with SHAKE-256
- PKE with Kyber.CPAPKE from [SAB$^+$20]
- ℓ with $\ell = 256$
- $\mathcal{M}$ with $\{0,1\}^{256}$

In all of the parameter sets for CRYSTALS-KYBER (Kyber512, Kyber768, Kyber1024 as well as their 90s variants), the spaces of message, ciphertext and public key are such that the above requirements are met.

To obtain SABER, we instantiate:

- H and *KDF* with SHA3-256
- G_1, G_2 with SHA3-512 (as above, G_1's output comes from the first half and G_2's from the second half)
- PKE with Saber.PKE from [DKR$^+$20]
- ℓ with $\ell = 256$
- $\mathcal{M}$ with $\{0,1\}^{256}$

In all of the parameter sets for SABER (LightSaber-KEM, Saber-KEM and FireSaber-KEM), the spaces of message, ciphertext and public key are such that the above requirements are met.

Note that while the hashing of the message in line 6 is not present in [DKR$^+$20, Section 2.5], it is present in the technical specification in [DKR$^+$20, Section 8.5.2]. We will regard it as a part of SABER in this work but otherwise SABER simply corresponds to the scheme KS' below and has a slightly better bound on the IND-CCA adversary's advantage.

3.3.3 Slightly improved version of [HHK17, Theorem 3.2]

The original theorem 3.2 in [HHK17] had two issues that we want to point out here. Firstly, [HHK17, Theorem 3.2] contained an error as the term

$2q_G + 1$ should have instead been $2(q_G + q_P) + 1$. This stems from the fact that queries to the PCO oracle also caused a query to G hence contributing to the probability of the event QUERY. This mistake was corrected in [Höv21b, Theorem 2.1.3] and pointed out in an updated version of the original paper.

Secondly, even the bound given in [Höv21b, Theorem 2.1.3] can be made slightly tighter. We now state an improved version of this theorem.

Theorem 3.8 (PKE IND-CPA $\overset{\text{ROM}}{\Longrightarrow}$ T[PKE, G] OW-PCVA) *Let PKE' := T[PKE, G] in the following. Assume PKE to be δ-correct and γ-spread. Then, for any OW-PCVA adversary $\mathcal{B}$ that issues at most q_G queries to the random oracle G, q_P queries to a plaintext checking oracle* PCO, *and q_V queries to a validity checking oracle* CVO, *there exists an IND-CPA adversary $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{OW\text{-}PCVA}}_{\mathsf{PKE'}}(\mathcal{B}) \leq (q_G + q_P) \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{q_G + q_P + 1}{|\mathcal{M}|} + 3 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$$

and the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

A proof of this new theorem can be found in Appendix A. We have informed the author of [Höv21b] about our findings and they have confirmed that our improved analysis is correct [Höv22].

3.3.4 Security of the KS transform

Proof Outline In order to prove the security of the KS transformation, we will again define multiple schemes leading up to KS:

- DPKE will be the output of a modified T transformation of PKE as in [HHK17] but also incorporating the public key hash. It is defined in Figure 3.13.
- The KEM scheme KS-INTER is constructed very similarly to how the $\mathsf{U}^{\not\bot}_m$ transformation works in [HHK17]. It is defined in Figure 3.14.
- Another KEM scheme KS' will be obtained using a custom transformation. It is defined in Figure 3.15.
- And finally, KS can be obtained by modifying the message generation of KS'. It is defined in Figure 3.12.

Security Goals These schemes all use independently uniform message distributions, except for the final scheme KS. Nonetheless, we will prove that KS also gives good security guarantees in the ROM. Assuming that PKE is IND-CPA secure and δ-correct, we will show that:

- DPKE is OW-VA secure and δ_1-correct in the ROM where $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$.

```
$\mathsf{KGen}_1(1^{\mathcal{K}})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^{\mathcal{K}})$
return $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Enc}_1(\mathsf{pk}, m)$
$r := G_2(m, H(\mathsf{pk}))$
$c := \mathsf{Enc}(\mathsf{pk}, m; r)$
return $c$

$\mathsf{Dec}_1(\mathsf{sk}, c)$
$m' := \mathsf{Dec}(\mathsf{sk}, c)$
$r := G_2(m', H(\mathsf{pk}))$
if $c \neq \mathsf{Enc}(\mathsf{pk}, m'; r)$ then
    return $\perp$
else return $m'$
```

Figure 3.13: The scheme DPKE where PKE = (KGen, Enc, Dec).

```
$\mathsf{KGen}_I(1^{\mathcal{K}})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^{\mathcal{K}})$
$\overline{\mathsf{sk}} := (\mathsf{sk}, \mathsf{pk})$
return $(\mathsf{pk}, \overline{\mathsf{sk}})$

$\mathsf{Encaps}_I(\mathsf{pk})$
$m \leftarrow_\$ \mathcal{M}$
$c := \mathsf{Enc}_1(\mathsf{pk}, m)$
$K' := G_1(m, H(\mathsf{pk}))$
return $(K', c)$

$\mathsf{Decaps}_I(\overline{\mathsf{sk}} = (\mathsf{sk}, z, \mathsf{pk}), c)$
$m' := \mathsf{Dec}_1(\mathsf{sk}, c)$
if $m' = \perp$ then
    return $\perp$
else
    return $K' := G_1(m', H(\mathsf{pk}))$
```

Figure 3.14: The scheme KS-INTER where DPKE = $(\mathsf{KGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ as in Figure 3.13.

```
$\mathsf{KGen}'(1^{\mathcal{K}})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KGen}(1^{\mathcal{K}})$
$z \leftarrow_\$ \{0,1\}^\ell$
$\overline{\mathsf{sk}} := (\mathsf{sk}, z, \mathsf{pk})$
return $(\mathsf{pk}, \overline{\mathsf{sk}})$

$\mathsf{Encaps}'(\mathsf{pk})$
$m \leftarrow_\$ \mathcal{M}$
$c := \mathsf{Enc}_1(\mathsf{pk}, m)$
$K' := G_1(m, H(\mathsf{pk}))$
$K := KDF(K', H(c))$
return $(K, c)$

$\mathsf{Decaps}'(\overline{\mathsf{sk}} = (\mathsf{sk}, z, \mathsf{pk}), c)$
$m' := \mathsf{Dec}_1(\mathsf{sk}, c)$
if $m' = \perp$ then
    return $K := KDF(z, H(c))$
else
    $K' := G_1(m', H(\mathsf{pk}))$
    return $K := KDF(K', H(c))$
```

Figure 3.15: The scheme KS' where DPKE = $(\mathsf{KGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ as in Figure 3.13.

- KS-INTER is IND-CCA secure and δ_1-correct.
- KS' is IND-CCA secure and δ_1-correct.
- KS is IND-CCA secure and δ_1-correct.

DPKE DPKE, as defined in Figure 3.13, can be understood as the output of a T transformation ([HHK17]) applied to PKE with a special random oracle $G_2'(x) := G_2(x, H(\mathsf{pk}))$. While we cannot construct G_2' independently of the game run, this looks like an arbitrary random oracle to the adversary. The reason is that there is only one generation of pk per game run and it happens before the adversary is first executed, i.e. by the time that the adversary begins executing G_2' is constructed and its outputs are independently uniformly at random from the space $\mathcal{S}$ - like any regular random oracle's. We prove that this addition of static inputs to the random oracle is secure by providing a reduction.

As a consequence, we inherit exactly the same guarantees as with a regular T transformation:

Corollary 3.9 (Security of DPKE in the ROM) *Assume PKE to be γ-spread. Then, for any OW-VA adversary $\mathcal{B}$ against DPKE issuing at most q_{G_2} queries to the random oracle G_2, and q_V queries to a validity checking oracle* Cvo*, there exists an IND-CPA adversary $\mathcal{A}$ against PKE, such that*

$$\mathit{Adv}\,{}^{\mathit{OW\text{-}VA}}_{\mathit{DPKE}}(\mathcal{B}) \leq q_{G_2} \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{q_{G_2}+1}{|\mathcal{M}|} + 3 \cdot \mathit{Adv}\,{}^{\mathit{IND\text{-}CPA}}_{\mathit{PKE}}(\mathcal{A})$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, DPKE is rigid and is δ_1-correct with $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$.

Proof Let G_2' be a random oracle defined as $G_2'(x) := G_2(x, H(\mathsf{pk}))$.

DPKE is the scheme where adversaries have access to G_2. Let DPKE-T be the scheme where adversaries only have access to G_2'. In other words, DPKE-T $:= T[\mathsf{PKE}, G_2']$ using the T transformation from [HHK17]. Both adversaries are allowed to query H on arbitrary inputs as the only output of H that is not completely independent from the security games is $H(\mathsf{pk})$ and its value as well as pk can be regarded as public knowledge.

We first reduce the security of DPKE to that of DPKE-T by providing a proof by reduction.

Let $\mathcal{B}$ be an OW-VA adversary against DPKE issuing at most q_{G_2} queries to the random oracle G_2. We can construct an OW-VA adversary $\mathcal{C}$ against DPKE-T that issues at most q_{G_2} queries to its random oracle G_2': $\mathcal{C}$ simulates an internal random oracle F using on-the-fly simulation (as already seen in e.g. the simulation of H_0 and H_1 in Game G_0 in Figure 3.11). All inputs to $\mathcal{C}$ are forwarded to $\mathcal{B}$, the output of $\mathcal{B}$ is the output of $\mathcal{C}$. Every query that $\mathcal{B}$ makes to Cvo is forwarded to the validity checking oracle available to $\mathcal{C}$. Every query $G_2(x)$ made by $\mathcal{B}$ is resolved as follows:

By definition x is of the form (m, s) for $m \in \mathcal{M}, s \in \mathcal{S}$.

- If $s = H(\mathsf{pk})$, we return $G_2'(m)$ (this is valid as G_2' is available to $\mathcal{C}$).
- Otherwise, the query is forwarded to the internal random oracle F which is not accessible by $\mathcal{B}$.

Since this is a perfect simulation of the OW-VA game towards $\mathcal{B}$, we have:

$$\mathsf{Adv}\,{}^{\mathsf{OW\text{-}VA}}_{\mathsf{DPKE}}(\mathcal{B}) = \mathsf{Adv}\,{}^{\mathsf{OW\text{-}VA}}_{\mathsf{DPKE\text{-}T}}(\mathcal{C})$$

Correctness of DPKE similarly reduces to the correctness of DPKE-T.

This concludes our proof by reduction. The rest of our claim depends on DPKE-T only. The security and correctness of DPKE-T in turn follows directly from Theorem 3.8 together with our assumption about the correctness of PKE.

Because we only require OW-VA security going forward, we disallow use of the plaintext checking oracle and set $q_P = 0$ to get a tighter reduction. In particular this means that there exists an IND-CPA adversary $\mathcal{A}$ against PKE, such that

$$\mathsf{Adv}\,^{\mathsf{OW\text{-}VA}}_{\mathsf{DPKE\text{-}T}}(\mathcal{B}) \leq q_{G_2} \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{q_{G_2} + 1}{|\mathcal{M}|} + 3 \cdot \mathsf{Adv}\,^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$$

Combining the two equations above yields the required bound.

Note that in [HHK17], there was no argument about why the T transformation adds rigidity to a non-rigid scheme. We have already explained why this is the case in the proof of Corollary 3.3. □

KS-INTER The scheme, as defined in Figure 3.14 is a $\mathsf{U}^{\not\perp}_m$ transformation as in [HHK17] with the single change that instead of a random oracle for key generation shared across all users, independent of the public key, we again use a public-key-dependent oracle $G_1'(x) := G_1(x, H(\mathsf{pk}))$ to generate the keys.

With the same reasoning as before, we get:

Corollary 3.10 (Security of KS-INTER in the ROM) *For any IND-CCA adversary $\mathcal{B}$ against KS-INTER, issuing at most q_D queries to the decapsulation oracle* Decaps_I *and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1, there exists an OW-VA adversary $\mathcal{A}$ against DPKE that makes at most q_D queries to the* Cvo *oracle and at most $q_{G_1} + q_{G_2}$ queries to the random oracle G_2 such that*

$$Adv\,^{IND\text{-}CCA}_{KS\text{-}INTER}(\mathcal{B}) \leq Adv\,^{OW\text{-}VA}_{DPKE}(\mathcal{A}) + \delta \cdot (q_{G_2} + 2 \cdot (q_{G_1} + q_D))$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, KS-INTER is δ_1-correct with $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$.

Proof Again, we split this proof into two parts. Let G_1' be a random oracle defined as $G_1'(x) := G_1(x, H(\mathsf{pk}))$. We define the scheme KS-INTER-U to be $\mathsf{U}^{\not\perp}_m[\mathsf{DPKE}, G_1']$ as in [HHK17].

The security of KS-INTER reduces to the security of KS-INTER-U. We provide a proof by reduction.

Let $\mathcal{B}$ be an IND-CCA adversary against KS-INTER issuing at most q_D queries to the decapsulation oracle Decaps_I and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1. We can construct an $\mathsf{IND\text{-}CCA}$ adversary $\mathcal{C}$ against $\mathsf{KS\text{-}INTER\text{-}U}$ that issues at most q_D queries to the decapsulation oracle Decaps_I and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1': $\mathcal{C}$ simulates an internal random oracle F using on-the-fly simulation (as already seen in e.g. the simulation of H_0 and H_1 in Game G_0 in Figure 3.11). All inputs to $\mathcal{C}$ are forwarded to $\mathcal{B}$, the output of $\mathcal{B}$ is the output of $\mathcal{C}$. Every query that $\mathcal{B}$ makes to Decaps_I is forwarded to the decapsulation oracle available to $\mathcal{C}$. Every query $G_1(x)$ made by $\mathcal{B}$ is resolved as follows:

By definition x is of the form (m, s) for $m \in \mathcal{M}, s \in \mathcal{S}$.

- If $s = H(\mathsf{pk})$, we return $G_1'(m)$ (this is valid as G_1' is available to $\mathcal{C}$).
- Otherwise, the query is forwarded to the internal random oracle F which is not accessible by $\mathcal{B}$.

Since this is a perfect simulation of the $\mathsf{OW\text{-}VA}$ game towards $\mathcal{B}$, we have:

$$\mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS\text{-}INTER}}(\mathcal{B}) = \mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS\text{-}INTER\text{-}U}}(\mathcal{C})$$

Correctness of DPKE similarly reduces to the correctness of $\mathsf{KS\text{-}INTER\text{-}U}$.

This concludes our proof by reduction. The rest of our claim depends on $\mathsf{KS\text{-}INTER\text{-}U}$ only. The security and correctness of $\mathsf{KS\text{-}INTER\text{-}U}$ in turn follows directly from [HHK17, Theorem 3.5] because DPKE is δ_1-correct with $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$, deterministic and rigid. Furthermore, $q_{\mathsf{Enc}_1, G_2} = 1$ and $q_{\mathsf{Dec}_1, G_2} = 1$ are upper bounds on the number of G_2 queries that Enc_1, resp. Dec_1 make upon a single invocation. This lets us deduce that there exists an $\mathsf{OW\text{-}VA}$ adversary $\mathcal{A}$ against DPKE that makes at most q_D queries to the Cvo oracle and at most $q_{G_1} + q_{G_2}$ queries to the random oracle G_2 such that

$$\mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS\text{-}INTER\text{-}U}}(\mathcal{B}) \le \mathsf{Adv}\,^{\mathsf{OW\text{-}VA}}_{\mathsf{DPKE}}(\mathcal{A}) + \delta \cdot (q_{G_2} + 2 \cdot (q_{G_1} + q_D))$$

Combining the two equations above yields the required bound. □

KS' Up to this point, we did not have to change a lot in comparison to the transformations in [HHK17]. For this scheme however, an idea from Kathrin Hövelmanns' [Höv21a] comes into play: $\mathsf{KS\text{-}INTER}$ is a secure scheme with explicit rejection which makes it easier for us to reduce the security of KS' to that of DPKE. Our reduction will introduce the secret z and move away from explicit rejection as well as change the way that the keys are generated by simulating the correct generation involving *KDF*.

Theorem 3.11 (Security of KS′ in the ROM) *For any IND-CCA adversary $\mathcal{B}$ against KS', issuing at most q_D queries to the decapsulation oracle* Decaps', *at most q_{KDF} queries to the random oracle KDF and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1, there exists an IND-CCA adversary $\mathcal{A}$ against*

KS-INTER that makes at most q_D queries to the Decaps_I *oracle and at most q_{G_2}, resp. q_{G_1} queries to the random oracles G_2 and G_1 such that*

$$Adv\,^{IND\text{-}CCA}_{KS'}(\mathcal{B}) \leq Adv\,^{IND\text{-}CCA}_{KS\text{-}INTER}(\mathcal{A}) + \frac{q_{KDF}}{2^{\ell}}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, KS' is δ_1-correct with $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$.

Proof The proof proceeds similarly in spirit to the proofs for [HHK17, Theorem 3.5] and Theorem 3.1. We adapt it here to our new situation.

It is easy to verify the correctness bound as KS-INTER is also δ_1-correct. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KS', issuing at most q_{KDF} queries to the random oracle *KDF*. Consider the games given in Figure 3.16.

Game G_0 Since game G_0 is the original IND-CCA game w.r.t. KS',

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS'}}(\mathcal{B})$$

Game G_1 In game G_1 we make two changes. First, we raise flag QUERY and abort if $KDF(K'', H(c^*))$ is queried (lines 16 and 17).

This change does not affect the queries to H even if the oracles H and *KDF* were identical because all other queries to H whose output is ever used are queries with domains $\mathcal{PK}$ or $\mathcal{C}$ and the domain of the *KDF* queries we modify is $\mathcal{K}' \times \mathcal{S}$. Thus, we have domain separation between the H queries and the modified *KDF* queries since we assumed $(\mathcal{K}' \times \mathcal{S}) \cap \mathcal{PK} = \emptyset$ and $(\mathcal{K}' \times \mathcal{S}) \cap \mathcal{C} = \emptyset$ in section 3.3.1. If the oracles H and *KDF* are independent, our modification trivially has no effect on the H queries.

Second, we make the perfectly random key K_1^* only pseudorandom. That is, in the setup procedure, we replace $K_1^* \leftarrow_{\$} \{0,1\}^n$ by $K_1^* := KDF(K'', H(c^*))$, where K'' is a random value from $\{0,1\}^{\ell}$. The latter remains unnoticed by $\mathcal{B}$ unless $KDF(K'', H(c^*))$ is queried, in which case G_1 aborts. Let the event of $\mathcal{B}$ querying $KDF(K'', H(c^*))$ be QUERY. This means

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_0^{\mathcal{B}} \Rightarrow 1]\right| \leq \Pr[\mathsf{QUERY}]$$

Here, $\mathcal{B}$'s view in Game G_1 is not completely independent of K'', in the sense that had we chosen a different K'' and changed nothing else, $\mathcal{B}$ would notice because K_1^* changed. However, K'' is used only in the computation of K_1^* and K_1^* is by definition a perfectly random value. Notice however, that the event QUERY is well-defined already in Game G_0 and only occurs if $\mathcal{B}$ can guess K''. Due to the difference lemma, the probability of QUERY is the same in both games and in Game G_0, $\mathcal{B}$'s view is completely independent of the uniform secret K''. Therefore, the probability of QUERY is at most $q_{KDF} \cdot 2^{-\ell}$ and hence,

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_0^{\mathcal{B}} \Rightarrow 1]\right| \leq \frac{q_{KDF}}{2^{\ell}}$$

It remains to bound $\Pr[G_1^{\mathcal{B}} \Rightarrow 1]$. To this end, we construct an adversary $\mathcal{A}$ against the IND-CCA security of KS-INTER, simulating G_1 for $\mathcal{B}$ as in Figure 3.17.

Notice that if $b = 0$ then $\mathcal{A}$ receives $K_0'^{*} = G_1(m^*, H(\mathsf{pk}))$ for some message m^*. $\mathcal{A}$ then computes $KDF(K_0'^{*}, H(c^*))$ where $c^* = \mathsf{Enc}_I(\mathsf{pk}, m^*)$ and uses this as K_0^* for $\mathcal{B}$. This simulates perfectly the key generation for the case $b = 0$ in Game G_1.

If however, $b = 1$ then $\mathcal{A}$ receives $K_1'^{*}$, a value sampled uniformly at random. Now $\mathcal{A}$ again computes $KDF(K_1'^{*}, H(c^*))$, again for $c^* = \mathsf{Enc}_I(\mathsf{pk}, m^*)$ and some m^* but this time c^* is fully independent from $K_1'^{*}$. $\mathcal{A}$ then uses this newly computed value as K_1^* for $\mathcal{B}$. This simulates perfectly the key generation for the case $b = 1$ in Game G_1.

A is able to perfectly simulate Game G_1 towards $\mathcal{B}$ because it has all the necessary information to convert the keys (as seen above), it recognizes when its decapsulation oracle rejects (because KS-INTER is an explicit-rejection scheme) and it can pick z arbitrarily because KS-INTER does not use it. Also note that $\mathcal{A}$ wins its game against KS-INTER if and only if $\mathcal{B}$ wins the simulated game against KS'. Hence,

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS\text{-}INTER}}(\mathcal{A})$$

Collecting the probabilities and applying the reverse triangle inequality yields the required bound.

KS Finally, we only need to adapt the message generation from KS' (line 5 in Figure 3.15) to instead use the hash of the uniformly generated message (line 6 in Figure 3.12). This is the only difference between the schemes KS' and KS. For a random oracle, the hash of a uniformly sampled input will also be uniformly distributed in the range. Hence, we don't need to introduce our notion of non-uniform message generation here but we do need to isolate the hash operation from the rest of the game by forbidding (unlikely) queries about the uniformly generated message.

We can prove the following statement:

```
GAMES $G_0$ and $G_1$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
$z \leftarrow_\$ \{0,1\}^\ell$
$\overline{\mathsf{sk}} := (\mathsf{sk}, z, \mathsf{pk})$
$m^* \leftarrow_\$ \mathcal{M}$
$c^* := \mathsf{Enc}_I(\mathsf{pk}, m^*)$
$K' := G_1(m^*, H(\mathsf{pk}))$
$K'' \leftarrow_\$ \{0,1\}^\ell$
$K_0^* := KDF(K', H(c^*))$
$K_1^* \leftarrow_\$ \{0,1\}^n$                 // $G_0$
$K_1^* := KDF(K'', H(c^*))$              // $G_1$
$b \leftarrow_\$ \{0,1\}$
$b' \leftarrow \mathcal{B}^{\mathsf{Decaps}', KDF}(\mathsf{pk}, c^*, K_b^*)$
return $[\![b' = b]\!]$

$KDF(k', ch)$      // $(k', ch) \in \{0,1\}^\ell \times \mathcal{S}$
if $\exists K$ s.th. $(k', ch, K) \in \mathfrak{L}_{KDF}$
    return $K$
if $k' = K''$ then                       // $G_1$
    QUERY := true; abort                 // $G_1$
$K \leftarrow_\$ \mathcal{K}$
$\mathfrak{L}_{KDF} := \mathfrak{L}_{KDF} \cup \{(k', ch, K)\}$
return $K$
```

Figure 3.16: Games G_0 and G_1 for the proof of Theorem 3.11

```
$\mathcal{A}(\mathsf{pk}, c^*, K_b'^*)$
$K_b^* := KDF(K_b'^*, H(c^*))$
$z \leftarrow_\$ \{0,1\}^\ell$
$b' \leftarrow \mathcal{B}^{\mathsf{Decaps}'', KDF}(\mathsf{pk}, c^*, K_b^*)$
return $b'$

$\mathsf{Decaps}''(c \neq c^*)$
$K' := \mathsf{Decaps}_I(c)$
if $K' = \perp$ then
    return $K := KDF(z, H(c))$
else return $K := KDF(K', H(c))$
```

Figure 3.17: Adversary $\mathcal{A}$ against IND-CCA w.r.t. KS-INTER for the proof of Theorem 3.11, where *KDF* is defined as in Game G_0 of Figure 3.16

Theorem 3.12 (Security of KS in the ROM) *For any IND-CCA adversary $\mathcal{B}$ against KS, issuing at most q_D queries to the decapsulation oracle* Decaps*, at most q_H queries to the random oracle H, at most q_{KDF} queries to the random oracle KDF and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1, there exists an IND-CCA adversary $\mathcal{A}$ against KS' that makes at most q_D queries to the* Decaps' *oracle, at most q_{KDF} queries to the random oracle KDF and at most q_{G_2}, resp. q_{G_1} queries to the random oracles G_2 and G_1 such that*

$$Adv_{KS}^{IND\text{-}CCA}(\mathcal{B}) \leq Adv_{KS'}^{IND\text{-}CCA}(\mathcal{A}) + \frac{q_H}{|\mathcal{M}|}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, KS is δ_1-correct with $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$.

Proof The proof will use a single game hop to forbid queries of the form $H(m'^*)$.

It is easy to verify the correctness bound as $\mathsf{KS'}$ is also δ_1-correct. Let $\mathcal{B}$ be an adversary against the $\mathsf{IND\text{-}CCA}$ security of KS, issuing at most q_H queries to the random oracle H. Consider the games given in Figure 3.18.

Game G_0 Since game G_0 is the original $\mathsf{IND\text{-}CCA}$ game,

$$\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2} = \mathsf{Adv}\,_{\mathsf{KS}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B})$$

Game G_1 In game G_1 we make two changes. First, we raise flag QUERY and abort if $H(m'^*)$ is queried (lines 15 and 16).

This change does not affect the queries to *KDF* even if the oracles H and *KDF* were identical because all other queries to *KDF* whose output is ever used are queries with domains $\mathcal{K}' \times \mathcal{S}$ and the domain of the H queries we modify is $\mathcal{M}$. Thus, we have domain separation between the *KDF* queries and the modified H queries since we assumed $(\mathcal{K}' \times \mathcal{S}) \cap \mathcal{M} = \emptyset$ in section 3.3.1. If the oracles H and *KDF* are independent, our modification trivially has no effect on the *KDF* queries.

No matter whether the oracles H and *KDF* are identical or whether they are independent, our modification to the H queries also has no effect on other H queries we use because all other queries to H whose output is ever used are queries with domains $\mathcal{PK}$ or $\mathcal{C}$ and the domain of the H queries we modify is $\mathcal{M}$. We have domain separation here as well since we assumed $\mathcal{PK} \cap \mathcal{M} = \emptyset$ and $\mathcal{C} \cap \mathcal{M} = \emptyset$ in section 3.3.1.

Second, we make the pseudorandom message output by $H(m'^*)$ on line 5 perfectly random. That is, in the setup procedure, we replace $m^* := H(m'^*)$ by $m^* \leftarrow_\$ \mathcal{M}$. The latter remains unnoticed by $\mathcal{B}$ unless $H(m'^*)$ is queried, in which case G_1 aborts. Since $\mathcal{B}$'s view is independent of (the uniform secret) m'^* unless G_1 aborts (B only gets the inputs c^*, K_b^* and pk which do depend only on m^* but not on m'^*),

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_0^{\mathcal{B}} \Rightarrow 1]\right| \leq \frac{q_H}{|\mathcal{M}|}$$

As the Game G_1 is exactly the $\mathsf{IND\text{-}CCA}$ game against $\mathsf{KS'}$, we can construct the adversary $\mathcal{A}$ against $\mathsf{KS'}$ as simply running $\mathcal{B}$ while simulating the game G_1 towards $\mathcal{B}$ and copying $\mathcal{B}$'s outputs. Hence, there exists an adversary $\mathcal{A}$ against $\mathsf{KS'}$ such that

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathsf{Adv}\,_{\mathsf{KS'}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$$

Collecting the probabilities and applying the reverse triangle inequality yields the required bound.

```
GAMES $G_0$ and $G_1$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\mathcal{K}})$
    $z \leftarrow_{\$} \{0,1\}^{\ell}$
    $\overline{\mathsf{sk}} := (\mathsf{sk}, z, \mathsf{pk})$
    $m'^{*} \leftarrow_{\$} \mathcal{M}$
    $m^{*} := H(m'^{*})$                          // $G_0$
    $m^{*} \leftarrow_{\$} \mathcal{M}$                     // $G_1$
    $c^{*} := \mathsf{Enc}_I(\mathsf{pk}, m^{*})$
    $K' := G_1(m^{*}, H(\mathsf{pk}))$
    $K_0^{*} := KDF(K', H(c))$
    $K_1^{*} \leftarrow_{\$} \{0,1\}^{n}$
    $b \leftarrow_{\$} \{0,1\}$
    $b' \leftarrow \mathcal{B}^{\mathsf{Decaps},H}(\mathsf{pk}, c^{*}, K_b^{*})$
    return $[\![b' = b]\!]$

$H(m)$                                            // $m \in \mathcal{M}$
    if $\exists X$ s.th. $(m, X) \in \mathfrak{L}_H$
        return $X$
    if $m = m'^{*}$ then                          // $G_1$
        QUERY := true; abort                      // $G_1$
    $X \leftarrow_{\$} \mathcal{M}$
    $\mathfrak{L}_H := \mathfrak{L}_H \cup \{(m, X)\}$
    return $X$                                    □
```

Figure 3.18: Games G_0 and G_1 for the proof of Theorem 3.12

As this was the last reduction needed, we can now collect all the bounds in a final corollary.

Corollary 3.13 (Security of the full KS reduction in the ROM) *Assume PKE to be γ-spread. Then, for any IND-CCA adversary $\mathcal{B}$ against KS issuing at most q_D queries to the decapsulation oracle* Decaps*, at most q_H queries to the random oracle H, at most q_{KDF} queries to the random oracle KDF and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1, there exists an IND-CPA adversary $\mathcal{A}$ against PKE, such that*

$$\begin{aligned} Adv_{KS}^{IND\text{-}CCA}(\mathcal{B}) \leq \quad & (3 \cdot q_{G_1} + 2 \cdot q_{G_2} + 2 \cdot q_D) \cdot \delta + q_D \cdot 2^{-\gamma} + \frac{q_{KDF}}{2^{\ell}} \\ & + \frac{q_H + q_{G_2} + q_{G_1} + 1}{|\mathcal{M}|} + 3 \cdot Adv_{PKE}^{IND\text{-}CPA}(\mathcal{A}) \end{aligned}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$. Furthermore, if PKE is δ-correct then KS is δ_1-correct with $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$.

Proof This follows from Theorems and Corollaries 3.9, 3.10, 3.11 and 3.12.

Given adversary $\mathcal{B}$ against KS, we construct an IND-CCA adversary $\mathcal{A}_1$ against KS' as described in Theorem 3.12. Given adversary $\mathcal{A}_1$ against KS', we then construct another IND-CCA adversary $\mathcal{A}_2$ against KS-INTER as in Figure 3.17. Given adversary $\mathcal{A}_2$ against KS-INTER, we also construct an OW-VA adversary $\mathcal{A}_3$ against DPKE just as in [HHK17, Figure 17]. And finally, we employ Theorem 3.9 to get the claimed adversary $\mathcal{A}$ against the IND-CPA security of PKE.

From our constructions as well as Theorem 3.9, 3.10, 3.11 and 3.12 respectively, we conclude:

$$\begin{aligned}
\mathsf{Adv}\,^{\mathsf{OW\text{-}VA}}_{\mathsf{DPKE}}(\mathcal{A}_3) &\leq q_{G_2} \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{q_{G_2}+1}{|\mathcal{M}|} + 3 \cdot \mathsf{Adv}\,^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) \\
\mathsf{Adv}\,^{\mathsf{OW\text{-}VA}}_{\mathsf{KS\text{-}INTER}}(\mathcal{A}_2) &\leq \mathsf{Adv}\,^{\mathsf{OW\text{-}VA}}_{\mathsf{DPKE}}(\mathcal{A}_3) + \delta \cdot (q_{G_2} + 2 \cdot (q_{G_1} + q_D))) \\
\mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS'}}(\mathcal{A}_1) &\leq \mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS\text{-}INTER}}(\mathcal{A}_2) + \frac{q_{KDF}}{2^{\ell}} \\
\mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS}}(\mathcal{B}) &\leq \mathsf{Adv}\,^{\mathsf{IND\text{-}CCA}}_{\mathsf{KS'}}(\mathcal{A}_1) + \frac{q_H}{|\mathcal{M}|}
\end{aligned}$$

where the numbers of queries refer to the queries made by the adversary in the left-hand side of each line (e.g. in the first line q_{G_2} denotes the number of queries to G_2 made by $\mathcal{A}_3$). We now have to establish connections between these query numbers.

Assume, as in the corollary, that the adversary $\mathcal{B}$ makes at most q_D queries to the decapsulation oracle Decaps, at most q_H queries to the random oracle H, at most q_{KDF} queries to the random oracle *KDF* and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1.

According to our constructions, we can get the following bounds on the number of relevant queries for each adversary:

- $\mathcal{A}_1$ makes the same number of queries to all oracles as $\mathcal{B}$
- $\mathcal{A}_2$ makes the same number of queries to the oracles Decaps_I, G_2 and G_1
- $\mathcal{A}_3$ makes $q_{G_2} + q_{G_1}$ queries to the oracle G_2 and q_D queries to the Cvo oracle

Combining the inequalities above and the amounts of queries yields our claimed bound. □

3.3.5 Security of the concrete schemes

For the concrete schemes CRYSTALS-KYBER and SABER, given the instantiations described in Section 3.3.2, we now get the following bounds.

Corollary 3.14 (Security of CRYSTALS-KYBER in the ROM) *Assume* Kyber.*CPAPKE to be γ-spread. Then, for any IND-CCA adversary $\mathcal{B}$ against* Kyber.*CCAKEM issuing at most q_D queries to the decapsulation oracle* Decaps*, at most q_H queries to the random oracle H, at most q_{KDF} queries to the random oracle KDF and at most q_{G_2}, resp. q_{G_1} queries to its random oracles G_2 and G_1, there* *exists an* $\mathsf{IND\text{-}CPA}$ *adversary* $\mathcal{A}$ *against* Kyber.CPAPKE*, such that*

$$\begin{aligned} \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\text{Kyber}.\mathsf{CCAKEM}}(\mathcal{B}) \leq \quad & (3 \cdot q_{G_1} + 2 \cdot q_{G_2} + 2 \cdot q_D) \cdot \delta + q_D \cdot 2^{-\gamma} \\ & + \frac{q_{KDF} + q_H + q_{G_2} + q_{G_1} + 1}{2^{256}} + 3 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\text{Kyber}.\mathsf{CPAPKE}}(\mathcal{A}) \end{aligned}$$

and the running time of $\mathcal{A}$ *is about that of* $\mathcal{B}$*. Furthermore, if* Kyber.CPAPKE *is* δ*-correct then* Kyber.CCAKEM *is* δ_1*-correct with* $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$*.*

Proof This follows directly from Corollary 3.13 together with parts of the instantiations explained in Section 3.3.2, namely:

- PKE with Kyber.CPAPKE from [SAB+20]
- ℓ with $\ell = 256$
- $\mathcal{M}$ with $\{0,1\}^{256}$

and the fact that the CRYSTALS-KYBER KEM, Kyber.CCAKEM, is constructed as a KS transform. □

Corollary 3.15 (Security of SABER in the ROM) *Assume* Saber.PKE *to be* γ*-spread. Then, for any* $\mathsf{IND\text{-}CCA}$ *adversary* $\mathcal{B}$ *against* Saber.KEM *issuing at most* q_D *queries to the decapsulation oracle* Decaps*, at most* q_H *queries to the random oracle* $H = KDF$ *and at most* q_{G_2}*, resp.* q_{G_1} *queries to its random oracles* G_2 *and* G_1*, there exists an* $\mathsf{IND\text{-}CPA}$ *adversary* $\mathcal{A}$ *against* Saber.PKE*, such that*

$$\begin{aligned} \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{Saber.KEM}}(\mathcal{B}) \leq \quad & (3 \cdot q_{G_1} + 2 \cdot q_{G_2} + 2 \cdot q_D) \cdot \delta + q_D \cdot 2^{-\gamma} \\ & + \frac{2q_H + q_{G_2} + q_{G_1} + 1}{2^{256}} + 3 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Saber.PKE}}(\mathcal{A}) \end{aligned}$$

and the running time of $\mathcal{A}$ *is about that of* $\mathcal{B}$*. Furthermore, if* Saber.PKE *is* δ*-correct then* Saber.KEM *is* δ_1*-correct with* $\delta_1(q_{G_2}) := q_{G_2} \cdot \delta$*.*

Proof This follows directly from Corollary 3.13 together with parts of the instantiations explained in Section 3.3.2, namely:

- PKE with Saber.PKE from [DKR+20]
- The fact that H and KDF are instantiated as the same underlying function (hence $q_{KDF} = q_H$)
- ℓ with $\ell = 256$
- $\mathcal{M}$ with $\{0,1\}^{256}$

and the fact that the SABER KEM, Saber.KEM, is constructed as a KS transform. □

This concludes our discussion of CRYSTALS-KYBER and SABER as we have now successfully reduced the IND-CCA security of both of them (as instantiations of the KS transformation) to the IND-CPA security of their respective underlying PKE schemes. Note that for our work to be meaningful, there must still be formal proofs justifying the δ-correctness and IND-CPA security of those PKE schemes (which can be reduced to the underlying lattice assumptions) as well as the fact that they are indeed γ-spread, all with acceptable parameters.

3.4 Summary of Results

In this final section of the thesis, we will now summarize our results and compare the bounds we found to the bounds presented in the specifications (if any). Table 3.1 shows the bounds in the specification documents. Table 3.2 displays our results. Table 3.3 gives an overview of the assumptions we made for our bounds as well as the correctness we were able to prove for the respective KEMs.

In all entries, $\mathcal{B}$ is an IND-CCA adversary against the KEM scheme and $\mathcal{A}$ an IND-CPA or OW-CPA adversary against the PKE scheme. The bounds are understood with the quantifications $\forall\mathcal{B}\exists\mathcal{A}$ and the parameters q_{RO}, q_D are upper bounds on the respective queries made by $\mathcal{B}$: q_{RO} is an upper bound on the total number of queries to all different random oracles involved (more precise bounds are available in each of the respective sections) and q_D is a bound on the number of decapsulation queries. The running time of $\mathcal{A}$ is about that of $\mathcal{B}$.

Introducing a single bound q_{RO} for the sum of queries to all random oracles causes a certain loss of tightness compared to the earlier results. Nevertheless, we did feel that it aids readability and makes it easier to compare the different schemes. For the most precise bounds presented in this work, we refer the reader to the earlier sections cited in Table 3.2.

In our work, we have managed to provide full reductions from IND-CCA security of the KEMs to OW-CPA or IND-CPA security of the underlying PKE schemes. We have stated the resulting bounds in terms of concrete security. Our work reduces all KEM schemes to the respective PKE scheme outlined in the specification documents except for Classic McEliece which we chose to reduce even further to the Niederreiter PKE using Goppa Codes.

All of our reductions are tight in the ROM.

In the case of NTRU, we were only able to reduce to a non-standard notion of one-wayness due to the non-uniform message distribution. All other reductions use a standard notion.

For NTRU we also require rigidity of the underlying PKE scheme as NTRU does not explicitly add a re-encryption check as part of the scheme like the other three schemes we considered.

Scheme	Bound in specification
NTRU [CDH+20, Sec. 5.1]	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) + \frac{q_{RO}}{2^{256}}$
Classic McEliece [ABC+20, Sec. 6.3]	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) + q_D \cdot 2^{-256} + \frac{q_D}{\lvert\mathcal{M}\rvert}$
CRYSTALS-KYBER [SAB+20, Sec. 4.3.1]	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) + 4 \cdot q_{RO} \cdot \delta$
SABER [DKR+20, Thm. 6.3]	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq 3 \cdot \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) + q_{RO} \cdot \delta + (2q_{RO} + 1) \cdot 2^{-256}$

Table 3.1: The security bounds from the respective specifications. While NTRU did not explicitly state a bound in terms of concrete security, its specification document cited three possible sources for such a bound. Among them was [HHK17, Theorem 3.6], whose bound we adapted to the situation here (the uniform secret s is sampled from $\{0,1\}^\ell$ instead of $\mathcal{M}$ and we use the correction from [Höv21b] to replace q_D by q_{RO}).

Scheme	Bound in our work
NTRU, see Cor. 3.2	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq \mathsf{Adv}_{\mathsf{PKE,\ Sample\_rm}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) + \frac{q_{RO}}{2^{256}}$
Classic McEliece, see Cor. 3.7	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) + \frac{q_{RO}}{2^{256}}$
CRYSTALS-KYBER, see Cor. 3.14 SABER, see Cor. 3.15	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}) \leq 3 \cdot \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) + q_D \cdot 2^{-\gamma} + \frac{2q_{RO}+1}{2^{256}} + (3q_{RO} + 2 \cdot q_D) \cdot \delta$

Table 3.2: The security bounds we proved in this work. CRYSTALS-KYBER and SABER share the same bound in the table but for the case of CRYSTALS-KYBER, the bound can be improved by replacing the term $(2q_{RO}+1)/2^{256}$ by simply $(q_{RO}+1)/2^{256}$.

	Underlying PKE			KEM
Scheme	δ-correct	Rigidity	γ-spread	δ_1-correct with
NTRU	For $\delta = 0$	Det. and rigid	Not required	$\delta_1(q_{RO}) = 0$
Classic McEliece	For $\delta = 0$	Det. only	Not required	$\delta_1(q_{RO}) = 0$
CRYSTALS-KYBER	For some δ	Neither	For some γ	$\delta_1(q_{RO}) = \delta \cdot q_{RO}$
SABER	For some δ	Neither	For some γ	$\delta_1(q_{RO}) = \delta \cdot q_{RO}$

Table 3.3: The assumptions under which our bounds in Table 3.2 are valid as well as the resulting correctness of the KEM. These are stated explicitly in the Corollaries and their corresponding proofs cited in Table 3.2. The second column describes a maximum acceptable correctness error δ, the third column states whether we require determinism, rigidity, both or neither and the fourth column states whether or not we require γ-spreadness.

For Classic McEliece we arrived at a similarly tight bound as the specification stated but we ended up with different additional bounds (namely, we used q_{RO} and not q_D). This difference could simply stem from the different frameworks employed in the respective analyses but it is noteworthy nonetheless.

For CRYSTALS-KYBER and SABER we were forced to introduce the additional assumption of γ-spreadness to ensure that our reductions applied because we relied on an intermediate scheme with explicit rejection for our analysis. The NTRU and Classic McEliece reductions have no such requirement since their PKE scheme was already deterministic.

Finally, we have improved on the bound given in [HHK17, Theorem 3.2] and [Höv21b, Theorem 2.1.3] as Theorem 3.8.

Appendix A

Proof of Theorem 3.8

We will now prove our claim that Theorem 3.8 is indeed correct and provides a tighter bound than [Höv21b, Theorem 2.1.3] (the corrected version of [HHK17, Theorem 3.2]).

For the reader's convenience, we will now restate the original and outline the relevant parts of the proof in [Höv21b]. The original theorem stated:

Theorem (PKE IND-CPA $\overset{\text{ROM}}{\Longrightarrow}$ T[PKE, G] OW-PCVA) *Let PKE' := T[PKE, G] in the following. Assume PKE to be δ-correct and γ-spread. Then, for any OW-PCVA adversary $\mathcal{B}$ that issues at most q_G queries to the random oracle G, q_P queries to a plaintext checking oracle* Pco, *and q_V queries to a validity checking oracle* Cvo, *there exists an IND-CPA adversary $\mathcal{A}$ such that*

$$\mathit{Adv}_{\mathit{PKE'}}^{\mathit{OW\text{-}PCVA}}(\mathcal{B}) \leq (q_G + q_P) \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{2(q_G + q_P) + 1}{|\mathcal{M}|} + 3 \cdot \mathit{Adv}_{\mathit{PKE}}^{\mathit{IND\text{-}CPA}}(\mathcal{A})$$

and the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Note that the only difference between this and Theorem 3.8 is the factor of 2 in the second to last term. We will not change the argument presented in the original proof but instead only provide a better analysis on one of the events.

Proof The proof in [Höv21b] first argues that there exists an adversary $\mathcal{C}$ such that

$$\mathsf{Adv}_{\mathsf{PKE'}}^{\mathsf{OW\text{-}PCVA}}(\mathcal{B}) \leq (q_G + q_P) \cdot \delta + q_V \cdot 2^{-\gamma} + \frac{1}{|\mathcal{M}|} + 3 \cdot \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{C}) + \Pr[\mathsf{QUERY}] \tag{A.1}$$

In the proof, an IND-CPA adversary $\mathcal{D}$ is constructed which in turn uses $\mathcal{B}$. During a game run, two messages m_0^* and m_1^* are sample independently and uniformly at random. The event QUERY mentioned in the inequality above describes the event that adversary $\mathcal{B}$ queries the random oracle G on the input m_b^*, i.e. on the input for which $\mathcal{B}$ is given an encryption.

The other message m_{1-b}^* is independent from $\mathcal{B}$'s view. Let BADG be the event of $\mathcal{B}$ nevertheless querying G on m_{1-b}^*.

We will now improve on the bound for $\Pr[\mathsf{QUERY}]$ that is given in [Höv21b].

First notice that because m_{1-b}^* is independent from $\mathcal{B}$'s view and by definition we have the following two equations

$$\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{D}) = \left|\Pr[b = b'] - \frac{1}{2}\right| \tag{A.2}$$

$$\Pr[\mathsf{BADG}] \leq \frac{q_G + q_P}{|\mathcal{M}|} \tag{A.3}$$

From the way the adversary $\mathcal{D}$ is defined in the proof, we can calculate its advantage in all cases:

$$\begin{aligned}
\Pr[b = b' \mid \mathsf{QUERY} \wedge \mathsf{BADG}] &= \frac{1}{2} \\
\Pr[b = b' \mid \mathsf{QUERY} \wedge \overline{\mathsf{BADG}}] &= 1 \\
\Pr[b = b' \mid \overline{\mathsf{QUERY}} \wedge \mathsf{BADG}] &= 0 \\
\Pr[b = b' \mid \overline{\mathsf{QUERY}} \wedge \overline{\mathsf{BADG}}] &= \frac{1}{2}
\end{aligned}$$

Combining these equations yields:

$$\Pr[b = b' \mid \mathsf{BADG}] = \frac{1}{2}\Pr[\mathsf{QUERY} \mid \mathsf{BADG}] \tag{A.4}$$

$$\begin{aligned}
\Pr[b = b' \mid \overline{\mathsf{BADG}}] &= \Pr[\mathsf{QUERY} \mid \overline{\mathsf{BADG}}] + \frac{1}{2}\Pr[\overline{\mathsf{QUERY}} \mid \overline{\mathsf{BADG}}] \\
&= \frac{1}{2}\Pr[\mathsf{QUERY} \mid \overline{\mathsf{BADG}}] + \frac{1}{2}
\end{aligned} \tag{A.5}$$

And with a final calculation, we can bound the advantage of $\mathcal{D}$:

$$\begin{aligned}
\mathsf{Adv}\,^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{D}) &\overset{(A.2)}{=} \left|\Pr[b = b'] - \frac{1}{2}\right| \\
&= \left|\Pr[b = b' \mid \overline{\mathsf{BADG}}]\Pr[\overline{\mathsf{BADG}}] + \Pr[b = b' \mid \mathsf{BADG}]\Pr[\mathsf{BADG}] - \frac{1}{2}\right| \\
&\overset{(A.4),(A.5)}{=} \left|\left(\frac{1}{2}\Pr[\mathsf{QUERY} \mid \overline{\mathsf{BADG}}] + \frac{1}{2}\right)\Pr[\overline{\mathsf{BADG}}]\right. \\
&\qquad \left. + \left(\frac{1}{2}\Pr[\mathsf{QUERY} \mid \mathsf{BADG}]\right)\Pr[\mathsf{BADG}] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\left(\Pr[\mathsf{QUERY}] - \Pr[\mathsf{BADG}]\right)\right| \\
&\geq \frac{1}{2}\left(\Pr[\mathsf{QUERY}] - \Pr[\mathsf{BADG}]\right) \\
&\overset{(A.3)}{\geq} \frac{1}{2}\Pr[\mathsf{QUERY}] - \frac{q_G + q_P}{2|\mathcal{M}|}
\end{aligned}$$

And so we get the following bound for $\Pr[\mathsf{QUERY}]$:

$$\mathsf{Adv}\,^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{D}) + \frac{q_G + q_P}{\mathbf{2}|\mathcal{M}|} \geq \frac{1}{2}\Pr[\mathsf{QUERY}]$$

which is tighter than the bound provided in the two papers [HHK17, Höv21b] by a factor of 2 in the additive term.

Plugging the improved bound for $\Pr[\mathsf{QUERY}]$ back into (A.1) finishes our proof. □

Bibliography

[AAC+22] Gorjan Alagic, Daniel Apon*, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. Technical report, National Institute of Standards and Technology, 2022. available at https://doi.org/10.6028/NIST.IR.8413.

[AASA+20] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. Technical report, National Institute of Standards and Technology, 2020. available at https://doi.org/10.6028/NIST.IR.8309.

[ABC+20] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[BP18] Daniel J. Bernstein and Edoardo Persichetti. Towards KEM Unification, 2018. available at https://cr.yp.to/papers/tightkem-20180528.pdf.

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

[CDH+20] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[CJL+16] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on Post-Quantum Cryptography. Technical report, National Institute of Standards and Technology, 2016. available at http://dx.doi.org/10.6028/NIST.IR.8105.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[DKR+20] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

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

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

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on Theory of Computing*, pages 212–219, Philadephia, PA, USA, May 22–24, 1996. ACM Press. doi:10.1145/237814.237866.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-70500-2_12.

[Höv21a] Kathrin Hövelmanns. personal communication, March 2021.

[Höv21b] Kathrin Hövelmanns. *Generic constructions of quantum-resistant cryptosystems*. doctoralthesis, Ruhr-Universität Bochum, Universitätsbibliothek, 2021. doi:10.13154/294-7758.

[Höv22] Kathrin Hövelmanns. personal communication, March 2022.

[MVDJ18] Vasileios Mavroeidis, Kamer Vishi, Mateusz D., and Audun Jøsang. The impact of quantum computing on present cryptography. *International Journal of Advanced Computer Science and Applications*, 9(3), 2018. URL: https://doi.org/10.14569%2Fijacsa.2018.090354, doi:10.14569/ijacsa.2018.090354.

[SAB$^+$20] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. arXiv:https://doi.org/10.1137/S0036144598347011, doi:10.1137/S0036144598347011.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. https://eprint.iacr.org/2004/332.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-78372-7_17.

[Xag22] Keita Xagawa. Anonymity of NIST PQC round 3 KEMs. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 551–581, Trondheim, Norway, May 30 – June 3, 2022. Springer, Heidelberg, Germany. doi:10.1007/978-3-031-07082-2_20.

[Yas21] Kenji Yasunaga. Replacing Probability Distributions in Security Games via Hellinger Distance. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2021/14336, doi:10.4230/LIPIcs.ITC.2021.17.