varun maram

GENERIC ENHANCEMENTS OF POST-QUANTUM PUBLIC-KEY ENCRYPTION

diss. eth no. 29889

DISS. ETH NO. 29889

GENERIC ENHANCEMENTS OF POST-QUANTUM PUBLIC-KEY ENCRYPTION

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

VARUN MARAM
M.Sc. ETH in Computer Science, ETH Zurich

born on 4 January 1996
citizen of India

accepted on the recommendation of

Prof. Dr. Kenneth G. Paterson, examiner
Prof. Dr. Dennis Hofheinz, co-examiner
Prof. Dr. Andreas Hülsing, co-examiner

2023

To my parents.

ABSTRACT

Due to the threat of quantum computers breaking most widely-deployed public-key cryptography, standards bodies worldwide such as the US-based National Institute of Standards and Technology (NIST), the International Standards Organization (ISO), the Internet Engineering Task Force (IETF), and the European Telecommunications Standards Institute (ETSI) have already started working on standardizing quantum-resistant cryptographic primitives and algorithms.

In particular, NIST is in the process of standardizing digital signatures and public-key encryption (PKE) schemes. Focusing on the latter class of primitives, an important criterion of evaluation set by NIST for PKE schemes in its post-quantum cryptography (PQC) standardization process is on whether they achieve the traditional notion of IND-CCA security. However, a myriad of cryptographic applications have emerged in recent times which not only require properties beyond IND-CCA security from the underlying PKE schemes, but also require advanced functionalities from such primitives. Since the NIST PQC standards for PKE are intended to be widely used for decades to come, it is hence important to analyze such schemes through the lens of "beyond IND-CCA" security properties and "beyond basic PKE" functionalities.

This dissertation presents the following three contributions with respect to generically enhancing quantum-resistant PKE schemes in the above ways – especially in the context of NIST's standardization efforts.

- First, we revisit the main target of IND-CCA security, and examine the generic methods employed by certain important NIST PQC candidates to achieve this notion. In particular, we study *Kyber*, the current standard chosen by NIST, and *FrodoKEM*, a NIST third-round alternate candidate which is also currently recommended by the German Federal Office for Information Security (BSI). We point out subtle differences between the methods used by the above two schemes and the standard IND-CCA enhancing methods in the literature, and argue that these differences invalidate the initial IND-CCA security claims made for the schemes. Following our observations, we re-establish concrete IND-CCA security of Kyber and FrodoKEM in the post-quantum setting by tailoring our analysis to the above differences in a rigorous fashion.

- Second, we consider two specific “beyond IND-CCA” properties – namely, *anonymity* (or *key-privacy*) and *robustness* – which are quite important in modern privacy-enhancing applications. Focusing on the common design paradigm used by most NIST PQC candidates to construct PKE schemes, we provide a modular analysis of the aforementioned properties for PKE schemes built via this paradigm. We then apply our generic analysis to establish post-quantum anonymity and robustness of PKE schemes derived from Kyber and FrodoKEM. Along the way, we also highlight a surprising property of *Classic McEliece*, a NIST PQC fourth-round candidate which is also recommended by BSI, showing that it does not lead to robust PKE schemes.

- Finally, we consider enhancing the decryption functionality of quantum-resistant PKE primitives to a *threshold* (or, *distributed*) setting. This is relevant in view of NIST’s recent plans to also standardize such threshold cryptographic schemes. We first identify issues with the above design paradigm used by NIST PQC candidates for PKE in the context of obtaining IND-CCA secure and efficient threshold schemes. Then we present an alternative paradigm called the “Hybrid” framework which can be used to generically construct PKE schemes that have an efficient distributed decryption functionality, and at the same time are provably IND-CCA secure in a post-quantum setting. We also discuss applicability of our framework to certain NIST PQC schemes: namely, the fourth-round candidate Classic McEliece, and the third-round finalists *NTRU* and *Saber*.

ZUSAMMENFASSUNG

Angetrieben von der wachsenden Bedrohung weitverbreiteter Public-Key Kryptographie durch Quantencomputer haben Standardisierungsagenturen wie das US-basierte National Institute of Standards and Technology (NIST), die Internationale Organisation für Normung (ISO), die Internet Engineering Task Force (IETF), das Europäische Institut für Telekommunikationsnormen (ETSI) mit der Standardisierung von sogenannten Post-Quanten-Primitiven und Algorithmen begonnen.

Insbesondere NIST ist dabei Post-Quanten-Signaturen und Public-Key Verschlüsselung zu standardisieren. Bei letzterer ist die konventionelle IND-CCA Sicherheit von Verschlüsselungsverfahren ein wichtiges Kriterium des NIST. Eine Vielzahl an neu aufkommenden Anwendungen fordern jedoch zum einen Formen von Sicherheit, die IND-CCA Sicherheit übersteigen, und zum anderen erweiterte Funktionalität von kryptographischen Primitiven. Da NIST PQC Standards für ein weites Anwendungsfeld und für Jahrzehnte in der Zukunft anwendbar sein sollen, ist es wichtig diese Primitiven mit Hinblick auf solche Formen von Sicherheit und Funktionalität zu analysieren.

Diese Dissertation liefert die folgenden drei Beiträge, um Post-Quanten-Verschlüsselungsverfahren generisch zu verbessern – speziell im Kontext des NIST Standardisierungsprozesses.

- Zunächst wenden wir uns dem Hauptziel der IND-CCA Sicherheit zu und untersuchen generische Methoden, die in bestimmten wichtigen NIST PQC Kandidaten Anwendung finden. Insbesondere untersuchen wir *Kyber*, der aktuell von NIST ausgewählte Standard, und *FrodoKEM*, eine Alternative, die aktuell vom deutschen Bundesamt für Sicherheit in der Informationstechnik (BSI) empfohlen wird. Wir identifizieren subtile aber signifikante Diskrepanzen zwischen den von den beiden Verfahren genutzten und den in der Fachliteratur untersuchten Techniken, um IND-CCA Sicherheit zu erreichen. Dabei stellen wir fest, dass diese Unterschiede die ursprünglich behauptete Garantie der IND-CCA Sicherheit ungültig machen. Eine rigorose Modellierung der Abweichungen erlaubt uns die konkrete IND-CCA Sicherheit von Kyber und FrodoKEM durch eine massgeschneiderte Analyse wiederherzustellen.

- Zweitens betrachten wir zwei über IND-CCA Sicherheit hinausgehende Eigenschaften, *Anonymität* und *Robustheit*, welche in modernen datenschutzsensitiven Anwendungen eine wichtige Rolle spielen. Wir konzentrieren uns auf ein Designparadigma, das den meisten NIST PQC Kandidaten unterliegt, und geben eine modulare Analyse der genannten Eigenschaften für Verfahren, die auf diesem Paradigma basieren. Durch unsere allgemeine Analyse können wir Post-Quanten-Anonymität und Robustheit für zwei von Kyber und FrodoKEM abgeleitete Verfahren zeigen. Dabei finden wir eine überraschende Eigenschaft des *Classic McEliece* Verfahrens, einem anderen NIST PQC Kandidaten welches ebenfalls vom BSI empfohlen wird, welche die Robustheit verhindert.

- Drittens betrachten wir die Entschlüsselungsfunktionalität von Post-Quanten-Verfahren in einem "verteilten" Szenario. Dies ist relevant, da NIST die Standardisierung solcher verteilten Verfahren ebenfalls plant. Hierbei identifizieren wir Probleme des oben genannten Designparadigmas für die Konstruktion von IND-CCA-sicheren und effizienten verteilten Verfahren. Wir präsentieren einen Ansatz, das sogenannte "Hybrid" Paradigma, das es erlaubt, ein beweisbar IND-CCA-sicheres Verschlüsselungsverfahren mit effizienter verteilter Entschlüsselung generisch zu konstruieren. Zuletzt durchleuten wir die Anwendbarkeit unseres Paradigmas auf die NIST PQC Kandidaten Classic McEliece, *NTRU* und *Saber*.

ACKNOWLEDGEMENTS

This has to be one of the hardest things I have had to write in this thesis. For the technical chapters that follow, I only needed to think back to specific points in time during my doctoral studies when I was working on the corresponding results. However, now I have to look back at my time as a doctoral student (and beyond) as a whole, and thank people who have been a part of this journey – either knowingly or unknowingly. It is not a small feat by any means, but I will try my best . . .

I would like to begin by thanking Kenny, my advisor. By "advisor", I don't only mean his invaluable technical supervision of this thesis but also his guidance in other personal aspects. Kenny's passion for real-world cryptography has successfully rubbed off on me, and will undoubtedly continue to inspire me in the next steps of my career. He is also one of the coolest mentors I have ever had in general. There are so many more things I could thank Kenny for, but most importantly, I would like to thank him for the research group he has established here at ETH Zurich.

Speaking of the Applied Cryptography group, I couldn't have asked for a better set of ~~co-workers~~ friends. There are a lot of things I could thank each member for, but for the sake of brevity, I will mention the first thing that comes to my mind. In an alphabetical order, I would like to thank Alex for formally welcoming me as a doctoral student on behalf of VMI, Anu for sharing her magical baking with the group, Barbara for always taking the time to help me out – even with the most basic things, Ben for introducing me to the legendary "Chicken Katsu Curry" dish, Felix for always stressing the importance of a healthy work-life balance, Fernando for being my reliable pub quiz and bouldering partner, Francesca for her thanksgiving dinners, Igors for helping me out with my first stint as a head TA of the "Applied Cryptography" course, Jan for being the unofficial "lunch coordinator" of the group, Kien for the creativity that he brings to the group, Laura for making me a bit less scared of snakes (but not completely), Lenka for her jumpscares, Lukas for helping me with navigating the doctoral exam process, Matilda for being the best office mate ever (I'll especially miss your "Friday" renditions), Matteo for his cryptographic memes (some of which go over my head), Mia for arguably being my first friend in the group, Patrick for our discussions on Japanese animations and graphic novels (I'll also fondly remember our Norway trip), Shannon for introducing me to new indie artists and expanding my music tastes in general, Sikhar for advising me to do an internship at Visa Research, Simon for being "my man", Siniša for sharing his experiences as a doctoral student, Tianxin for her frequent updates on the latest tech news, and Zichen for being a part of our very first reading group. I would like to thank the visitors we had in our group as well for the relatively brief but nonetheless memorable time we shared; in particular, I thank Hannah and Sam for our shared interests in provable security and pub quizzing respectively. As the thesis coordinator, I got to know a bunch of bright students who added to the lively atmosphere of the group; in this regard, I thank Massimiliano, Giacomo, Miro, Marc (Ilunga), Semira, and Christian. I especially would also like to thank "my Bachelor's students" Oliver, Marc (Himmelberger), and Melanie for our insightful discussions and for making my supervision job much easier.

I would like to thank all the wonderful people that I have had a chance to collaborate with on projects during my doctoral studies; this thesis would not have been possible without their contributions. Specifically, I thank Keita, Paul, Kenny, Kelong, Daniele (Cozzo), Nigel, Navid, Daniel (Masny), Melanie, Sikhar, Srini, Chen-Da, and Ueli. Sincere thanks go to Dennis and Andreas for their time and effort in reviewing this thesis. Special thanks go to Bernadette for her help with the doctoral administrative matters (I'm constantly amazed with the speed she replies to my emails with). I would also like to thank the community of researchers that I got to interact with at various conferences, workshops, symposiums, summer and winter schools, and beyond. For playing an important role in my education in sciences (and arts), I also thank the various professors and lecturers who have inspired me with their courses during my pre-doctoral studies – i.e., secondary and high school, Bachelor's, Master's, etc. On a different note, I'd like to thank Dan Brown for his novel "*Digital Fortress*" which introduced me to the world of cryptography in secondary school; I sometimes wonder whether I would've taken up a career in cryptography if it were not for that novel . . .

During my time as a doctoral student, I was quite fortunate to get to know people from various other groups at ETH Zurich; I would like to thank them for making me feel part of something larger – beyond the norms of a research group. I will start with a group of people that can be aptly called "ETH Crypto people + friends"; this group primarily consists of members from the Information Security and Cryptography, and Foundations of Cryptography research groups. Specifically, I would like to thank Fabio, Giovanni, Gianluca, Chen-Da, Guilherme, Marta, Konstantin, Dennis, Nico, Sebastian, Julia, Roman, Akin, Karen, Michael (Reichle), Bogdan, Suvradip, Cecilia, Kristina, Diana, Andrei, Daniel (Dorda), Goran, Jonathan, Rares, Kasper, Meghana, Simon (Rastikian), Nicolas (Küchler), and Hidde for the fun lunch discussions, after-work beers, hikes, pub quizzes, parties, etc. (special thanks go to Nico for translating the thesis abstract to German). In particular, I will miss the "Lunch?" spam messages on our eponymous Signal group.

I will now move to a different group, namely, the "Super Kondi" ~~group~~ cult which consists of members from the Information Security, System Security, and Programming Methodology research groups. I would like to thank Felix (Linker), Friederike, Linard, Martin, Max, Simon (Erni), Sofia, Srdjan, Thibault, Jonáš, Nicolas (Klose), Daniele (Lain), Graciana, and Edoardo for making me look forward to Tuesday evenings ("Hey! Ho!") and the dinner afterwards. I would also like to thank other members of the Institutes of Information Security, Programming Languages and Systems – namely, Ivan, Mihael, Xenia, Tobias, Reza, Karel, Daniel (Paleka), Daniele (Coppola), Supraja, Benedict, Moritz, Patrick (Schaller), Kari, and Gaurav – for our friendly chats during the ZISC lunch seminars, VMI events, etc.

Another group I want to mention is the SIPLAB whose members have been great neighbors in the CNB workspace. I would like to thank Andreas, Tiffany, Paul, Rayan, Juliete, and Andy for the fun VR experiments, Nintendo Switch parties, friendly conversations in the kitchen area, etc. I would also like to thank the VMI organization for its memorable summer retreats, and organizing other social events such as board game nights, paint & sip events, and pub nights. Such events allowed me to connect with people from different parts of the D-INFK department that I wouldn't have normally met; I thank such people with whom I've had warm interactions (my apologies for not naming them explicitly as there were many).

As a part of my doctoral studies, I had a couple of fruitful internships at Visa Research during the summers of 2021 and 2022. I would like to thank my manager Gaven, Sikhar, Daniel (Masny), Pratyay, Srini, and Navid for their excellent mentorship. I would also like to thank my fellow interns during the 2022 internship for the fun we had in trying out different restaurants in Palo Alto after work, playing in pool tournaments at dive bars across San Francisco, exploring different California beaches during the weekends, etc. Specifically, I thank Andrea, Adithya, Titouan, Ioanna, Gayathri, Mert, Michael (Cuenca), Minje, Jihye, Uday, Shorya, Yibin, Mazharul, and Jingjie.

For *simulating* (cryptographic pun intended) a home away from home in Zurich, I would like to thank Sanchu, Krunal, Manish, Prerit, Jahnavi, Simran, Shaalu, Urvashi, Shweta, Daan, Anmol, Siva, and Samarth; I will cherish our get-togethers, dinner parties, visits to Christmas markets, etc. I would also like to thank Rajan for his pub quizzes in Zurich which I looked forward to (and tried to make my friends look forward to) with excitement.

Back home, I would like to thank my parents and my sister for their unconditional love – and my dog Parker for its (conditional?) love – encouragement and support since forever; I always look forward to our video calls, and even the daily "Good morning!" messages make my day. I dedicate this thesis to my parents in part because of the various hardships they had to go through to give me and my sister the best education possible. Mere words don't do justice to express the gratitude I have towards them. I thank my extended family – cousins, grandparents, uncles, aunts, etc. – as well for always being there for me. I would also like to thank my school and university friends – namely, Diggi, Praful, Utsav, Abhilash ("Part 2"), Raakhi, Suyash, Kalyan, Viraaj, Harsha, Nitish, Karthik, and Nikhil – for helping me stay grounded.

Finally, I am sure I must have missed mentioning a few more people in these acknowledgments. I apologize in advance, but I have a thesis submission deadline to meet. However, I'm also glad that these handful of pages are not enough to thank all the people who had a role in my doctoral journey. In any case, if you are reading this and if we had a friendly interaction at some point, *thank you!*

CONTENTS

1 INTRODUCTION

In recent years, research groups in academia and industry have been devoting a significant amount of effort towards building computers that operate on the basis of quantum mechanical principles (or “quantum computers" for short). A main motivation for this effort is that a quantum computer can solve certain mathematical problems which are assumed to be difficult for traditional computers. But though this may have positive applications in areas such as pharmaceutical and chemical sciences, it also means that if quantum computers could be realized on a large-scale, they would be able to break most public-key cryptosystems deployed currently. This would have serious consequences on the security of digital communications on the Internet and elsewhere.

The above observation has essentially spawned the area of cryptographic research known as *post-quantum cryptography (PQC)*. A goal of PQC is to develop practical cryptographic systems that can resist attacks mounted by large-scale quantum computers. Even though it is not certain when such quantum computers might become a real possibility, standards bodies worldwide such as the US-based National Institute of Standards and Technology (NIST), the International Standards Organization (ISO), the Internet Engineering Task Force (IETF), and the European Telecommunications Standards Institute (ETSI) have nonetheless started working on standardizing cryptographic primitives and algorithms whose security relies on specific mathematical problems being intractable even for quantum computers. Such standardization efforts allow for a widespread deployment of PQC solutions, in preparation for the quantum computing era.

In particular, NIST is currently in the process of standardizing digital signatures and public-key encryption (PKE) schemes in its PQC standardization process. The latter category of primitives will be the main focus of this thesis. An important criterion of evaluation set by NIST for the PKE candidates is whether the schemes achieve *active security* – i.e., the standard notion of *IND-CCA security* – in a post-quantum setting. But in general, it is more difficult to prove that a certain scheme achieves IND-CCA security when compared to notions of *passive security* – i.e., *OW-CPA or IND-CPA security* – especially in a post-quantum adversarial model. Hence, most NIST PQC candidates for PKE employ some generic transformations that enhance passive security of a "base" scheme to active security of the transformed scheme. More specifically, almost all of these transformations can be seen as variants of the well-known class of *Fujisaki-Okamoto (FO) transformations* [1–4].

The FO transformations are quite well-studied in the literature wherein they were shown to offer IND-CCA security (e.g., in the works of [5–8]) – albeit in a heuristic setting called the *quantum random oracle model (QROM)* [9]. Roughly speaking, in the QROM, hash functions used by a cryptographic scheme are idealized as publicly accessible random oracles where an adversary can make queries *in quantum superposition* to such oracles.[1] Despite the need for the above heuristic, FO transforms are quite popular with cryptographic practitioners as they allow the construction of actively secure PKE schemes that are *efficient* in practice. However, upon a close inspection of the variants of FO transforms used by certain important NIST PQC candidates – such as *Kyber* [11], the current "winner" of the standardization process – it turns out that known QROM security results on standard FO transformations in the literature do not directly apply to the aforementioned variants. This is due to subtle but significant differences between the standard FO transforms and the variants used in the NIST PQC process – especially in terms of provable security in the QROM. Hence, given the importance placed by NIST on the above PKE candidates, and the potential widespread deployment of these schemes in the real world, it is imperative to revisit the FO-style variants and formally analyze their security properties in the post-quantum setting.

The chosen standards are also envisioned to be widely used for decades to come. And there is a range of cryptographic applications that have emerged in recent times where IND-CCA security of the underlying PKE primitives may not suffice. Hence, it is also important to perform a broader analysis of the NIST PQC candidates with respect to certain "beyond IND-CCA" security properties that are required by these emerging applications. A class of such applications that will be of focus in this thesis is related to *anonymous* digital communications where, broadly speaking, identities of the communicating parties should be hidden from other parties in the network. The corresponding security properties – namely, *anonymity* (or *key-*

[1] QROM is a generalization of the so-called *random oracle model (ROM)* [10] which was introduced in a pre-quantum setting. In the ROM, an adversary is only given *classical* access to random oracles modelling the underlying hash functions. But as pointed out in [9], in a post-quantum setting, an adversary could evaluate a hash function on a quantum superposition of inputs; and this is not captured in the ROM. Hence, the QROM became the "de facto" security model for assessing the post-quantum security of cryptosystems.

privacy) [12] and *robustness* [13] – have not been examined in detail for general post-quantum PKE schemes. This includes the NIST PQC candidates in particular.

Roughly speaking, a PKE scheme is said to be *anonymous* if a ciphertext does not leak anything about which public key was used to create it, thereby hiding the identity of the intended recipient. Anonymous PKE is a fundamental component of several deployed anonymity systems, such as anonymous credential systems [14]. The property of *robustness* for PKE schemes, on the other hand, was shown to be an essential conjunct of anonymity in [13]; the robustness property guarantees that it is hard to produce a ciphertext which decrypts validly under two different secret keys. But to the best of our knowledge, there have been few works that help us understand how to build anonymous, robust post-quantum PKE schemes, or particularly, whether the NIST schemes provide these properties.

Finally, NIST's PQC standardization process currently only considers the basic PKE primitive which, in general, could be considered to have restricted functionalities. This is in contrast to what some of the existing (pre-quantum) cryptographic algorithms have to offer – e.g., *identity-based encryption* and *threshold public-key encryption*. Focusing on the latter class of algorithms, roughly speaking, threshold schemes allow secret cryptographic operations such as decryption to happen in a distributed fashion involving multiple users, while at the same time, guaranteeing the security of the overall execution even if a certain fraction of users are compromised. NIST has in fact initiated separate plans to standardize threshold schemes for (potentially post-quantum) cryptographic primitives. Coming back to NIST's PQC standardization process, it turns out that the general paradigm used by PKE candidates in their respective constructions – i.e., using an FO-style transform in conjunction with the so-called *KEM-DEM*[2] *paradigm* [15] – does not result in threshold schemes that are both secure *and* efficient. Therefore, in order to prepare NIST's threshold cryptography standardization efforts for the quantum-computing era, there is a need for an alternative paradigm to construct efficient and post-quantum secure threshold PKE schemes.

1.1 THESIS CONTRIBUTIONS

In this thesis, we study ways to *enhance* quantum-resistant PKE schemes in a *generic* manner – especially in the context of NIST's PQC standardization

[2] The primitive KEM (resp., DEM) stands for "Key (resp., Data) Encapsulation Mechanism".

process. There are two important things to note here: namely, our usage of the words “generic” and “enhance”.

By “generic”, we mean that our focus will be on analysing frameworks that construct enhanced PKE schemes which are not tied to any specific hardness assumption – in other words, the frameworks can be instantiated with any such assumption (the above “CPA→CCA” enhancing FO transforms are examples of such frameworks). The main advantage of such generic frameworks is that they enable a *modular* construction of advanced PKE schemes from simpler primitives. This allows scheme designers to focus on instantiating the simpler primitives with appropriate hardness assumptions, which is a much easier task when compared to *directly* constructing advanced schemes from such assumptions.

The word “enhance”, on the other hand, has three distinctive meanings corresponding to the three key contributions of this thesis detailed below; each of our contributions address an issue identified previously with regards to NIST’s PQC standardization process.

I) Analysing IND-CCA Security Enhancements of NIST FO Variants (Chapter 3)

As mentioned above, certain important NIST PQC candidates employ variants of the standard FO transforms for their respective PKE/KEM constructions. And as a consequence of these variations, we argue that the traditional IND-CCA security enhancing properties of the standard FO transformations – established in the literature in a post-quantum setting (i.e., QROM) – do not formally extend to these candidates’ FO-style variants.

More concretely, we revisit the generic FO-variants used in Kyber [11], the currently chosen NIST PQC standard, and *FrodoKEM* [16], a third-round NIST PQC alternate candidate which is currently recommended by the German Federal Office for Information Security (BSI) [17]. We first identify issues with the initial QROM IND-CCA security claims in the above schemes’ NIST specification documents; we do so by zooming in on the subtle differences between their FO-variants and the standard FO transforms. Then we proceed to re-establish concrete IND-CCA security of these NIST PQC schemes in the QROM by proving the corresponding “CPA→CCA” enhancing properties of their respective FO-variants in a rigorous manner. For FrodoKEM, our proof achieves the *same* tightness as the one (incorrectly) claimed in its specification document. For Kyber, we achieve similar tightness except for an additional term in the IND-CCA security bounds that has to do with collision-resistance of an underlying hash (formal details are presented in Section 3.2). We also discuss ways to achieve even tighter security proofs for FrodoKEM and Kyber in view of recent proof techniques introduced in the QROM literature.

It is also worth mentioning that NIST has recently started plans [18, 19] to essentially replace the FO-variant currently used in Kyber with one of the standard FO transforms. As addressed by a representative of the Kyber team [20], this is in part because of our arguments on how the differences between Kyber's variant of the FO transform and the standard FO transforms invalidate the initial QROM IND-CCA security claims made for the scheme in its NIST specification document [11]. In a sense, NIST's decision also showcases the real-world impact this thesis has had on the PQC standardization process.

II) "Beyond IND-CCA" Enhancements: Anonymity and Robustness (Chapters 4, 5)

We then investigate whether the traditional IND-CCA security analysis of NIST PQC schemes can be enhanced to target the aforementioned properties of anonymity and robustness. Namely, we first provide a generic modular theory of anonymity and robustness for PKE schemes built via the KEM-DEM paradigm, since this paradigm is used by most NIST candidates. As a part of our analysis, we introduce formal security definitions of these "beyond IND-CCA" properties for the KEM primitive which, to the best of our knowledge, have not been considered before in the literature. The subsequent results of our analysis vary depending on whether the underlying KEM performs "*explicit rejection*" (i.e., returns a special error symbol "$\perp$" when decapsulating an invalid ciphertext) or "*implicit rejection*" (i.e., the KEM decapsulation never returns "$\perp$" for any ciphertext). On a high level, we show that explicit rejection KEMs transfer their anonymity and robustness properties to PKE schemes obtained via the corresponding KEM-DEM paradigm; whereas implicit rejection KEMs, in general, do not transfer these properties.

This latter result poses a problem because most NIST PQC candidates for PKE use an underlying KEM that is implicitly rejecting. However as noted above, these candidates also use variants of the standard FO transforms for their respective KEM constructions. In this thesis, we analyse one such standard *implicitly-rejecting* FO transformation called $\mathsf{FO}^{\not\perp}$, as introduced in [4] (also see Figure 3.2 for a formal description), with respect to its anonymity and robustness enhancing properties in the QROM. We show that the $\mathsf{FO}^{\not\perp}$ transform confers these properties not only to the constructed KEM, but also to the final KEM-DEM composed PKE scheme. In other words, we show that KEMs built via the $\mathsf{FO}^{\not\perp}$ transform can bypass our above negative result on implicit rejection KEMs being unable to transfer anonymity and robustness properties in the KEM-DEM paradigm generically.

Finally, we look into the applicability of our above generic analysis of $\mathsf{FO}^{\not\perp}$-based KEMs to three specific KEMs related to NIST's PQC standardization process: namely, the current standard Kyber [11], the third-round alternate candidate FrodoKEM [16], and the fourth-round candidate *Classic McEliece (CM)* [21] which (in addition to FrodoKEM) is also recommended by the German federal agency BSI for use. We observe that our generic analysis cannot be extended to CM because of an inherent lack of robustness in the scheme. More concretely, we show with respect to PKE schemes obtained from CM via the standard KEM-DEM paradigm that, for any plaintext m, it is possible to construct a ciphertext c where c always decrypts to m under *any* secret key. In this regard, our work exposes the limitations of CM as a general-purpose KEM for the wide range of applications that can be envisioned for NIST PQC candidates. On the bright side, we successfully adapt our analysis on the anonymity and robustness enhancing properties of $\mathsf{FO}^{\not\perp}$ in the QROM to the specific FO-type variant used by FrodoKEM – as was also the case w.r.t. IND-CCA security – to show that FrodoKEM indeed results in PKE schemes with the corresponding "beyond IND-CCA" security properties. We also show a similar positive result for Kyber. Here we adapt our techniques that were used to prove Kyber's concrete IND-CCA security above to also establish its post-quantum anonymity and robustness.

III) Functionality Enhancements: Efficient Threshold Decryption (Chapter 6)

In the last part of this thesis, we seek to enhance the functionality of quantum-resistant PKE primitives to that of threshold schemes. As highlighted above, most PKE candidates in NIST's PQC standardization process use the standard KEM-DEM paradigm for their constructions. However in a threshold setting, if we want to maintain IND-CCA security of the overall paradigm, we would need to apply a threshold (or, distributed) decryption procedure to the *symmetric* DEM component – and "thresholdizing" symmetric cryptographic primitives is quite expensive, especially for large input messages. One way around this problem is to *leak* the DEM key in the clear after executing distributed decryption of the KEM component, so that the users can perform decryption of the DEM component using the leaked key locally. However, we show that this simple attempt to thresholdize the

KEM-DEM paradigm leads to an insecure scheme, in the IND-CCA sense, because an adversary can exploit the leaked DEM keys (see Section 6.2 for more formal details). So in other words, the requirements of (IND-CCA) security and efficiency seem to be at odds with respect to threshold PKE schemes derived from the standard KEM-DEM paradigm.

To overcome the above issue, we propose an alternative to the KEM-DEM framework called "Hybrid" which can be used to generically construct PKE schemes that have an efficient threshold decryption procedure, while at the same time, being IND-CCA secure in the post-quantum setting. Notably, our analysis is in the QROM. On a high level, our Hybrid transform does leak the DEM key but *after* performing some checks during the distributed decryption; in this regard, our approach can be seen as closely related to the so-called *Tag-KEM* framework [22]. We then formally prove that these checks guarantee the above leakage does not affect IND-CCA security of the overall threshold implementation, specifically in the QROM. We also (briefly) discuss the potential applicability of our Hybrid framework to certain schemes in the NIST PQC standardization process, namely the fourth-round candidate Classic McEliece [21] and the third-round finalist *NTRU* [23]. Given NIST's recent plans to standardize threshold cryptographic schemes [24], we hope our generic framework serves as a stepping stone to more efficient – and post-quantum secure – constructions of threshold PKE schemes.

1.2 PUBLICATIONS

The material in this thesis is based on the following publications:

- (Chapters 3, 4, 5) Paul Grubbs, **Varun Maram**, Kenneth G. Paterson: "Anonymous, Robust Post-Quantum Public-Key Encryption", *In 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2022), pages 402-432* [25].

- (Chapters 3, 5) **Varun Maram**, Keita Xagawa: "Post-Quantum Anonymity of Kyber", *In 26th IACR International Conference on Practice and Theory of Public-Key Cryptography (PKC 2023), pages 3-35* [26].

- (Chapter 6) Kelong Cong, Daniele Cozzo, **Varun Maram**, Nigel P. Smart: "Gladius: LWR Based Efficient Hybrid Public-Key Encryption with Distributed Decryption", *In 27th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2021), pages 125-155* [27].

During my doctoral studies, I also co-authored the following publications that are not covered in this thesis.

- Navid Alamati, **Varun Maram**: "Quantum CCA-Secure PKE, Revisited", *To Appear in 27th IACR International Conference on Practice and Theory of Public-Key Cryptography (PKC 2024)* [28].
- Navid Alamati, **Varun Maram**, Daniel Masny: "Non-Observable Quantum Random Oracle Model", *In 14th International Conference on Post-Quantum Cryptography (PQCrypto 2023), pages 417-444* [29].
- Melanie Jauch, **Varun Maram**: "Quantum Cryptanalysis of OTR and OPP: Attacks on Confidentiality, and Key-Recovery", *To Appear in 30th International Conference on Selected Areas in Cryptography (SAC 2023)* [30].
- **Varun Maram**, Daniel Masny, Sikhar Patranabis, Srinivasan Raghuraman: "On the Quantum Security of OCB" *In IACR Transactions on Symmetric Cryptology, Volume 2022 (ToSC 2022), Issue 2, pages 379-414* [31].
- **Varun Maram**: "On the Security of NTS-KEM in the Quantum Random Oracle Model" *In 8th International Workshop on Code-Based Cryptography (CBCrypto 2020), pages 1-19* [32].
- Chen-Da Liu-Zhang, **Varun Maram**, Ueli Maurer: "On Broadcast in Generalized Network and Adversarial Models" *In 24th International Conference on Principles of Distributed Systems (OPODIS 2020), pages 25:1-25:16* [33].

2

PRELIMINARIES

2.1 NOTATION

For a finite set $\mathcal{S}$, we write "$x \leftarrow_{\$} \mathcal{S}$" to denote that x is sampled uniformly at random from $\mathcal{S}$; in general, for arbitrary sampling distributions in $\mathcal{S}$, we just write "$x \leftarrow \mathcal{S}$" and then describe the corresponding distribution explicitly when not clear from the context. For a logical statement P, we define the boolean value $[P]$ to be 1 if P is satisfied and 0 otherwise.

For probabilistic algorithms A, we use "$y \leftarrow A(x)$" to denote the randomized output y following the output distribution of A on input x; we also sometimes specify the randomness r used in A as "$y := A(x;r)$" to denote the *deterministic* computation of y. We use "A^{O}" to denote that the algorithm A has access to oracle O; we will also make it clear whether A has *classical* or *quantum* access to O when describing our setting.

2.2 QUANTUM RANDOM ORACLE MODEL

As mentioned above, the quantum random oracle model (QROM) – reintroduced by Boneh *et. al.* [9] in a cryptographic context – is an idealized model where a hash function is modeled as a publicly and *quantumly* accessible random oracle in a formal security analysis of the corresponding cryptosystem; the QROM can be seen as a post-quantum generalization of the well-known (classical) ROM [10] wherein the hash oracles only allow *classical* access. Following [9], we model the above quantum random oracles $O\colon \{0,1\}^n \to \{0,1\}^m$ as a unitary mapping $|x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus O(x)\rangle$, where $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$; we refer the reader to [34] for the basics of quantum computation and information.

We now introduce some folklore lemmas in the QROM which will be used extensively in the post-quantum security proofs in this thesis. The following lemma allows one to *perfectly* simulate a quantum random oracle in the view of an adversary.

Lemma 1 (Simulating a QRO [35, Theorem 6.1])**.** *Let $\Omega_{\mathbf{H}}$ be the set of all functions $\mathbf{H} : \mathcal{X} \to \mathcal{Y}$ and $\Omega_{\mathbf{f}}$ be the set of all 2q-wise independent functions $\mathbf{f} : \mathcal{X} \to \mathcal{Y}$. Let $f \leftarrow_{\$} \Omega_{\mathbf{f}}$ and $H \leftarrow_{\$} \Omega_{\mathbf{H}}$. Then the advantage any quantum* *algorithm has in distinguishing the oracles f and H when making q quantum queries is identically zero.*

The second lemma intuitively states that a quantum random oracle can be used as a *quantum-accessible* pseudo-random function (PRF), even if the distinguisher is given full access to the quantum random oracle in addition to the PRF oracle.

Lemma 2 (PRF based on a QRO [5, Lemma 4]). *Let $\Omega_{\mathbf{H}}$ be the set of all functions $\mathbf{H} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ and $\Omega_{\mathbf{R}}$ be the set of all functions $\mathbf{R} : \mathcal{X} \rightarrow \mathcal{Y}$. Let $H \leftarrow\!\!\$\, \Omega_{\mathbf{H}}$, $k \leftarrow\!\!\$\, \mathcal{K}$ and $R \leftarrow\!\!\$\, \Omega_{\mathbf{R}}$. Define the oracles $F_0 = H(k, \cdot)$ and $F_1 = R(\cdot)$. Consider a quantum algorithm A^{H,F_i} that makes at most q quantum queries to H and F_i $(i \in \{0,1\})$. If ("the PRF key") k is chosen independently from A^{H,F_i}'s view, then we have*

$$|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]| \leq \frac{2q}{\sqrt{|\mathcal{K}|}}$$

The lemmas below provide a generic reduction from a hiding-style property (indistinguishability) to a one-wayness-style property (unpredictability) in the QROM. They are also popularly referred to as the *One-Way To Hiding (OW2H) lemmas* in the literature, originally appearing in [36]. We first state the original OW2H lemma of [36] and later state a generalized version of the OW2H lemma from [37]. As will be seen in Chapter 3, different parts of our security analysis of Kyber use different versions of the OW2H lemma for the sake of convenience.

Lemma 3 (Original OW2H [36]). *Let $\Omega_{\mathbf{H}}$ be the set of all functions $\mathbf{H}\colon \mathcal{X} \rightarrow \mathcal{Y}$ and let $H \leftarrow\!\!\$\, \Omega_{\mathbf{H}}$. Consider a quantum algorithm A^H that makes at most q quantum queries to H. Let B^H be a quantum algorithm that on input x does the following: picks $i \leftarrow\!\!\$\, \{1, \ldots, q\}$ and $y \leftarrow\!\!\$\, \mathcal{Y}$, runs $A^H(x, y)$ until (just before) the i-th query, measures the argument of the query in the computational basis and outputs the measurement outcome (if A makes less than i queries, B outputs $\perp \notin \mathcal{X}$). Let*

$$\begin{aligned} P_A^1 &= \Pr[1 \leftarrow A^H(x, H(x)) \mid x \leftarrow\!\!\$\, \mathcal{X}] \\ P_A^2 &= \Pr[1 \leftarrow A^H(x, y) \mid x \leftarrow\!\!\$\, \mathcal{X},\, y \leftarrow\!\!\$\, \mathcal{Y}] \\ P_B &= \Pr[x \leftarrow B^H(x) \mid x \leftarrow\!\!\$\, \mathcal{X}]. \end{aligned}$$

Then, we have $|P_A^1 - P_A^2| \leq 2q\sqrt{P_B}$.

Lemma 4 (Generalized OW2H [37, Theorem 3])**.** *Let $\mathcal{S} \subseteq \mathcal{X}$ be a random subset sampled arbitrarily; similarly, let z be an arbitrarily random bit string. Let $\Omega_{\mathbf{H}}$ be the set of all functions $\mathbf{H}\colon \mathcal{X} \to \mathcal{Y}$, and let the functions $G \leftarrow \Omega_{\mathbf{H}}$ and $H \leftarrow \Omega_{\mathbf{H}}$ such that $G(x) = H(x)$ for all $x \notin \mathcal{S}$; in particular, $\mathcal{S}, G, H, z$ may have arbitrary joint distribution.*

Let A^O be a quantum algorithm making q quantum queries to $O \in \{G, H\}$.[1] *Let B^H be a quantum algorithm that on input z does the following: picks $i \leftarrow\!\!\$ \{1, \dots, q\}$, runs $A^H(z)$ until (just before) the i-th query, measures all query input registers in the computational basis, and outputs the set $\mathcal{T} = \{t_1, \dots, t_{|\mathcal{T}|}\}$ of measurement outcomes. Let*

$$\begin{aligned} P_{\text{left}} &= \Pr[1 \leftarrow A^H(z)] \\ P_{\text{right}} &= \Pr[1 \leftarrow A^G(z)] \\ P_{\text{guess}} &= \Pr[\mathcal{S} \cap \mathcal{T} \neq \emptyset \mid \mathcal{T} \leftarrow B^H(z)]. \end{aligned}$$

Then, $|P_{\text{left}} - P_{\text{right}}| \leq 2q\sqrt{P_{\text{guess}}}$. The same result also holds with B^G instead of B^H in the definition of P_{guess}.

Note that the original OW2H lemma (i.e., Lemma 3) can be seen as a corollary of Lemma 4 wherein we have $H \leftarrow\!\!\$ \Omega_H$, $x \leftarrow\!\!\$ \mathcal{X}$, $y \leftarrow\!\!\$ \mathcal{Y}$, $\mathcal{S} = \{x\}$ such that $G(x) = y$, and $z = (x, H(x))$. In this case, note that we have $P_{\text{left}} = P_A^1$ and $P_{\text{guess}} = P_B$ when compared to Lemma 3. We also have $P_{\text{right}} = \Pr[1 \leftarrow A^G(x, H(x)) \mid x \leftarrow\!\!\$ \mathcal{X}]$ which is the same as $\Pr[1 \leftarrow A^H(x, y) \mid x \leftarrow\!\!\$ \mathcal{X}, y \leftarrow\!\!\$ \mathcal{Y}]$ $(= P_A^2)$ since $H(x)$ and y have the same uniform distribution over $\mathcal{Y}$.

The following lemma gives a lower bound for a decisional variant of the so-called *generic quantum search problem*.

Lemma 5 (Generic Search Problem [38, 39])**.** *Let $\gamma \in [0, 1]$ and $\mathcal{Z}$ be a finite set. Let $\Omega_{\mathbf{N}}$ be the set of all functions $\mathbf{N}\colon \mathcal{Z} \to \{0, 1\}$. Define the function $N_0 \leftarrow \Omega_{\mathbf{N}}$ as follows: for each $z \in \mathcal{Z}$, $N_0(z) = 1$ with probability p_z ($p_z \leq \gamma$), and $N_0(z) = 0$ else. Let $N_1 \leftarrow \Omega_{\mathbf{N}}$ be the function such that $N_1(z) = 0$ $\forall z \in \mathcal{Z}$.*

Let A^O be a quantum algorithm making q quantum queries to $O \in \{N_0, N_1\}$. Then we have

$$|\Pr[1 \leftarrow A^{N_0}] - \Pr[1 \leftarrow A^{N_1}]| \leq 2q\sqrt{\gamma}.$$

[1] Strictly speaking, the generalized OW2H lemma of [37] takes into account the *parallel* oracle queries made by A^O by having q to be the so-called *query depth* of A^O. In this thesis, we will not consider parallel queries of A^O for the sake of simplicity and denote q to be the *query number* of A^O. But our subsequent security proofs can be modified to also consider parallel oracle queries in a straightforward way.

The following lemma describes the collision-resistance of quantum random oracles.

Lemma 6 (Collision-resistance of QROs [40, Theorem 3.1]). *There is a universal constant C (< 648) such that the following holds: Let $\Omega_{\mathbf{H}}$ be the set of all functions $\mathbf{H}\colon \mathcal{X} \to \mathcal{Y}$ and let $H \leftarrow\$ \Omega_{\mathbf{H}}$. For any quantum algorithm A^H making q quantum queries to H, we have*

$$\Pr[H(x_0) = H(x_1) \wedge x_0 \neq x_1 \mid (x_0, x_1) \leftarrow A^H] \leq \frac{C(q+1)^3}{|\mathcal{Y}|}.$$

2.3 CRYPTOGRAPHIC PRIMITIVES

In this section, we define some standard cryptographic primitives which will be used in the thesis. Other primitives and/or their corresponding security properties that are only relevant in certain chapters will be defined in said chapters.

2.3.1 *Public-Key Encryption*

Definition 1 (Public-Key Encryption (PKE) Scheme). *A PKE scheme* PKE, *defined over message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ (and encryption randomness space $\mathcal{R}$), consists of the following triple of efficient algorithms* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*:*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$*: a probabilistic* key-generation *algorithm that outputs a pair of keys* $(\mathsf{pk}, \mathsf{sk})$. pk *and* sk *are called the public/encryption key and private/decryption key, respectively.*
- $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$*: a probabilistic* encryption *algorithm that takes as input encryption key* pk*, message $m \in \mathcal{M}$ and randomness $r \leftarrow\$ \mathcal{R}$, and outputs ciphertext $c \in \mathcal{C}$; we also denote this operation as "$c := \mathsf{Enc}(\mathsf{pk}, m; r)$".*
- $m/\bot := \mathsf{Dec}(\mathsf{sk}, c)$*: a deterministic* decryption *algorithm that takes as input decryption key* sk *and ciphertext c, and outputs message $m \in \mathcal{M}$ or a rejection symbol $\bot \notin \mathcal{M}$.*

Definition 2 (Correctness of PKE [4]). *We say that* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *with message space $\mathcal{M}$, is δ-*correct *if*

$$\mathbb{E}\left[\max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(\mathsf{sk}, c) \neq m \mid c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)]\right] \leq \delta,$$

where the expectation is taken over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$. *Furthermore, if $\delta = 0$, then we just say that* PKE *is* perfectly correct.

```
$\mathsf{OW\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$m^* \leftarrow\!\$\ \mathcal{M}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*)$
$m' \leftarrow \mathcal{A}(\mathsf{pk}, c^*)$
return $[m' = m]$
```

```
$\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\$\ \{0, 1\}$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}^{\textsc{Dec}_{\perp}}(\mathsf{pk})$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}}(c^*, \mathsf{st})$
return $[b' = b]$
```

```
$\textsc{Dec}_a(c)$
if $c = a$ then return $\perp$
$m := \mathsf{Dec}(\mathsf{sk}, c)$
return $m$
```

FIGURE 2.1: Security games for PKE schemes. In the IND-CCA security game, m_0 and m_1 are messages in $\mathcal{M}$ of equal length; also st is some state information maintained by the adversary $\mathcal{A}$.

Definition 3 (γ-Spreadness of PKE). *We say that* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *defined over message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and encryption randomness space $\mathcal{R}$, is* γ-spread *if for every key-pair* $(\mathsf{pk}, \mathsf{sk})$, *message $m \in \mathcal{M}$ and ciphertext $c \in \mathcal{C}$, we have*

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

Definition 4 (Rigidity of PKE [41]). *We say that a* deterministic *PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *defined over ciphertext space $\mathcal{C}$, is* rigid *if for every key-pair* $(\mathsf{pk}, \mathsf{sk})$ *and ciphertext $c \in \mathcal{C}$, we have that either* $\mathsf{Dec}(\mathsf{sk}, c) = \perp$ *or* $\mathsf{Enc}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, c)) = c$.

To define security for all cryptographic primitives – including PKE schemes – in this thesis, we will be using the code-based game-playing framework of Bellare and Rogaway [42]. We will also be using the *concrete security* paradigm wherein we explicitly measure the success probability and resource usage of adversaries in such games; in particular, we will not relate the above quantities of interest to a so-called security parameter, in contrast to the *asymptotic security* paradigm. Finally, we will be working with the so-called "game-hopping" technique, as analyzed by Shoup [43], to formally prove security of cryptographic primitives in the above framework.

We now define some basic security notions for PKE schemes: namely, One-Wayness under Chosen-Plaintext Attacks (OW-CPA), and Indistinguishability under Chosen-Plaintext Attacks (IND-CPA) and under Chosen-Ciphertext Attacks (IND-CCA).

Definition 5 (OW-CPA, IND-CPA/-CCA Security of PKE). *Given a PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space $\mathcal{M}$, we define the game* *w.r.t. its* OW-CPA security *in Figure 2.1 and the* OW-CPA advantage measure *for adversary $\mathcal{A}$ against* PKE *as*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) = \Pr[\mathsf{OW\text{-}CPA}_{\mathsf{PKE}}^{\mathcal{A}} = 1].$$

(In general, "$\mathsf{G}_{\mathsf{P}}^{\mathcal{A}} = b$" denotes the security game G, *"played" by an adversary $\mathcal{A}$ against the cryptographic primitive* P, *outputting the bit $b \in \{0,1\}$.)*

Similarly, we define the game w.r.t. its IND-CCA *security in Figure 2.1 and the* IND-CCA advantage measure *for adversary $\mathcal{A}$ against* PKE *as*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) = \left| \Pr[\mathsf{IND\text{-}CCA}_{\mathsf{PKE}}^{\mathcal{A}} = 1] - \frac{1}{2} \right|.$$

If we remove the adversaries' access to decryption oracles in the IND-CCA security game, we obtain the corresponding game for IND-CPA security; *the* IND-CPA advantage measure *is defined in the same fashion as that of IND-CCA.*

It is also well-known that IND-CPA security of a PKE scheme with a sufficiently large message space implies its OW-CPA security [4, 44]. More formally:

Lemma 7 ([4, Lemma 2.3])**.** *Let* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with message space $\mathcal{M}$. For any OW-CPA adversary $\mathcal{A}$ against* PKE, *there exists an IND-CPA adversary $\mathcal{B}$ against* PKE *with the same running time as that of $\mathcal{A}$ such that*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) \leq \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}) + \frac{1}{|\mathcal{M}|}.$$

For PKE schemes – and other primitives in this thesis – whose security analysis in the post-quantum setting relies on modeling their component hash functions as quantum random oracles, the corresponding security games are extended in a straightforward manner in the QROM by additionally giving the adversaries *quantum* access to the hash oracles; however, the adversaries still have *classical* access to the remaining oracles, such as the decryption oracle in the IND-CCA security game, since they represent operations performed on users' devices that are assumed to be classical (i.e., non-quantum) in the scenario of post-quantum cryptography.[2]

Following [4, 5], we also make the convention that the number q_O of queries made by an adversary $\mathcal{A}$ to an oracle O counts the total number of

[2] This is in contrast to the so-called "*quantum cryptography*" scenario where the users' devices can be quantum, and hence for example, the adversaries can have quantum access to the decryption oracle [45]. However, we will not consider this setting in the thesis.

times O is executed in the corresponding security game – i.e., the number of $\mathcal{A}$'s explicit queries to O plus the number of implicit queries to O made during the execution of the game.

Now a standard way to construct efficient PKE schemes in practice is to compose two other primitives – namely, *key-encapsulation mechanism (KEM)* and *data-encapsulation mechanism (DEM)* – in a way known as the *KEM-DEM paradigm*. We first formally define the aforementioned primitives and then describe the paradigm in more detail.

2.3.2 *Key Encapsulation Mechanism*

Definition 6 (Key Encapsulation Mechanism (KEM)). *A KEM scheme* KEM, *defined over encapsulated key space* $\mathcal{K}$ *and ciphertext space* $\mathcal{C}$, *consists of the following triple of efficient algorithms* $(\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$*:*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$*: a probabilistic* key-generation *algorithm that outputs a pair of keys* $(\mathsf{pk}, \mathsf{sk})$. pk *and* sk *are called the public/encapsulation key and private/decapsulation key, respectively.*
- $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$*: a probabilistic* encapsulation *algorithm that takes as input encapsulation key* pk*, and outputs ciphertext* $c \in \mathcal{C}$ *and its corresponding encapsulated key* $k \in \mathcal{K}$.
- $k/\bot := \mathsf{Decap}(\mathsf{sk}, c)$*: a deterministic* decapsulation *algorithm that takes as input decapsulation key* sk *and ciphertext* c*, and outputs key* $k \in \mathcal{K}$ *or a rejection symbol* $\bot \notin \mathcal{K}$.

Definition 7 (Correctness of KEM). *We say that* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *is* δ-correct *if*

$$\Pr[\mathsf{Decap}(\mathsf{sk}, c) \neq k \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}, (c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})] \leq \delta.$$

In particular, we say that KEM *is* perfectly correct *if* $\delta = 0$.

Definition 8 (IND-CCA Security of KEM). *Given a KEM scheme* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *with* $\mathcal{K}$ *as its encapsulated key space, we define the game w.r.t. its* IND-CCA security *in Figure 2.2 and the* IND-CCA advantage measure *for adversary* $\mathcal{A}$ *against* KEM *as*

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

```
$\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$\ \{0, 1\}$
$(c^*, k_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$
$k_1^* \leftarrow\!\!\$\ \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k_b^*)$
return $[b' = b]$
```

```
$\textsc{Decaps}_a(c)$
if $c = a$ then return $\bot$
$k := \mathsf{Decap}(\mathsf{sk}, c)$
return $k$
```

FIGURE 2.2: IND-CCA security game for KEMs.

2.3.3 *Data Encapsulation Mechanism*

Definition 9 (Data Encapsulation Mechanism (DEM)). *A DEM scheme* DEM*, defined over key space $\mathcal{K}$, message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, consists of the following triple of efficient algorithms* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*:*

- $k \leftarrow \mathsf{KGen}$*: a probabilistic* key-generation *algorithm that outputs a single key $k \in \mathcal{K}$. This key k is used for both encryption and decryption, and hence is often referred to as "*symmetric key*" in the literature; this is in contrast to the above "*asymmetric key*" primitives of PKE and KEM.*
- $c \leftarrow \mathsf{Enc}(k, m)$*: a probabilistic* encryption *algorithm that takes as input key k and message $m \in \mathcal{M}$, and outputs ciphertext $c \in \mathcal{C}$.*
- $m/\bot := \mathsf{Dec}(k, c)$*: a deterministic* decryption *algorithm that takes as input key k and ciphertext c, and outputs message $m \in \mathcal{M}$ or a rejection symbol $\bot \notin \mathcal{M}$.*

Definition 10 (Correctness of DEM). *We say that* $\mathsf{DEM} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*, with message space $\mathcal{M}$, is* perfectly correct *if for any message $m \in \mathcal{M}$, we have*

$$\Pr[\mathsf{Dec}(k, c) = m \mid k \leftarrow \mathsf{KGen}, c \leftarrow \mathsf{Enc}(k, m)] = 1.$$

Definition 11 (One-time IND-CCA Security of DEM). *Given a DEM scheme* $\mathsf{DEM} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space $\mathcal{M}$, we define the game w.r.t. its* one-

```
$\mathsf{otIND\text{-}CCA}^{\mathcal{A}}_{\mathsf{DEM}}$
    $k \leftarrow \mathsf{KGen}$
    $b \leftarrow\$ \{0,1\}$
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp}$
    $c^* \leftarrow \mathsf{Enc}(k, m_b)$
    $b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}}(c^*, \mathsf{st})$
    return $[b' = b]$

$\textsc{Dec}_a(c)$
    if $c = a$ then return $\perp$
    $m := \mathsf{Dec}(k, c)$
    return $m$
```

Figure 2.3: One-time IND-CCA security game for DEMs. Here m_0 and m_1 are messages in $\mathcal{M}$ of equal length; also st is some state information maintained by the adversary $\mathcal{A}$.

time[3] IND-CCA (otIND-CCA) security *in Figure 2.3 and the* otIND-CCA advantage measure *for adversary $\mathcal{A}$ against* DEM *as*

$$\mathbf{Adv}^{\mathsf{otIND\text{-}CCA}}_{\mathsf{DEM}}(\mathcal{A}) = \left| \Pr[\mathsf{otIND\text{-}CCA}^{\mathcal{A}}_{\mathsf{DEM}} = 1] - \frac{1}{2} \right| .$$

If we remove the adversaries' access to decryption oracles in the otIND-CCA security game, we obtain the corresponding game for one-time IND-CPA security*; the* otIND-CPA advantage measure *is defined in the same fashion as that of otIND-CCA.*

THE KEM-DEM PARADIGM. As mentioned above, composing an asymmetric KEM and a symmetric DEM is a standard way to construct PKE; the resulting schemes are often called *"hybrid"* PKE. Given a KEM scheme $\mathsf{KEM} = (\mathsf{KGen}^{\mathsf{kem}}, \mathsf{Encap}, \mathsf{Decap})$ and a DEM scheme $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}, \mathsf{Dec})$, the hybrid PKE scheme $\mathsf{PKE}^{\mathsf{hy}} = (\mathsf{KGen}^{\mathsf{hy}}, \mathsf{Enc}^{\mathsf{hy}}, \mathsf{Dec}^{\mathsf{hy}})$ resulting from their composition according to the KEM-DEM paradigm is described in Figure 2.4. Moreover, it is well-known that if KEM is IND-CCA secure and DEM is otIND-CCA secure, then the resulting $\mathsf{PKE}^{\mathsf{hy}}$ is IND-CCA secure [15]. More formally, we have the following:

Lemma 8. *Let* $\mathsf{KEM} = (\mathsf{KGen}^{\mathsf{kem}}, \mathsf{Encap}, \mathsf{Decap})$ *be a KEM scheme and* $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}, \mathsf{Dec})$ *be a DEM scheme, and* $\mathsf{PKE}^{\mathsf{hy}} = (\mathsf{KGen}^{\mathsf{hy}}, \mathsf{Enc}^{\mathsf{hy}}, \mathsf{Dec}^{\mathsf{hy}})$

[3] The term "one-time" comes from the fact that in the security game (see Fig. 2.3), the adversary $\mathcal{A}$ is allowed to ask for the encryption of a pair of messages (m_0, m_1) *only once*. Note that $\mathcal{A}$ cannot locally compute encryptions of arbitrary messages since the encryption key k is secret; this is in contrast to the IND-CCA security game for PKE (see Fig. 2.1) where the encryption key pk is known.

```
$\mathsf{KGen}^{\mathsf{hy}}$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}^{\mathsf{kem}}$
    return $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Enc}^{\mathsf{hy}}(\mathsf{pk}, m)$
    $(c_0, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$
    $c_1 \leftarrow \mathsf{Enc}(k, m)$
    $c := (c_0, c_1)$
    return $c$

$\mathsf{Dec}^{\mathsf{hy}}(\mathsf{sk}, c)$
    Parse $c = (c_0, c_1)$
    $k := \mathsf{Decap}(\mathsf{sk}, c_0)$
    if $k = \bot$, return $\bot$
    $m := \mathsf{Dec}(k, c_1)$
    return $m$
```

FIGURE 2.4: Hybrid PKE scheme $\mathsf{PKE}^{\mathsf{hy}} = (\mathsf{KGen}^{\mathsf{hy}}, \mathsf{Enc}^{\mathsf{hy}}, \mathsf{Dec}^{\mathsf{hy}})$ built via the composition of $\mathsf{KEM} = (\mathsf{KGen}^{\mathsf{kem}}, \mathsf{Encap}, \mathsf{Decap})$ and $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}, \mathsf{Dec})$ using the KEM-DEM paradigm.

be the hybrid PKE scheme resulting from the composition of KEM *and* DEM *using the KEM-DEM paradigm (see Fig. 2.4). Then for any* IND-CCA *adversary* $\mathcal{A}_{\mathrm{hy}}$ *against* $\mathsf{PKE}^{\mathsf{hy}}$*, there exist adversaries* $\mathcal{A}_{\mathrm{kem}}$ *and* $\mathcal{A}_{\mathrm{dem}}$ *targeting the* IND-CCA *security of* KEM *and* otIND-CCA *security of* DEM *respectively such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq 2 \cdot \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}_{\mathrm{kem}}) + \mathbf{Adv}^{\mathsf{otIND\text{-}CCA}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}).$$

Moreover, $\mathcal{A}_{\mathrm{kem}}$ *and* $\mathcal{A}_{\mathrm{dem}}$ *run in the same time as* $\mathcal{A}_{\mathrm{hy}}$*. If* $\mathcal{A}_{\mathrm{hy}}$ *makes* q *decryption oracle queries, then* $\mathcal{A}_{\mathrm{kem}}$ *makes at-most* q *decapsulation queries and* $\mathcal{A}_{\mathrm{dem}}$ *makes at-most* q *decryption queries.*

AUTHENTICATED ENCRYPTION. In the KEM-DEM paradigm above, note that using a one-time IND-CCA secure DEM is sufficient. However in practice, we use DEMs satisfying a stronger notion of security called (one-time) *authenticated encryption (AE)*. To understand this notion, we first need to define the notion of *ciphertext integrity (INT-CTXT security)*.

Definition 12 (Ciphertext integrity of DEM). *Given a DEM scheme* $\mathsf{DEM} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*, we define the game w.r.t. its* INT-CTXT security *in Figure 2.5 and the* INT-CTXT advantage measure *for adversary* $\mathcal{A}$ *against* DEM *as*

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}) = \Pr[\mathsf{INT\text{-}CTXT}^{\mathcal{A}}_{\mathsf{DEM}} = 1].$$

If adversaries are restricted to making at-most a single query to the encryption oracle in the above security game (i.e., $|\mathcal{L}| \leq 1$*), we obtain the corresponding game for* one-time INT-CTXT security*; the otINT-CTXT advantage measure is defined in the same fashion as that of INT-CTXT security.*

```
$\mathsf{INT\text{-}CTXT}^{\mathcal{A}}_{\mathsf{DEM}}$
$k \leftarrow \mathsf{KGen}$
$\mathcal{L} := \phi$
$\mathsf{win} := 0$
$\mathcal{A}^{\textsc{Enc},\textsc{Dec}}$
return $\mathsf{win}$

$\textsc{Enc}(m)$
$c \leftarrow \mathsf{Enc}(k, m)$
$\mathcal{L} := \mathcal{L} \cup \{c\}$
return $c$

$\textsc{Dec}(c)$
$m := \mathsf{Dec}(k, c)$
if $m \neq \perp \wedge c \notin \mathcal{L}$ then
    $\mathsf{win} := 1$
return $\mathsf{win}$
```

FIGURE 2.5: Security game for INT-CTXT security of DEMs.

Now a DEM is said to offer (one-time) AE security if it is (one-time) IND-CPA secure[4] and provides (one-time) INT-CTXT security. It is also well-known that (one-time) AE security of a DEM implies its (one-time) IND-CCA security (hence we often use one-time AE secure DEMs in the KEM-DEM composition). Restricting our focus to one-time security in context of the KEM-DEM paradigm, we have the following formal lemma:

Lemma 9 ([46, Theorem 9.1], adapted). *Let* $\mathsf{DEM} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a DEM scheme. For any one-time IND-CCA adversary* $\mathcal{A}$ *against* DEM*, there exists a one-time IND-CPA adversary* $\mathcal{B}$ *and a one-time INT-CTXT adversary* $\mathcal{B}'$ *against* DEM*, both with the same running time as that of* $\mathcal{A}$*, such that*

$$\mathbf{Adv}^{\mathsf{otIND\text{-}CCA}}_{\mathsf{DEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{otIND\text{-}CPA}}_{\mathsf{DEM}}(\mathcal{B}) + 2 \cdot \mathbf{Adv}^{\mathsf{otINT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{B}').$$

2.3.4 *Message Authentication Code*

Definition 13 (Message Authentication Code (MAC)). *A MAC scheme* MAC*, defined over key space* $\mathcal{K}$*, message space* $\mathcal{M}$ *and tag space* $\mathcal{T}$*, consists of the following triple of efficient algorithms* $(\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$*:*

- $k \leftarrow \mathsf{KGen}$*: a probabilistic* key-generation *algorithm that outputs a single key* $k \in \mathcal{K}$*.*

[4] In the notion of (plain) IND-CPA security, adversaries can ask for encryptions of *multiple* pairs of messages (m_0, m_1) of equal length, in contrast to *one-time* IND-CPA security, before outputting the bit b' – while still having no access to the decryption oracle (cf. Definition 11).

```
$\mathsf{SUF\text{-}CMA}^{\mathcal{A}}_{\mathsf{MAC}}$
$k \leftarrow \mathsf{KGen}$
$\mathcal{L} := \phi$
$\mathsf{win} := 0$
$\mathcal{A}^{\textsc{Tag},\textsc{Vf}}$
return $\mathsf{win}$

$\textsc{Tag}(m)$
$t \leftarrow \mathsf{Tag}(k, m)$
$\mathcal{L} := \mathcal{L} \cup \{(m, t)\}$
return $t$

$\textsc{Vf}(m, t)$
if $\mathsf{Vf}(k, m, t) = 1 \wedge (m, t) \notin \mathcal{L}$ then
    $\mathsf{win} := 1$
return $\mathsf{win}$
```

Figure 2.6: SUF-CMA security game for MACs.

- $t \leftarrow \mathsf{Tag}(k, m)$*: a (potentially) probabilistic* tag *algorithm that takes as input key k and message* $m \in \mathcal{M}$*, and outputs a tag* $t \in \mathcal{T}$.
- $b := \mathsf{Vf}(k, m, t)$*: a deterministic* verification *algorithm that takes as input key k, message m, and tag t, and outputs a bit* $b \in \{0, 1\}$ *where* $b = 0$ *and* $b = 1$ *are synonymous with "reject" and "accept" respectively.*

Definition 14 (Correctness of MAC)**.** *We say that* $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$*, with message space* $\mathcal{M}$*, is* perfectly correct *if for any message* $m \in \mathcal{M}$*, we have*

$$\Pr[\mathsf{Vf}(k, m, t) = 1 \mid k \leftarrow \mathsf{KGen}, t \leftarrow \mathsf{Tag}(k, m)] = 1.$$

We now define a standard notion of security for MACs: namely, Strong Unforgeability under Chosen-Message Attacks (SUF-CMA security).

Definition 15 (SUF-CMA Security of MAC)**.** *Given a MAC scheme* $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$*, we define the game w.r.t. its* SUF-CMA security *in Figure 2.6 and the* SUF-CMA advantage measure *for adversary* $\mathcal{A}$ *against* MAC *as*

$$\mathbf{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{A}) = \Pr[\mathsf{SUF\text{-}CMA}^{\mathcal{A}}_{\mathsf{MAC}} = 1].$$

ENCRYPT-THEN-MAC. Here we describe a standard way to construct AE schemes by combining an IND-CPA secure[5] symmetric encryption scheme[6]

[5] See Footnote 4 of this chapter.

[6] Symmetric encryption schemes are syntactically equivalent to DEMs. However, it is a convention in the literature to refer to AE-secure (or, IND-CCA secure) symmetric key encryption, at-least in the context of KEM-DEM paradigm, as "DEMs". Hence, we will be referring to the IND-CPA secure building blocks used to construct AE-secure DEMs via the EtM transform as just "symmetric encryption schemes".

```
$\mathsf{KGen}^{\mathsf{dem}}$
    $k_0 \leftarrow \mathsf{KGen}^{\mathsf{se}}$
    $k_1 \leftarrow \mathsf{KGen}^{\mathsf{mac}}$
    return $(k_0, k_1)$
```

```
$\mathsf{Enc}^{\mathsf{dem}}((k_0, k_1), m)$
    $c \leftarrow \mathsf{Enc}(k_0, m)$
    $t \leftarrow \mathsf{Tag}(k_1, c)$
    return $(c, t)$
```

```
$\mathsf{Dec}^{\mathsf{dem}}((k_0, k_1), (c, t))$
    if $\mathsf{Vf}(k_1, c, t) = 0$, return $\perp$
    $m := \mathsf{Dec}(k_0, c)$
    return $m$
```

FIGURE 2.7: DEM $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}^{\mathsf{dem}}, \mathsf{Dec}^{\mathsf{dem}})$ built via the composition of $\mathsf{SE} = (\mathsf{KGen}^{\mathsf{se}}, \mathsf{Enc}, \mathsf{Dec})$ and $\mathsf{MAC} = (\mathsf{KGen}^{\mathsf{mac}}, \mathsf{Tag}, \mathsf{Vf})$ using the EtM transform.

and a SUF-CMA secure MAC via so-called *Encrypt-then-Mac (EtM)* transform. Given a symmetric encryption scheme $\mathsf{SE} = (\mathsf{KGen}^{\mathsf{se}}, \mathsf{Enc}, \mathsf{Dec})$ and a MAC $\mathsf{MAC} = (\mathsf{KGen}^{\mathsf{mac}}, \mathsf{Tag}, \mathsf{Vf})$, the DEM $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}^{\mathsf{dem}}, \mathsf{Dec}^{\mathsf{dem}})$ obtained by their composition according to the EtM transform is described in Figure 2.7. The following lemma shows the AE-security of such DEMs.

Lemma 10 ([46, Theorem 9.2])**.** *Let* $\mathsf{SE} = (\mathsf{KGen}^{\mathsf{se}}, \mathsf{Enc}, \mathsf{Dec})$ *be a symmetric encryption scheme and* $\mathsf{MAC} = (\mathsf{KGen}^{\mathsf{mac}}, \mathsf{Tag}, \mathsf{Vf})$ *be a MAC scheme, and* $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}^{\mathsf{dem}}, \mathsf{Dec}^{\mathsf{dem}})$ *be the DEM resulting from the composition of* SE *and* MAC *using the EtM transform (see Fig. 2.7). Then:*

1. *For any INT-CTXT adversary* $\mathcal{A}_{\mathrm{dem}}$ *against* DEM*, there exists an SUF-CMA adversary* $\mathcal{A}_{\mathrm{mac}}$ *with the same running time as that of* $\mathcal{A}_{\mathrm{dem}}$ *such that*

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) \leq \mathbf{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{A}_{\mathrm{mac}}).$$

2. *For any IND-CPA adversary* $\mathcal{A}_{\mathrm{dem}}$ *against* DEM*, there exists an IND-CPA adversary* $\mathcal{A}_{\mathrm{se}}$ *with the same running time as that of* $\mathcal{A}_{\mathrm{dem}}$ *such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{SE}}(\mathcal{A}_{\mathrm{se}}).$$

3

IND-CCA SECURITY ENHANCEMENTS, REVISITED

One of the most well-known, and well-used, enhancements for practical public-key encryption schemes in the post-quantum setting – at-least in the context of NIST's post-quantum cryptography (PQC) standardization process – is the so-called *"Fujisaki-Okamoto (FO) transformation"* [1, 2]. Roughly speaking, the FO transformation generically amplifies the passive security (i.e., OW-/IND-CPA security) of PKE schemes to active security (i.e., IND-CCA security) – albeit in the heuristical random oracle model (ROM) [10]. Subsequently, some "modern" variants of the FO transformation were proposed in the literature [3, 4] which constructed IND-CCA secure KEMs, instead of PKE schemes, since the former primitive is more versatile in terms of applications – e.g., building authenticated key exchange, PKE schemes via the "KEM-DEM paradigm" [15] (see Section 2.3 above), etc.

In this chapter, we will consider certain modern FO variants in [4]: namely, $\mathsf{FO}_m^{\not\perp}, \mathsf{FO}_m^{\perp}, \mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\perp}$, as described in Figures 3.1 and 3.2. The main difference between $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}_m^{\perp}$ – similarly, between $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\perp}$ – is that in the former, the decapsulation algorithm never outputs $\perp$ when rejecting invalid ciphertexts, as opposed to the latter (see "$\mathsf{Decap}(\mathsf{sk}', c)$" in Figs. 3.1 and 3.2). In this context, the $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}^{\not\perp}$ transforms are said to be *implicitly-rejecting*, and $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}^{\perp}$ are said to be *explicitly-rejecting*.

It was formally proven in [3, 4] that the above four standard FO transforms offer IND-CCA security in the (classical) ROM. Moving to the post-quantum setting, it was initially shown in the works of [5, 6] that the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}^{\not\perp}$ transforms do achieve IND-CCA security in the QROM. However, these – and related – works could not establish post-quantum IND-CCA security for the explictly-rejecting $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}^{\perp}$ *without* modifying the underlying transform; e.g., [4, 47] provide QROM IND-CCA security proofs for modified versions of $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}^{\perp}$ which include an extra "key confirmation" hash in the ciphertext. (A detailed discussion on state-of-the-art provable IND-CCA security results for the explicitly-rejecting and implicitly-rejecting FO transforms in the QROM is provided in Subsection 3.2.2 below.)

Shifting our focus to NIST's PQC standardization process, most final-round KEM candidates employed *variants* of the standard implicitly-rejecting

```
$\mathsf{KGen}'$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$s \leftarrow \perp$
$\boxed{s \leftarrow\!\!\$\ \mathcal{M}}$
$\mathsf{sk}' \leftarrow (\mathsf{sk}, s)$
$\textbf{return } (\mathsf{pk}, \mathsf{sk}')$

$\mathsf{Encap}(\mathsf{pk})$
$m \leftarrow\!\!\$\ \mathcal{M}$
$r \leftarrow G_r(m)$
$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
$\bar{k} \leftarrow G_k(m)$
$\textbf{return } (c, \bar{k})$

$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$r' \leftarrow G_r(m')$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
$\textbf{if } c' = c \textbf{ then}$
    $\textbf{return } G_k(m')$
$\boxed{\textbf{else return } G_k(s, c)}$
$\textbf{else return } \perp$
```

Figure 3.1: The KEMs $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, G_r, G_k]$ and $\boxed{\mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, G_r, G_k]}$. For notational simplicity, we set $s \leftarrow \perp$ for $\mathsf{FO}_m^{\perp}$. Here $\mathcal{M}$ is the message space of $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and G_r, G_k are hash functions with appropriate domain and co-domain.

```
$\mathsf{KGen}'$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$s \leftarrow \perp$
$\boxed{s \leftarrow\!\!\$\ \mathcal{M}}$
$\mathsf{sk}' \leftarrow (\mathsf{sk}, s)$
$\textbf{return } (\mathsf{pk}, \mathsf{sk}')$

$\mathsf{Encap}(\mathsf{pk})$
$m \leftarrow\!\!\$\ \mathcal{M}$
$r \leftarrow G_r(m)$
$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
$\bar{k} \leftarrow G_k(m, c)$
$\textbf{return } (c, \bar{k})$

$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$r' \leftarrow G_r(m')$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
$\textbf{if } c' = c \textbf{ then}$
    $\textbf{return } G_k(m', c)$
$\boxed{\textbf{else return } G_k(s, c)}$
$\textbf{else return } \perp$
```

Figure 3.2: The KEMs $\mathsf{FO}^{\perp}[\mathsf{PKE}, G_r, G_k]$ and $\boxed{\mathsf{FO}^{\not\perp}[\mathsf{PKE}, G_r, G_k]}$. For notational simplicity, we set $s \leftarrow \perp$ for $\mathsf{FO}^{\perp}$. The descriptions of PKE, $\mathcal{M}$, and hashes G_r, G_k are the same as that in Figure 3.1.

$\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}^{\not\perp}$ transforms, given their provable IND-CCA security guarantees in the QROM; such candidates also cite these results in the literature *as-they-are* to claim post-quantum IND-CCA security of their respective KEMs. However as will be argued in the following sections, some of these candidates use FO-variants which differ significantly from $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}^{\not\perp}$, thereby invalidating a direct application of QROM proof techniques used to analyze the standard transforms to establish IND-CCA security of the NIST PQC schemes.

More specifically, we will revisit the FO-variants used in two important NIST PQC candidates – namely *Kyber* [11], which is the current "winner" of the NIST PQC standardization process [48], and *FrodoKEM* [16], which was a final-round NIST PQC alternate candidate and is currently recommended by the German Federal Office for Information Security (BSI) [17] – and their corresponding IND-CCA security claims in the QROM. After identifying issues with their initial security claims, we will proceed to (re-)establish concrete IND-CCA security of these NIST PQC schemes in the QROM via tailor-made proofs that account for variations between their FO-variants and the standard (implicitly-rejecting) FO transforms. Our post-quantum security analyses of FrodoKEM and Kyber are presented in Sections 3.1 and 3.2 respectively.

3.1 IND-CCA SECURITY OF FRODOKEM IN THE QROM

In this section, we analyze the concrete IND-CCA security of FrodoKEM in the QROM. First we describe the scheme, specifically the variant of FO transform used by it, in more detail in Subsection 3.1.1. In Subsection 3.1.2, we discuss problems in the initial QROM IND-CCA security claims for FrodoKEM in its NIST PQC specification document [16], and then provide a high-level overview of our new security proof which (re-)establishes IND-CCA security of FrodoKEM in the QROM with the *same* concrete bounds as claimed in the specification document [16, Theorem 5.8]; formal details of our QROM proof follow in Subsection 3.1.3. Finally in Subsection 3.1.4, we discuss some relevant proof techniques in the QROM literature which can be used to potentially obtain even *tighter* security bounds for FrodoKEM.

3.1.1 *Specification of FrodoKEM*

FrodoKEM is a lattice-based KEM which relies on hardness of the well-known *learning-with-errors (LWE)* problem [49] for its post-quantum security

```
$\mathsf{KGen}'$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
    $s \leftarrow\!\!\$ \{0,1\}^{256}$
    $\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$
    $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$
    return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Encap}(\mathsf{pk})$
    $m \leftarrow\!\!\$ \{0,1\}^{256}$
    $h \leftarrow H(\mathsf{pk})$
    $(\overline{k}, r) \leftarrow G(m, h)$
    $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
    $k \leftarrow H'(\overline{k}, c)$
    return $(c, k)$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
    Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
    $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
    $(\overline{k}', r') \leftarrow G(m', h)$
    $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
    if $c' = c$ then
        return $H'(\overline{k}', c)$
    else return $H'(s, c)$
```

FIGURE 3.3: The $\mathsf{FO}^{\text{frodo}}$ transform used in FrodoKEM. Here we have FrodoPKE $=$ $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and FrodoKEM $=$ $(\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$; also we have hash functions H, H' with 256-bit outputs and function G with 512-bit outputs.

claims. The KEM is constructed by applying an FO-type transform, which we refer to as $\mathsf{FO}^{\text{frodo}}$, on a base PKE scheme called "FrodoPKE" (see [16] for a detailed specification); the $\mathsf{FO}^{\text{frodo}}$ transform is described in detail in Figure 3.3.

Note that in our description of $\mathsf{FO}^{\text{frodo}}$, we are technically using parameters of FrodoKEM targeting "Level 5" security as specified by NIST. However, our subsequent IND-CCA security analysis of FrodoKEM in the QROM can be extended in a straightforward fashion to account for other parameter sets as well.

3.1.2 *Technical Overview*

In FrodoKEM's NIST PQC specification document, specifically in [16, Section 5.1.2], it was claimed that prior QROM IND-CCA security results established for the standard $\mathsf{FO}^{\not\perp}$ transform (see Fig. 3.2) in the literature – particularly, the results of Jiang *et al.* [5] – also apply to the $\mathsf{FO}^{\text{frodo}}$ transform used by the scheme in a similar fashion; in fact, $\mathsf{FO}^{\text{frodo}}$ is also referred to as "$\mathsf{FO}^{\not\perp'}$" in [16]. However, we argue that the specific proof techniques used by Jiang *et al.* [5], for example, to obtain concrete IND-CCA security bounds for $\mathsf{FO}^{\not\perp}$ in the QROM do not directly apply to FrodoKEM's variant of the FO transform – this is because of some significant differences between $\mathsf{FO}^{\text{frodo}}$ and the standard $\mathsf{FO}^{\not\perp}$ transforms.

Namely, an important trick used in [5] for the security proofs of $\mathsf{FO}^{\not\perp}$ is to replace the computation of the key "$k \leftarrow H'(m, c)$"[1] with "$k \leftarrow H''(g(m))(= H''(c))$" for function $g(\cdot) = \mathsf{Enc}(\mathsf{pk}, \cdot; G_r(\cdot))$ and a secret random function $H''(\cdot)$; note that in this case, we simply have $\mathsf{Decap}(\mathsf{sk}, c) = H''(c)$ leading to an "efficient" simulation of the decapsulation oracle without using the secret key sk. To justify this replacement, the authors of [5] then argue about the injectivity of $g(\cdot)$, relying on the correctness of the underlying PKE scheme to establish this.

But in FrodoKEM, keys are computed as "$k \leftarrow H'(\bar{k}, c)$" (see "$\mathsf{Encap}(\mathsf{pk})$" in Fig. 3.3) where the "pre-key" $\bar{k}$ is derived as a hash of the message m (to be specific, $(\bar{k}, r) \leftarrow G(m, H(\mathsf{pk}))$). So there is an extra *layer* of hashing between m and the computation of k. Hence, to use a similar trick as [5], we would require some additional injectivity arguments. Thus, strictly speaking, the proof techniques in [5] do not directly apply to FrodoKEM.

Nevertheless, we are able to overcome the above barrier by adapting the analysis of $\mathsf{FO}^{\not\perp}$ in [5] to obtain an explicit IND-CCA security proof for FrodoKEM in the QROM, with the *same* tightness as claimed in the specification document [16, Theorem 5.8]. The formal proof can be found in Subsection 3.1.3 that follows. But before that, we give a high-level overview of our approach below.

Note that we can replace the step "$(\bar{k}, r) \leftarrow G(m, H(\mathsf{pk}))$" in FrodoKEM's encapsulation (Lines 2 and 3 in "$\mathsf{Encap}(\mathsf{pk})$", Fig. 3.3) by "$\bar{k} \leftarrow G_k(m)$" and "$r \leftarrow G_r(m)$" for two fresh random oracles G_k, $G_r : \{0,1\}^{256} \rightarrow \{0,1\}^{256}$ (similar to the description of $\mathsf{FO}^{\not\perp}$ transform in Fig. 3.2). Now our key observation is that the extra layer of hashing "$G_k(\cdot)$" between m and $\bar{k}$ is actually *length-preserving*, i.e., the hash function has the same domain and range. So following [4, 50], we can replace the random oracle $G_k(\cdot)$ with a random *polynomial* of degree $2q_G - 1$ over a finite field representation of $\{0,1\}^{256}$ (i.e., a $2q_G$-wise independent function); here q_G is the number of queries made to oracle G in the IND-CCA security reduction for FrodoKEM. Because of Lemma 1 in Section 2.2, this change is perfectly indistinguishable to an adversary making at most q_G queries to G_k. This will allow us to recover m from a corresponding pre-key value $\bar{k}$ by computing roots of the polynomial $G_k(x) - \bar{k}$. Hence we can *invert* this "nested" hashing of m in order to apply the trick of Jiang *et al.* [5] above. Namely, we can now replace the key derivation "$k \leftarrow H'(\bar{k}, c)$" with "$k \leftarrow H''(g(m))(= H''(c))$"

[1] Note that in Line 4 of "$\mathsf{Encap}(\mathsf{pk})$", Figure 3.2, we have keys to be derived as "$\bar{k} \leftarrow G_k(m, c)$". However for a better comparison of the $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\mathsf{frodo}}$ transforms, we are renaming "$\bar{k}$" to "k" and "G_k" to "H'" in the former transform so as to make the notation consistent with that of the latter transform.

for function $g(\cdot) = \mathsf{Enc}(\mathsf{pk}, \cdot\,; G_r(\cdot))$, where in addition, m is a root of the polynomial $G_k(x) - \bar{k}$.

3.1.3 *Security Analysis*

We now formally prove IND-CCA security of the scheme FrodoKEM = $\mathsf{FO}^{\text{frodo}}[\text{FrodoPKE}, G, H, H']$ (see Figure 3.3) in the QROM with the following concrete bounds:

Theorem 1. *Given the base scheme* FrodoPKE = (KGen, Enc, Dec) *is δ-correct, for any* IND-CCA *adversary $\mathcal{A}$ against* FrodoKEM = (KGen′, Encap, Decap) *issuing at most q_G and $q_{H'}$ queries to the quantum random oracles G and H' respectively, there exists an* IND-CPA *adversary $\mathcal{B}$ against* FrodoPKE *such that*

$$\mathbf{Adv}^{\text{IND-CCA}}_{\text{FrodoKEM}}(\mathcal{A}) \leq 2(q_G + q_{H'})\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\text{FrodoPKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{2q_{H'}}{2^{128}} + 4q_G\sqrt{\delta}$$

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

The proof that follows is structurally similar to that of [5, Theorem 1]. But the key component of this proof that overcomes the barrier described in Subsection 3.1.2 above is encapsulated in the "$\mathsf{G}_5 \to \mathsf{G}_8$" game-hops.

Proof. Denote $\Omega_{\mathbf{G}_2}$, $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$ and $\Omega_{\mathbf{H}'}$ to be the set of all functions $\mathbf{G}_2 : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{G} : \{0,1\}^{256} \to \{0,1\}^{256}$, $\mathbf{H} : \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ and $\mathbf{H}' : \mathcal{C} \to \{0,1\}^{256}$ respectively, where $\mathcal{C}$ is the ciphertext space of FrodoPKE/FrodoKEM.

Let $\mathcal{A}$ be an adversary in the IND-CCA game for FrodoKEM issuing at most q_G and $q_{H'}$ quantum queries to the random oracles G and H' respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_{11}$ described in Fig. 3.4.

Game G_0: The game G_0 is exactly the IND-CCA game for FrodoKEM. Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{IND-CCA}}_{\text{FrodoKEM}}(\mathcal{A}).$$

Game G_1: In game G_1, we modify the decapsulation oracle $\textsc{Decaps}_{c^*}$ such that $H^{\text{rej}}(c)$ is returned instead of $H'(s, c)$ for an invalid ciphertext c, where the random oracle H^{rej} is not directly accessible to $\mathcal{A}$. Here we can use Lemma 2 w.r.t. the pseudorandomness of $H'(s, \cdot)$, with "PRF key" $s \leftarrow\!\!\$\ \{0,1\}^{256}$, to obtain the following via a straightforward reduction:

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{2q_{H'}}{\sqrt{2^{256}}}.$$

```
Games $\mathsf{G}_0 - \mathsf{G}_{11}$
$(\mathsf{pk}, \mathsf{sk}') \leftarrow \mathsf{KGen}'$
$G_2 \xleftarrow{\$} \Omega_{\mathbf{G}_2}; G_r \xleftarrow{\$} \Omega_{\mathbf{G}}$
$G_r^{\mathrm{good}} \leftarrow \Omega_{\mathbf{G}}$   // Sampling distribution
    // described in description of $\mathsf{G}_4$ below.
$G_r := G_r^{\mathrm{good}}$   // $\mathsf{G}_4 - \mathsf{G}_8$
$G_k \xleftarrow{\$} \Omega_{\mathbf{G}}$   // $\mathsf{G}_0 - \mathsf{G}_4$
$G_k \xleftarrow{\$} \Omega_{\mathbf{poly}}$   // $\mathsf{G}_5 - \mathsf{G}_{11}$
$H_2 \xleftarrow{\$} \Omega_{\mathbf{H}}; H^{\mathrm{rej}} \xleftarrow{\$} \Omega_{\mathbf{H}'}$
$H_3 \xleftarrow{\$} \Omega_{\mathbf{G}}; H^{\mathrm{acc}} \xleftarrow{\$} \Omega_{\mathbf{H}'}$
$b \xleftarrow{\$} \{0,1\}$
$m^* \xleftarrow{\$} \{0,1\}^{256}$
$(\overline{k}^*, r^*) \leftarrow G(m^*, H(\mathsf{pk}))$   // $\mathsf{G}_0 - \mathsf{G}_2$
$r^* \leftarrow G_r(m^*)$   // $\mathsf{G}_3 - \mathsf{G}_9$
$r^* \xleftarrow{\$} \{0,1\}^{256}$   // $\mathsf{G}_{10} - \mathsf{G}_{11}$
$\overline{k}^* \leftarrow G_k(m^*)$   // $\mathsf{G}_3 - \mathsf{G}_7$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$
$k_0^* \leftarrow H'(\overline{k}^*, c^*)$   // $\mathsf{G}_0 - \mathsf{G}_7$
$k_0^* \leftarrow H_3(m^*)$   // $\mathsf{G}_8 - \mathsf{G}_9$
$k_0^* \xleftarrow{\$} \{0,1\}^{256}$   // $\mathsf{G}_{10} - \mathsf{G}_{11}$
$k_1^* \xleftarrow{\$} \{0,1\}^{256}$
$\mathsf{inp} \leftarrow (\mathsf{pk}, (c^*, k_b^*))$
$i \xleftarrow{\$} \{1, \ldots, q_G + q_{H'}\}$   // $\mathsf{G}_{11}$
run $\mathcal{A}^{G, H', \textsc{Decaps}_{c^*}}(\mathsf{inp})$ until
    $i$-th query to $G_r \times H_3$   // $\mathsf{G}_{11}$
measure the $i$-th query and let the
    outcome be $m'$   // $\mathsf{G}_{11}$
return $[m' = m^*]$   // $\mathsf{G}_{11}$
$b' \leftarrow \mathcal{A}^{G, H', \textsc{Decaps}_{c^*}}(\mathsf{inp})$
return $[b' = b]$
```

```
$G(m, h)$
$(\overline{k}, r) \leftarrow G_2(m, h)$
if $h = H(\mathsf{pk})$ then   // $\mathsf{G}_2 - \mathsf{G}_{11}$
    $r \leftarrow G_r(m)$   // $\mathsf{G}_2 - \mathsf{G}_{11}$
    $\overline{k} \leftarrow G_k(m)$   // $\mathsf{G}_2 - \mathsf{G}_{11}$
return $(\overline{k}, r)$
```

```
$H'(\overline{k}, c)$
return $H_2(\overline{k}, c)$   // $\mathsf{G}_0 - \mathsf{G}_5$
Compute set of roots $S$
    of polynomial $G_k(x) - \overline{k}$
if $\exists m' \in S$ s.t.
    $\mathsf{Enc}(\mathsf{pk}, m'; G_r(m')) = c$
    if $c = c^*$ then   // $\mathsf{G}_8 - \mathsf{G}_{11}$
        return $H_3(m')$   // $\mathsf{G}_8 - \mathsf{G}_{11}$
    return $H^{\mathrm{acc}}(c)$
return $H_2(\overline{k}, c)$
```

```
$\textsc{Decaps}_a(c)$
if $c = a$ then return $\bot$
return $H^{\mathrm{acc}}(c)$   // $\mathsf{G}_7 - \mathsf{G}_{11}$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' = \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$   // $\mathsf{G}_0 - \mathsf{G}_2$
$r' \leftarrow G_r(m')$   // $\mathsf{G}_3 - \mathsf{G}_6$
$\overline{k}' \leftarrow G_k(m')$   // $\mathsf{G}_3 - \mathsf{G}_6$
if $\mathsf{Enc}(\mathsf{pk}, m'; r') = c$ then
    return $H'(\overline{k}', c)$
else return $H'(s, c)$   // $\mathsf{G}_0$
else return $H^{\mathrm{rej}}(c)$   // $\mathsf{G}_1 - \mathsf{G}_6$
```

FIGURE 3.4: Games G_0 – G_{11} for the proof of Theorem 1.

Game G_2**:** In game G_2, we implicitly divide the G-queries into two categories: (1) query is of the form (m, h) with $h = H(\mathsf{pk})$ and (2) the remaining queries. We then respond to the queries from the respective categories with $(G_k(m), G_r(m))$ and $G_2(m, h)$ respectively, where G_k, G_r are internal random oracles. It is not hard to verify that the output distributions of the G-oracle in games G_1 and G_2 are equivalent. Therefore,

$$\Pr[\mathsf{G}_2 = 1] = \Pr[\mathsf{G}_1 = 1].$$

Game G_3**:** In game G_3, we make the following changes w.r.t. the G-oracle evaluation. First, we generate the values $\overline{k}^*, r^*$ in setup of the game as "$\overline{k}^* \leftarrow G_k(m^*)$" and "$r^* \leftarrow G_r(m^*)$" (effectively replacing the step "$(\overline{k}^*, r^*) \leftarrow G(m^*, H(\mathsf{pk}))$" in G_2). We then similarly generate the values $\overline{k}', r'$ w.r.t. the decapsulation oracle $\textsc{Decaps}_{c^*}$ as "$\overline{k}' \leftarrow G_k(m')$" and "$r' \leftarrow G_r(m')$" (replacing the step "$(\overline{k}', r') \leftarrow G(m', h)$" in G_2, where $h = H(\mathsf{pk})$ since we assume honest generation of $(\mathsf{pk}, \mathsf{sk}')$ in the setup).

Since these changes are "cosmetic" in nature following our modification to oracle G in game G_2, we have

$$\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_2 = 1].$$

Game G_4**:** In game G_4, we change the random oracle G_r such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}, \mathsf{sk})$. To be specific, given a FrodoPKE key-pair $(\mathsf{pk}, \mathsf{sk})$ and a message $m \in \{0,1\}^{256}$, define

$$\mathcal{R}_{\text{good}}((\mathsf{pk}, \mathsf{sk}), m) = \{r \in \{0,1\}^{256} \mid \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m; r)) = m\}.^{2}$$

Now define the oracle $G_r^{\text{good}} \leftarrow \Omega_{\mathbf{G}}$ such that $G_r^{\text{good}}(m)$ is sampled according to a uniform distribution in $\mathcal{R}_{\text{good}}((\mathsf{pk}, \mathsf{sk}), m)$. In G_4, we then replace the random oracle G_r with G_r^{good}. Note that the task of distinguishing between G_3 and G_4 is the same as that of distinguishing between the oracles G_r and G_r^{good}. And the latter distinguishing probability can be bounded by using Lemma 5 in a similar fashion as in the analysis of the "$\mathsf{G}_1 \rightarrow \mathsf{G}_2$" game-hop in the proof of [5, Theorem 1], while relying on the δ-correctness of FrodoPKE; by following the reduction in [5], it is not hard to obtain

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq 2q_G\sqrt{\delta}.$$

[2] Note that $\{0,1\}^{256}$ is both the message space and encryption randomness space of FrodoPKE when targeting NIST PQC "Level 5" security; see Subsection 3.1.1.

Game G_5**:** In game G_5, we replace the random oracle G_k with a $2q_G$-wise independent function, following Lemma 1. Random polynomials of degree $2q_G - 1$ over the finite field representation of the message space $\{0,1\}^{256}$ are $2q_G$-wise independent. Let $\Omega_{\mathbf{poly}}$ be the set of all such polynomials. We are then replacing the step "$G_k \leftarrow\!\!\$\ \Omega_{\mathbf{G}}$" with "$G_k \leftarrow\!\!\$\ \Omega_{\mathbf{poly}}$" in G_5. From Lemma 1, as this change is indistinguishable when the oracle G_k is queried at most q_G times, we have

$$\Pr[\mathsf{G}_5 = 1] = \Pr[\mathsf{G}_4 = 1].$$

Game G_6**:** In game G_6, we implicitly divide the H'-queries into two disjoint categories: (1) query is of the form $(\bar{k}, c)$ such that there exists $m \in \{0,1\}^{256}$ which is a root of the polynomial $G_k(x) - \bar{k}$ (recall that G_k is now a polynomial) *and* $\mathsf{Enc}(\mathsf{pk}, m; G_r(m)) = c$, and (2) the remaining queries. We then respond to queries from the respective categories with $H^{\text{acc}}(c)$ and $H_2(\bar{k}, c)$, where H^{acc} is an internal random oracle not directly accessible to the adversary $\mathcal{A}$.

Focusing on H'-queries in "category (1)", note that it is not possible for two distinct queries $(\bar{k}', c)$ and $(\bar{k}'', c)$ to result in the same output $H^{\text{acc}}(c)$. The reason is, as G_r now samples "good" random coins, there can exist at most one value m that satisfies $\mathsf{Enc}(\mathsf{pk}, m; G_r(m)) = c$. And since $G_k(\cdot)$ is a deterministic function, the above follows. Therefore, the output distributions of the H'-oracle in the games G_5 and G_6 are equivalent, and we get

$$\Pr[\mathsf{G}_6 = 1] = \Pr[\mathsf{G}_5 = 1].$$

Game G_7**:** In game G_7, we change the $\textsc{Decaps}_{c^*}$ oracle such that there is no need for the secret key sk'. Namely, $H^{\text{acc}}(c)$ is returned for the decapsulation of any ciphertext c w.r.t. sk'. Let $m' = \mathsf{Dec}(\mathsf{sk}, c)$, $r' = G_r(m')$ and $\bar{k}' = G_k(m')$. Now consider the following two cases:

1. $\underline{\mathsf{Enc}(\mathsf{pk}, m'; r') = c}$: In this case, the $\textsc{Decaps}_{c^*}$ oracle returns $H'(\bar{k}', c)$ in game G_6 and $H^{\text{acc}}(c)$ in game G_7. It is not hard to see that we have $H'(\bar{k}', c) = H^{\text{acc}}(c)$ in G_6, since the query $(\bar{k}', c)$ falls under "category (1)" w.r.t. oracle H'. Therefore, $\textsc{Decaps}_{c^*}$ oracles of games G_6 and G_7 return the same value $H^{\text{acc}}(c)$.

2. $\underline{\mathsf{Enc}(\mathsf{pk}, m'; r') \neq c}$: In this case, the $\textsc{Decaps}_{c^*}$ oracle returns $H^{\text{rej}}(c)$ in game G_6 and $H^{\text{acc}}(c)$ in game G_7. In game G_6, as the random function H^{rej} is independent of all other oracles, the output $H^{\text{rej}}(c)$ is uniformly random in the adversary $\mathcal{A}$'s view. In game G_7, the only way

> $\mathcal{A}$ gets prior access to the value $H^{\text{acc}}(c)$ is if it made a H'-query $(\overline{k}'', c)$ such that $\mathsf{Enc}(\mathsf{pk}, m''; G_r(m'')) = c$ (and $G_k(m'') = \overline{k}''$). But since G_r samples "good" random coins, we have $\mathsf{Dec}(\mathsf{sk}, c) = m'' = m'$ leading to a contradiction of "$\mathsf{Enc}(\mathsf{pk}, m'; r') \neq c$". Therefore, such a prior access is not possible and $H^{\text{acc}}(c)$ will also be a uniformly random value in $\mathcal{A}$'s view.

As the output distributions of the $\textsc{Decaps}_{c^*}$ oracle in G_6 and G_7 are the same in both cases, we have

$$\Pr[\mathsf{G}_7 = 1] = \Pr[\mathsf{G}_6 = 1].$$

Game G_8**:** In game G_8, we make a further modification to the evaluation of "category (1)" H'-queries of the form $(\overline{k}, c^*)$ as follows, where c^* is the challenge ciphertext computed in the setup: respond to the corresponding "category (1)" query with $H_3(m)$, where m is a (lexicographically minimal) root of polynomial $G_k(x) - \overline{k}$ that satisfies $\mathsf{Enc}(\mathsf{pk}, m; G_r(m)) = c^*$. Here H_3 is another internal independent random oracle.

Since we established in the "$\mathsf{G}_5 \to \mathsf{G}_6$" game-hop that there cannot be two distinct "category (1)" H'-queries $(\overline{k}^*, c^*)$ and $(\overline{k}', c^*)$, this further change to the H'-oracle only affects the H'-query $(\overline{k}^*, c^*)$, where $\overline{k}^* = G_k(m^*)$ for the secret message m^* sampled uniformly at random in the setup (and $\mathsf{Enc}(\mathsf{pk}, m^*; G_r(m^*)) = c^*$). W.r.t. this query, the H' oracle would return $H^{\text{acc}}(c^*)$ in G_7, and $H_3(m^*)$ in G_8. The adversary $\mathcal{A}$'s view would be identical even after this change because the random value $H^{\text{acc}}(c^*)$ is only accessible to $\mathcal{A}$ via the H'-oracle in G_7, and in particular, not through the $\textsc{Decaps}_{c^*}$ oracle since c^* is a forbidden decapsulation query. Hence in G_8, we are effectively replacing a uniformly random value that can only be accessed via the H'-oracle by $\mathcal{A}$ with another uniformly random value. Hence, the output distributions of the H'-oracle in the games G_7 and G_8 are equivalent. Therefore, we have

$$\Pr[\mathsf{G}_8 = 1] = \Pr[\mathsf{G}_7 = 1].$$

Following the above modification, we make a "cosmetic" change in the setup where the "real" key k_0^* defined in the setup is now generated as "$k_0^* \leftarrow H_3(m^*)$" (instead of "$k_0^* \leftarrow H'(\overline{k}^*, c^*)$"). This change does not affect the game in any way.

Game G_9**:** In game G_9, we reset the random oracle G_r so that it returns uniformly random coins from $\{0,1\}^{256}$ instead of returning only "good"

```
$A^{G_r \times H_3}(m^*, (r^*, k_0^*))$
$(\mathsf{pk}, \mathsf{sk}') \leftarrow \mathsf{KGen}'$
$G_2 \leftarrow\!\$\ \Omega_{\mathbf{G_2}}; G_k \leftarrow\!\$\ \Omega_{\mathbf{poly}}$
$H_2 \leftarrow\!\$\ \Omega_{\mathbf{H}}; H^{\mathrm{acc}} \leftarrow\!\$\ \Omega_{\mathbf{H'}}$
$b \leftarrow\!\$\ \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$
$k_1^* \leftarrow\!\$\ \{0,1\}^{256}$
$\mathsf{inp} \leftarrow (\mathsf{pk}, (c^*, k_b^*))$
$b' \leftarrow \mathcal{A}^{G,H',\textsc{Decaps}_{c^*}}(\mathsf{inp})$
return $[b' = b]$

$G(m, h)$
if $h = H(\mathsf{pk})$ then
    $r \leftarrow G_r(m)$
    $\bar{k} \leftarrow G_k(m)$
else $(\bar{k}, r) \leftarrow G_2(m, h)$
return $(\bar{k}, r)$
```

```
$H'(\bar{k}, c)$
Compute set of roots $S$
 of polynomial $G_k(x) - \bar{k}$
if $\exists m' \in S$ s.t.
 $\mathsf{Enc}(\mathsf{pk}, m'; G_r(m')) = c$
  if $c = c^*$ then
    return $H_3(m')$
  return $H^{\mathrm{acc}}(c)$
return $H_2(\bar{k}, c)$

$\textsc{Decaps}_a$
if $c = a$ then return $\bot$
return $H^{\mathrm{acc}}(c)$
```

FIGURE 3.5: Algorithm $A^{G_r \times H_3}$ for the proof of Theorem 1.

random coins. Since this change, in a sense, is the "inverse" of the game-hop "$\mathsf{G}_3 \to \mathsf{G}_4$", by using a similar analysis, we obtain

$$|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_8 = 1]| \leq 2q_G\sqrt{\delta}.$$

Game G_{10}: In the set-up of game G_{10}, we generate the values r^* and k_0^* such that they are uniformly random values independent of any oracles, i.e., we replace the step "$r^* \leftarrow G_r(m^*)$" with "$r^* \leftarrow\!\$\ \{0,1\}^{256}$" and "$k_0^* \leftarrow H_3(m^*)$" with "$k_0^* \leftarrow\!\$\ \{0,1\}^{256}$". Note that in this game, both the "real" and "random" keys are sampled uniformly at random from $\{0,1\}^{256}$ (i.e., both keys have the exact same distribution). Hence, the challenge bit b is independent from $\mathcal{A}$'s view and we get

$$\Pr[\mathsf{G}_{10} = 1] = \frac{1}{2}.$$

Now we use the original OW2H lemma (Lemma 3) to bound the difference in the success probabilities of $\mathcal{A}$ in G_9 and G_{10}. Let A be an oracle algorithm that has quantum access to the random oracle $G_r \times H_3$, where $G_r, H_3 \leftarrow\$ \Omega_{\mathbf{G}}$ and $(G_r \times H_3)(m) = (G_r(m), H_3(m))$. Figure 3.5 describes $A^{G_r \times H_3}$'s operation on input $(m^*, (r^*, k_0^*))$. Note that the algorithm $A^{G_r \times H_3}$ makes at most $q_G + q_{H'}$ number of queries to the random oracle $G_r \times H_3$ to respond to $\mathcal{A}$'s G-oracle and H'-oracle queries.[3]. With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_9 = 1]$ and $P_A^2 = \Pr[\mathsf{G}_{10} = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 3 w.r.t. the algorithm $A^{G_r \times H_3}$; to analyze the corresponding probability P_B in Lemma 3, we define game G_{11} as shown in Figure 3.4 such that $P_B = \Pr[\mathsf{G}_{11} = 1]$. From Lemma 3, we now have

$$|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_{10} = 1]| \leq 2(q_G + q_{H'})\sqrt{\Pr[\mathsf{G}_{11} = 1]}$$

Finally, we bound the success probability of $\mathcal{A}$ in G_{11} by a reduction to the OW-/IND-CPA security of the base FrodoPKE scheme. Specifically, we construct an OW-CPA adversary $\mathcal{B}'$ against FrodoPKE such that on input a public-key pk along with a ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ for uniformly random (secret) message $m^* (\leftarrow\$ \{0,1\}^{256})$ and randomness $r^* (\leftarrow\$ \{0,1\}^{256})$ chosen by the OW-CPA challenger, $\mathcal{B}'$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_{11} (e.g., starting with sampling a uniformly random bit $b \leftarrow\$ \{0,1\}$).

- Uses two different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2 and G_k respectively, two different $2q_{H'}$-wise independent functions to simulate the random oracles H^{acc} and H_2 respectively, and two different $2(q_G + q_{H'})$-wise independent functions to perfectly simulate the random oracles G_r and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 1. Also evaluates $\mathcal{A}$'s G- and H'-queries using the oracle $G_r \times H_3$; the random oracles G and H' are simulated in the same way as in G_{11}.

- Answers decapsulation queries using the oracle H^{acc} as in G_{11}.

- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^* \leftarrow\$ \{0,1\}^{256}$ and responds with $(\mathsf{pk}, (c^*, k^*))$.

- Selects $i \leftarrow\$ \{1, \ldots, q_G + q_{H'}\}$, measures the i-th query to oracle $G_r \times H_3$ and returns the outcome m'.

[3] For example, if $A^{G_r \times H_3}$ wants to respond to $\mathcal{A}$'s H'-query, then $A^{G_r \times H_3}$ prepares a uniform superposition of all states in the output register corresponding to G_r (see [50] for particulars of this “trick”).

Again, it is not hard to see that $\Pr[\mathsf{G}_{11} = 1] \leq \mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathrm{FrodoPKE}}(\mathcal{B}')$. From Lemma 7, since we know that IND-CPA security of a PKE scheme with a sufficiently large message space also implies its OW-CPA security, corresponding to adversary $\mathcal{B}'$, there exists an IND-CPA adversary $\mathcal{B}$ against FrodoPKE such that

$$\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathrm{FrodoPKE}}(\mathcal{B}') \leq \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathrm{FrodoPKE}}(\mathcal{B}) + \frac{1}{2^{256}}$$

where the running time of $\mathcal{B}$ is that of $\mathcal{B}'$, and $\{0,1\}^{256}$ is the message space of FrodoPKE.

Hence by collecting all of the above bounds, we finally arrive at

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathrm{FrodoKEM}}(\mathcal{A}) \leq 2(q_G + q_{H'})\sqrt{\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathrm{FrodoPKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{2q_{H'}}{2^{128}} + 4q_G\sqrt{\delta}.$$

□

Furthermore, IND-CPA security of the base FrodoPKE scheme was rigorously established in [16, Subsection 5.1.4] while relying on hardness of the LWE problem; the δ-correctness property of FrodoPKE has also been concretely analyzed in [16, Subsection 2.2.7].

3.1.4 *Related Work*

Note that in our QROM IND-CCA security proof for FrodoKEM in Subsection 3.1.3 above, we relied on the original OW2H lemma introduced by Unruh [36]. Subsequent to his work, variants of the OW2H lemma have been proposed in the literature – notably, in the works of [51, 52] – which allow for tighter IND-CCA security bounds w.r.t. the standard (implicitly-rejecting) FO transforms in the QROM.

However, the aforementioned tight security proofs make an additional assumption on the base PKE scheme being *injective* (as defined in [51]; we will discuss this assumption in a bit more detail in Subsection 3.2.2 below when we consider Kyber). But to the best of our knowledge, a formal analysis of the injectivity of FrodoPKE is lacking in the literature – in contrast to Kyber [53]. Hence, we leave such an analysis of FrodoPKE as an open question, in the context of obtaining tighter IND-CCA security proofs for FrodoKEM in the QROM.

3.2 IND-CCA SECURITY OF KYBER IN THE QROM

After analysing FrodoKEM, we now focus on Kyber in this section wherein we formally prove IND-CCA security of the NIST PQC standard in the QROM with concrete bounds. We first describe the scheme in more detail in Subsection 3.2.1. In Subsection 3.2.2, we highlight some issues with the QROM IND-CCA security claims made for Kyber in its NIST PQC specification document [11], and then provide a high-level overview of our new approach to establish its (tight) IND-CCA security. In Subsection 3.2.3, we present a detailed analysis of Kyber in the QROM which contains all formal details of our IND-CCA security proof. We conclude the section by comparing our analysis to some alternative analyses of Kyber in prior and subsequent work in Subsection 3.2.4.

3.2.1 *Specification of Kyber*

As described in [11], Kyber is a lattice-based KEM whose claimed IND-CCA security relies on hardness of the so-called *module learning-with-errors (MLWE)* problem [54]. Kyber – or more formally, $\mathsf{Kyber.KEM}$ – is constructed by first starting with a base PKE scheme $\mathsf{Kyber.PKE}$ and then applying a tweaked Fujisaki-Okamoto (FO) transform to it in order to obtain the final KEM. The tweaked FO transform, which we call $\mathsf{FO}^{\mathsf{kyber}}$, is described in detail in Figure 3.6; we also refer the reader to [11, Section 1.2] for a detailed specification of $\mathsf{Kyber.PKE}$.

One thing to note is that in our description of $\mathsf{FO}^{\mathsf{kyber}}$, we have the key-deriving hash function H' to only return outputs of bit-length 256. However, Kyber technically instantiates H' with the extendable-output function SHAKE-256 which can return outputs of arbitrary length. But rest assured, our subsequent IND-CCA security analysis of Kyber can be modified in a straightforward manner to account for encapsulated keys (derived from H') with arbitrary length.

3.2.2 *Technical Overview*

As can be seen in Figure 3.6, Kyber implements a transform that deviates *even further* from the $\mathsf{FO}^{\not\perp}$ transform than FrodoKEM does. Specifically, the keys in Kyber are computed as "$k \leftarrow H'(\overline{k}, H(c))$" where the "pre-key" $\overline{k}$ is

```
$\mathsf{KGen}'$
1 :  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
2 :  $s \leftarrow\$ \{0,1\}^{256}$
3 :  $\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$
4 :  $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$
5 :  return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Encap}(\mathsf{pk})$
1 :  $m \leftarrow\$ \{0,1\}^{256}$
2 :  $m \leftarrow H(m)$
3 :  $h \leftarrow H(\mathsf{pk})$
4 :  $(\bar{k}, r) \leftarrow G(m, h)$
5 :  $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
6 :  $k \leftarrow H'(\bar{k}, H(c))$
7 :  return $(c, k)$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
1 :  Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
2 :  $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :  $(\bar{k}', r') \leftarrow G(m', h)$
4 :  $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 :  if $c' = c$ then
6 :      return $H'(\bar{k}', H(c))$
7 :  else return $H'(s, H(c))$
```

FIGURE 3.6: The transform $\mathsf{FO}^{\mathsf{kyber}}$ used in Kyber. Here we have $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and $\mathsf{Kyber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$. Also we have hash functions H, H' with 256-bit outputs and function G with 512-bit outputs.

derived as a hash of the message[4] m (to be specific, $(\bar{k}, r) \leftarrow G(m, H(\mathsf{pk}))$). Again there is an extra hashing step between m and the computation of k, as we have seen for FrodoKEM. But at the same time, there is also a "nested" hashing of ciphertext in the key-derivation (i.e., Kyber uses "$H(c)$" instead of just "c") as opposed to the standard "single" hashing in $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\mathsf{frodo}}$ (see Fig. 3.3).

We argue that this "extra" hash of the ciphertext acts as a barrier when trying to apply the generic QROM proof techniques used in the literature, for the standard (implicitly-rejecting) FO transforms, towards establishing the post-quantum IND-CCA security of Kyber with the *same* bounds as that for the standard transforms – contradicting what was claimed in its NIST PQC specification document [11]. Specifically, it was claimed in [11, Section 4.3.2] that the techniques used by Saito *et al.* [6] to establish IND-CCA security for the $\mathsf{FO}_m^{\not\perp}$ transform[5] (see Fig. 3.1) in the QROM can also be applied to $\mathsf{FO}^{\mathsf{kyber}}$. Now Saito *et al.* [6] essentially use the same trick as Jiang *et al.* [5] use for the $\mathsf{FO}^{\not\perp}$ transform (see Subsection 3.1.2 above for details of

[4] Technically, the "message" m is a hash of a random message $m' \leftarrow\$ \{0,1\}^{256}$ (see Line 2 in "$\mathsf{Encap}(\mathsf{pk})$", Fig. 3.6). However for the purpose of this overview, we will ignore this detail. But our subsequent formal analysis of Kyber in Subsection 3.2.3 will take this extra hash into account.

[5] More formally, Saito *et al.* [6] analyze the so-called "$\mathsf{U}_m^{\not\perp}$" transform of [4], with an additional "re-encryption check" during decapsulation (see Lines 4 and 5 in "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.1); $\mathsf{U}_m^{\not\perp}$ can essentially be seen as $\mathsf{FO}_m^{\not\perp}$ acting on *deterministic* base PKE schemes.

the trick) to simulate decapsulation oracles without knowledge of the secret key in their IND-CCA security reduction. (In fact, Bindel *et al.* [51] show that concrete IND-CCA security of the $\mathsf{FO}_m^{\not\perp}$ transform is essentially equivalent to that of $\mathsf{FO}^{\not\perp}$ in the QROM.) Hence as already argued in Subsection 3.1.2, this trick cannot be extended in a straightforward fashion to $\mathsf{FO}^{\mathsf{kyber}}$ because of the extra hashes in key-derivation.

At least for $\mathsf{FO}^{\mathsf{frodo}}$, we were able to account for the "nested" hashing of message because it was *length-preserving*. However, this is not the case for "$H(c)$" in Kyber. In fact, we believe that an IND-CCA security reduction for Kyber in the QROM, following the proof strategies used for $\mathsf{FO}_m^{\not\perp}$ (and $\mathsf{FO}^{\not\perp}$) in the literature, would need to rely on the collision-resistance of H when modelled as a quantum random oracle; and this would necessarily result in an additive "collision-resistance" term in the security bounds for Kyber, thereby deviating from the tight bounds we have for the aforementioned implicitly-rejecting standard FO transforms in the literature (e.g., in [51, 52]). But modulo this additive term, we will now describe our approach to establish tight IND-CCA security for Kyber in the QROM.

We begin by first describing an *alternative* – and "*simpler*" – approach to prove IND-CCA security of Kyber in the QROM, and then contrasting it with our approach. Before discussing the tweaked FO variant, namely $\mathsf{FO}^{\mathsf{kyber}}$, used by Kyber (described in Figure 3.6), let us again consider some standard FO transforms in the literature [3, 4], namely the explicitly-rejecting $\mathsf{FO}_m^{\perp}$ and the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$, described in Figure 3.1.

For ease of exposition, we now consider a simplified[6] version of $\mathsf{FO}^{\mathsf{kyber}}$ where the only main difference compared to $\mathsf{FO}_m^{\perp}$ is that, instead of stopping at "$\overline{k} \leftarrow G_k(m)$" (Line 4 in $\mathsf{Encap}(\mathsf{pk})$, Fig. 3.1) during encapsulation, there is an extra layer of hashing to compute the final encapsulated key. Namely, Kyber outputs keys of the form "$k \leftarrow H'(\overline{k}, H(c))$" where H, H' are two additional hash functions; decapsulation proceeds analogously where instead of returning a $\perp$ when rejecting a ciphertext, Kyber *implicitly* rejects by returning $H'(s, H(c))$. Hence, (this simplified version of) Kyber can be seen as a "wrapper" scheme w.r.t. the $\mathsf{FO}_m^{\perp}$ KEM with appropriate modifications to the encapsulation and decapsulation steps. As a result, the IND-CCA security of Kyber can be easily shown by relying on the IND-CCA security of the underlying $\mathsf{FO}_m^{\perp}$ KEM.

To sketch out the proof, we start with the IND-CCA security game w.r.t. (the simplified) Kyber where the adversary gets a challenge ciphertext

[6] In this simplified variant, we are ignoring the additional hashes of the message m and public key pk in Lines 2 and 3 respectively in $\mathsf{Encap}(\mathsf{pk})$, Figure 3.6.

c^* and the *real* encapsulated key "$H'(\overline{k}^*, H(c^*))$" (refer to Subsection 2.3.2 for a precise description of the IND-CCA security games for KEMs). We then modify the game via the following "hybrids":

1. In the first hybrid, we provide the adversary with a new encapsulated key "$H'(\overline{k}', H(c^*))$", where $\overline{k}'$ is an independent and uniformly random value. This modification is justified by relying on IND-CCA security of the underlying $\mathsf{FO}_m^\perp$ KEM. Because note that $\overline{k}^*$ can be seen as the "real" encapsulated key of the $\mathsf{FO}_m^\perp$ KEM and $\overline{k}'$ a "random" key, and IND-CCA security of $\mathsf{FO}_m^\perp$ implies (computational) indistinguishability of both these keys. One important thing worth noting here is that in the reduction to IND-CCA security of $\mathsf{FO}_m^\perp$, we can simulate the decapsulation oracle of Kyber as follows. We first sample the secret $s \leftarrow\$ \mathcal{M}$. Then to simulate the "Kyber-decapsulation" of a ciphertext c, we first perform the "$\mathsf{FO}_m^\perp$-decapsulation" of c: if the result is a key $\overline{k}$, we return the "Kyber-key" as $H'(\overline{k}, H(c))$; if the result is $\perp$, we return the "Kyber-key" as $H'(s, H(c))$. Note that for this reduction to work, it is crucial that the underlying FO transform, $\mathsf{FO}_m^\perp$, is explicitly rejecting, in order to perfectly simulate the rejection of ciphertexts during decapsulation.

2. In the second and final hybrid, we again switch back to the IND-CCA security game w.r.t. Kyber where the adversary gets a uniformly *random* encapsulated key "$\overline{k}$" which is independent of c^*. This modification is again justified by relying on the *pseudorandomness* provided by the quantum random oracle $H'(\overline{k}', \cdot)$ (see Lemma 2) – i.e., since the "PRF key" $\overline{k}'$ is independent of c^*, one can argue the (statistical) indistinguishability of the keys "$H'(\overline{k}', H(c^*))$" and "$\overline{k}$".

The IND-CCA security of (the simplified) Kyber in the QROM hence follows since the adversary cannot efficiently distinguish between the real and random encapsulated keys "$H'(\overline{k}^*, H(c^*))$" and "$\overline{k}$" respectively in the above hybrids.

However, a major issue with the above approach to prove *concrete* (and *tight*) IND-CCA security of Kyber is related to our dependence on the IND-CCA security of $\mathsf{FO}_m^\perp$ *in the QROM* in the first place. IND-CCA security of the $\mathsf{FO}_m^\perp$ transform, with concrete bounds, has been notoriously hard to prove in the QROM – in contrast to its *implicitly-rejecting* variant, i.e., $\mathsf{FO}_m^{\not\perp}$. Namely, a long sequence of prior works [5, 6, 51, 52, 55] provided concrete IND-CCA security proofs for $\mathsf{FO}_m^{\not\perp}$ in the QROM, with each follow-up improving the tightness of the corresponding reduction. For example, Kuchta *et al.* [52] were the first to provide a security proof that avoided a square-root advantage loss w.r.t. the weak (IND-CPA/OW-CPA) security of the underlying PKE scheme; this loss seemed inherent with previous reductions for the FO transforms in the QROM. To also showcase the relative simplicity of analyzing the IND-CCA security of $\mathsf{FO}_m^{\not\perp}$ in the QROM, Unruh [56] showed a framework for *formally verifying* the corresponding post-quantum security proof of the implicitly-rejecting transform provided in [55].

When it comes to the *explicitly-rejecting* $\mathsf{FO}_m^{\perp}$ transform, the story is arguably more complicated. Looking at prior work, some starting steps were taken in [4, 37, 47, 50] in this regard wherein concrete IND-CCA security proofs for *modified* versions of the $\mathsf{FO}_m^{\perp}$ transform – which include an additional "key confirmation" hash in the ciphertext – were provided (however, security proofs in [37, 50] were later found to have bugs in them [37]). The *unmodified* $\mathsf{FO}_m^{\perp}$ transform was later analyzed in [57, 58] in the QROM; however, the provided security proofs had some subtle gaps [7]. These gaps were subsequently resolved in [7, 8] resulting in the first IND-CCA security proofs for the original $\mathsf{FO}_m^{\perp}$ transform in the QROM with concrete bounds. Quite recently, at the time of writing this thesis, Ge *et al.* [59] provided a tighter security proof for the explicitly-rejecting $\mathsf{FO}_m^{\perp}$ which avoided a square-root advantage loss w.r.t. passive security of the underlying PKE scheme, similar to the result of Kuchta *et al.* [52] for the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$.

However, the above IND-CCA security analyses of $\mathsf{FO}_m^{\perp}$ in [7, 8, 59] assume certain computational and statistical properties of the underlying PKE scheme which are not well-studied w.r.t. the NIST PQC standard Kyber. These properties include *(weak) γ-spreadness*, so-called *Find Failing Plaintext (FFP) security* (as introduced in [8]), etc. This is in contrast to the aforementioned QROM security proofs for the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$ in the literature. For example, the tight IND-CCA security proof of $\mathsf{FO}_m^{\not\perp}$ by Kuchta *et al.* [52] makes an additional assumption on the underlying PKE scheme – namely, that the scheme satisfy a property called *injectivity* (as defined in [51])[7]; in subsequent work, Ding *et al.* [53] rigorously established injectivity for Kyber by providing concrete bounds.

This brings us to our approach to establish tight IND-CCA security for Kyber in the QROM. In essence, we provide a way to salvage the above "wrapper-based" approach – *even when the underlying FO transform is implicitly-rejecting*. As noted in the above reduction, we crucially relied on the explicit-rejection of $\mathsf{FO}_m^{\perp}$ in order to perfectly simulate decapsulation oracles. But if we start with the $\mathsf{FO}_m^{\not\perp}$ transform, it is not so straightforward how to simulate the "Kyber-decapsulation" oracle using the "$\mathsf{FO}_m^{\not\perp}$-decapsulation" oracle especially when the latter oracle rejects ciphertexts; as described in Figure 3.1 (Line 9), the rejection output $G_k(s, c)$ still "looks" like a valid key.

To resolve the above simulation issue, we start with the $\mathsf{FO}_m^{\not\perp}$ transform and modify its decapsulation algorithm in a way such that the overall IND-CCA security of the transform in the QROM is affected negligibly (in a statistical sense). Similarly, we also modify the decapsulation procedure used in the actual Kyber scheme such that (i) the IND-CCA security of the original and modified schemes are statistically equivalent, and (ii) the IND-CCA security of the modified scheme can be reduced to the IND-CCA security of the modified $\mathsf{FO}_m^{\not\perp}$ transform wherein we can now simulate the "modified-Kyber-decapsulation" oracle using the "modified-$\mathsf{FO}_m^{\not\perp}$-decapsulation" oracle perfectly in the corresponding reduction. It is then not hard to see that this *indirectly* allows us to base IND-CCA security of the actual Kyber scheme on that of the unmodified $\mathsf{FO}_m^{\not\perp}$ transform, with a negligible loss in tightness.

Finally, another advantage of using our "wrapper-based" approach w.r.t. the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$ transform – when compared to the explicitly-rejecting $\mathsf{FO}_m^{\perp}$ – is related to establishing "beyond IND-CCA" security properties for Kyber in the post-quantum setting. As will be seen in Chapters 4 and 5, we will focus on one such property called *anonymity* (or, *key-privacy*) as introduced in [12]; roughly speaking, anonymity guarantees that a ciphertext does not reveal the public key used to generate it, thereby hiding the recipient's identity. Now given our discussion above on the relative ease

[7] More technically, Kuchta *et al.* [52] require a *deterministic* version of the PKE scheme, where the random coins used in encryption are derived from hashing the input message (cf. the "T" transform defined in [4] and also described in Fig. 4.9), to be injective. Roughly speaking, such deterministic PKE schemes are said to be *η-injective* if the probability that the deterministic encryption function for a random choice of public key (which is honestly generated) and random choice of the above hash function (modelled as a random oracle) is *not* injective is upper-bounded by η.

```
$\mathsf{KGen}'$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
    $s \leftarrow\!\!\$\ \{0,1\}^{256}$
    $\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$
    $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$
    return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Encap}(\mathsf{pk})$
    $m \leftarrow\!\!\$\ \{0,1\}^{256}$
    $h \leftarrow H(\mathsf{pk})$
    $(\bar{k}, r) \leftarrow G(m, h)$
    $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
    return $(c, \bar{k})$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
    Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
    $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
    $(\bar{k}', r') \leftarrow G(m', h)$
    $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
    if $c' = c$ then
        return $\bar{k}'$
    else return $H'(s, c)$
```

Figure 3.7: The $\mathsf{PKE} \rightarrow \mathsf{KEM}$ transform $\mathsf{FO}_{\mathrm{pre}}^{\mathrm{kyber}}$.

to prove IND-CCA security properties of $\mathsf{FO}_m^{\not\perp}$ when compared to $\mathsf{FO}_m^{\perp}$ in the QROM literature, subsequent works [60] went on to prove "beyond IND-CCA" properties such as anonymity for the implicitly-rejecting transform in the QROM. In Chapter 5, we will then show how to adapt the above "wrapper-based" approach to also establish post-quantum anonymity of Kyber.

3.2.3 *Security Analysis*

Towards proving the *concrete* IND-CCA security of Kyber in the QROM, we first consider an intermediate $\mathsf{PKE} \rightarrow \mathsf{KEM}$ transform $\mathsf{FO}_{\mathrm{pre}}^{\mathrm{kyber}}$, described in Figure 3.7. Note that the $\mathsf{FO}_{\mathrm{pre}}^{\mathrm{kyber}}$ transform is essentially identical to the $\mathsf{FO}_m^{\not\perp}$ transform (described in Fig. 3.1) in the context of proving IND-CCA security of the obtained KEM. That is, the existing IND-CCA security theorems w.r.t. $\mathsf{FO}_m^{\not\perp}$ in the QROM derived in the literature – e.g., in [5, 6, 51, 52], as discussed above – apply to $\mathsf{FO}_{\mathrm{pre}}^{\mathrm{kyber}}$ *as-it-is* because of the following reasons:

- Note that $\mathsf{FO}_{\mathrm{pre}}^{\mathrm{kyber}}$ uses a single hash function G to compute both the encapsulated key $\bar{k}$ and the random coins r for the deterministic encryption of m during encapsulation, whereas $\mathsf{FO}_m^{\not\perp}$ uses two separate hash functions for the same. However, these two computations are equivalent when the corresponding hash functions are modeled as independent random oracles with appropriate output lengths.

- Similarly, $\mathsf{FO}^{\mathrm{kyber}}_{\mathrm{pre}}$ uses the hash $H(\mathsf{pk})$ to compute $\bar{k}$ and r during encapsulation (and $H(\mathsf{pk})$ is also included in the final KEM's secret key sk'), in contrast to $\mathsf{FO}^{\not\perp}_{m}$. But this change preserves the relevant IND-CCA theorems from $\mathsf{FO}^{\not\perp}_{m}$ to $\mathsf{FO}^{\mathrm{kyber}}_{\mathrm{pre}}$ with trivial changes to the corresponding proofs, to accommodate the inclusion of $H(\mathsf{pk})$, because the IND-CCA security game only involves a *single* user's public key pk – as opposed to multi-user security notions such as *anonymity* (or *ANO-CCA security* as will be formalized in Chapters 4 and 5) which involves *two* public-keys.

Now let $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ be the KEM obtained by applying $\mathsf{FO}^{\mathrm{kyber}}_{\mathrm{pre}}$ transform on the base PKE scheme $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, i.e., $\overline{\mathsf{Kyber}}.\mathsf{KEM} = \mathsf{FO}^{\mathrm{kyber}}_{\mathrm{pre}}[\mathsf{Kyber.PKE}, G, H, H']$. In the following, we show that IND-CCA security of the actual $\mathsf{Kyber.KEM} = \mathsf{FO}^{\mathrm{kyber}}[\mathsf{Kyber.PKE}, G, H, H']$ (see Figure 3.6) in the QROM can be tightly reduced to the corresponding IND-CCA security of $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ (modulo some additive terms in the security bounds).

Theorem 2. *For any* IND-CCA *adversary $\mathcal{A}$ against the scheme* $\mathsf{Kyber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_H and $q_{H'}$ queries to the quantum random oracles H and H' respectively, there exists an* IND-CCA *adversary $\overline{\mathcal{A}}$ against* $\overline{\mathsf{Kyber}}.\mathsf{KEM} = (\overline{\mathsf{KGen}}', \overline{\mathsf{Encap}}, \overline{\mathsf{Decap}})$ *issuing at most q'_D classical queries to the decapsulation oracle, and q'_H and $q'_{H'}$ queries to the quantum random oracles H and H' respectively – with $q'_{H'} \leq q_{H'}$ and $(q'_D + q'_H) \leq q_H$ – such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}}) + \frac{7q_{H'} + 2q_H}{2^{128}} + \frac{C(q_H+1)^3}{2^{256}},$$

where C (< 648) is the constant from Lemma 6, and the running time of $\overline{\mathcal{A}}$ is about the same as that of $\mathcal{A}$.

The proof essentially follows the "wrapper-based" approach described in Subsection 3.2.2 above but with respect to the *implicitly-rejecting* $\mathsf{FO}^{\not\perp}_{m}$ transform. Formal details follow.

Proof. Denote $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$, $\Omega_{\mathbf{H}'}$, $\Omega_{\mathbf{H}''}$ and $\Omega_{\overline{\mathbf{H}}}$ to be the set of all functions $\mathbf{G} : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{H} : \{0,1\}^{256} \cup \mathcal{PK} \cup \mathcal{C} \to \{0,1\}^{256}$, $\mathbf{H}' : \{0,1\}^{256} \times (\{0,1\}^{256} \cup \mathcal{C}) \to \{0,1\}^{256}$, $\mathbf{H}'' : \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ and $\overline{\mathbf{H}} : \{0,1\}^{256} \to \{0,1\}^{256}$ respectively, where $\mathcal{PK}$ is the space of all $\mathsf{Kyber.PKE}$ public keys and $\mathcal{C}$ is the ciphertext space of $\mathsf{Kyber.PKE}$.

Let $\overline{\mathcal{A}}$ be an IND-CCA adversary against $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ issuing at most q'_D classical queries to the decapsulation oracles, and q'_H and $q'_{H'}$ queries to the

```
Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$
    $G \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H' \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
    $H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H}''}; \overline{H} \leftarrow\!\!\$\ \Omega_{\overline{\mathbf{H}}}$
    $(\mathsf{pk}, \mathsf{sk}') \leftarrow \overline{\mathsf{KGen}}'$
    $(c^*, \overline{k}_0^*) \leftarrow \overline{\mathsf{Encap}}(\mathsf{pk})$
    $\overline{k}_1^* \leftarrow\!\!\$\ \{0,1\}^{256}$
    $b \leftarrow\!\!\$\ \{0,1\}$
    $b' \leftarrow \overline{\mathcal{A}}^{G,H,H',\overline{\textsc{Decaps}_{c^*}}}(\mathsf{pk}, c^*, \overline{k}_b^*)$
    return $[b' = b]$

$\overline{\textsc{Decaps}_a}(c)$
    if $c = a$ then return $\perp$
    Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
    $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
    $(\overline{k}', r') \leftarrow G(m', h)$
    $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
    if $c' = c$ then
        return $\overline{k}'$
    else return $H'(s, c)$    // $\overline{\mathsf{G}}_0$
    else return $H''(c)$    // $\overline{\mathsf{G}}_1$
    else return $\overline{H}(H(c))$    // $\overline{\mathsf{G}}_2$
```

FIGURE 3.8: Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$ for the proof of Theorem 2. Here Enc and Dec are the encryption and decryption algorithms of the base $\mathsf{Kyber.PKE}$ scheme.

quantum random oracles H and H' respectively. Consider the sequence of games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$ described in Figure 3.8 which only differ in the way their corresponding decapsulation oracles $\overline{\textsc{Decaps}_{c^*}}$ reject invalid ciphertexts.

Game $\overline{\mathsf{G}}_0$: This game is exactly the IND-CCA game for $\overline{\mathsf{Kyber}}.\mathsf{KEM}$. Hence,

$$\left|\Pr[\overline{\mathsf{G}}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}}).$$

Game $\overline{\mathsf{G}}_1$: In this game, the $\overline{\textsc{Decaps}_{c^*}}$ oracle is modified such that $H''(c)$ is returned instead of $H'(s, c)$ for an invalid ciphertext c, where H'' is a fresh *internal* random oracle not directly accessible to $\overline{\mathcal{A}}$. Using Lemma 2 w.r.t. the pseudorandomness of $H'(s, \cdot)$ during decapsulation, where we have the "PRF key" $s \leftarrow\!\!\$\ \{0,1\}^{256}$, it is not hard to obtain the following via a straightforward reduction:

$$\left|\Pr[\overline{\mathsf{G}}_1 = 1] - \Pr[\overline{\mathsf{G}}_0 = 1]\right| \leq \frac{2q'_{H'}}{2^{128}}.$$

Game $\overline{\mathsf{G}}_2$: In this game, we again modify the $\overline{\textsc{Decaps}_{c^*}}$ oracle such that $\overline{H}(H(c))$ is returned instead of $H''(c)$ for an invalid ciphertext c, where $\overline{H}$ is another fresh internal random oracle not directly accessible to $\overline{\mathcal{A}}$. Note that the oracles H'' and $\overline{H}$ are only accessible to $\overline{\mathcal{A}}$ *indirectly* via the $\overline{\textsc{Decaps}_{c^*}}$ oracle. Now in the view of adversary $\overline{\mathcal{A}}$, the output distributions of the

$\overline{\textsc{Decaps}_{c^*}}$ oracle in games $\overline{\mathsf{G}}_1$ and $\overline{\mathsf{G}}_2$ with regards to invalid ciphertexts c are identical *unless* $\overline{\mathcal{A}}$ queries the decapsulations of two invalid ciphertexts c_1 and c_2 such that $H(c_1) = H(c_2)$ (and $c_1 \neq c_2$). Since decapsulation queries are considered to be classical in the QROM, we can bound the probability of such an event by collision-resistance of the QRO H – as described in Lemma 6 – again via a straightforward reduction. Hence, we have[8],

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \Pr[\overline{\mathsf{G}}_1 = 1]\right| \leq \frac{C(q'_H + q'_D + 1)^3}{2^{256}},$$

where C (< 648) is the constant from Lemma 6.

Hence by collecting the above bounds, we obtain

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right| \leq \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\overline{\mathcal{A}}) + \frac{2q'_{H'}}{2^{128}} + \frac{C(q'_H + q'_D + 1)^3}{2^{256}}, \quad (3.1)$$

which will be useful shortly when we now focus on proving concrete IND-CCA security of the *actual* scheme of Kyber.

Let $\mathcal{A}$ be an IND-CCA adversary against Kyber.KEM issuing at most q_H and $q_{H'}$ queries to the quantum random oracles H and H' respectively. Consider the sequence of games G_0 – G_8 described in Figure 3.9.

Game G_0**:** This game is basically the IND-CCA game for Kyber.KEM where the adversary $\mathcal{A}$ gets the "real" encapsulated key k^*, i.e., $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$.

Game G_1**:** Here we essentially do not execute the "$m \leftarrow H(m)$" step during encapsulation (Line 2 in "$\mathsf{Encap}(\mathsf{pk})$", Fig. 3.6) in this game's setup. We now use the original OW2H lemma (Lemma 3) to bound the difference in $\mathcal{A}$'s "behavior" in games G_0 and G_1. In the context of applying Lemma 3, let $x := m_0^* \leftarrow\!\$ \{0,1\}^{256}$ and $y := m_1^* \leftarrow\!\$ \{0,1\}^{256}$, and consider an oracle algorithm A^H making at-most q_H queries to H such that $A^H(m_0^*, H(m_0^*))$ simulates the game G_0 towards $\mathcal{A}$ and $A^H(m_0^*, m_1^*)$ simulates G_1 towards $\mathcal{A}$.[9] To be more specific, A^H sets "m^*" in Line 4, Fig. 3.9, to be its second input (either $H(m_0^*)$ or m_1^*) when simulating the appropriate game (G_0 or G_1, respectively) towards $\mathcal{A}$.

[8] Recall from our convention (described in Section 2.3) that q'_H counts the total number of times H is invoked in the game $\overline{\mathsf{G}}_0$. However in $\overline{\mathsf{G}}_2$, H is *additionally* invoked when $\overline{\mathcal{A}}$ queries the decapsulation of an invalid ciphertext. Hence, H is queried at most $(q'_H + q'_D)$ many times in $\overline{\mathsf{G}}_2$ in the context of applying Lemma 6.

[9] Technically, we have the domain of random oracle H to be $\{0,1\}^{256} \cup \mathcal{PK} \cup \mathcal{C}$. However in Kyber, we have $\{0,1\}^{256} \cap (\mathcal{PK} \cup \mathcal{C}) = \phi$. Because of this domain separation, we can effectively apply Lemma 3 by restricting the domain of H to be $\mathcal{X} := \{0,1\}^{256}$.

```
Games $\mathsf{G}_0 - \mathsf{G}_8$
$G \xleftarrow{\$} \Omega_{\mathbf{G}}; H \xleftarrow{\$} \Omega_{\mathbf{H}}; H' \xleftarrow{\$} \Omega_{\mathbf{H'}}$
$H'' \xleftarrow{\$} \Omega_{\mathbf{H''}}; \overline{H} \xleftarrow{\$} \Omega_{\overline{\mathbf{H}}}$
$(\mathsf{pk}, \mathsf{sk}') \leftarrow \mathsf{KGen}'$
$m^* \xleftarrow{\$} \{0,1\}^{256}$
$m^* \leftarrow H(m^*)$   // $\mathsf{G}_0, \mathsf{G}_8$
$(\overline{k}_0^*, r^*) \leftarrow G(m^*, H(\mathsf{pk}))$
$\overline{k}_1^* \xleftarrow{\$} \{0,1\}^{256}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$
$k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$   // $\mathsf{G}_0 - \mathsf{G}_3$
$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$   // $\mathsf{G}_4$
$k^* \xleftarrow{\$} \{0,1\}^{256}$   // $\mathsf{G}_5 - \mathsf{G}_8$
$b' \leftarrow \mathcal{A}^{G,H,H',\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k^*)$
return $b'$
```

```
$\textsc{Decaps}_a(c)$
if $c = a$ then return $\bot$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
    return $H'(\overline{k}', H(c))$
else
    return $H'(s, H(c))$   // $\mathsf{G}_0$–$\mathsf{G}_1$, $\mathsf{G}_7$–$\mathsf{G}_8$
    return $H''(H(c))$   // $\mathsf{G}_2, \mathsf{G}_6$
    return $H'(\overline{H}(H(c)), H(c))$   // $\mathsf{G}_3$–$\mathsf{G}_5$
```

FIGURE 3.9: Games $\mathsf{G}_0 - \mathsf{G}_8$ for the proof of Theorem 2. Here Enc and Dec are the encryption and decryption algorithms of the base $\mathsf{Kyber.PKE}$ scheme.

Again in the context of Lemma 3, it is not hard to see that $\Pr[\mathsf{G}_0 = 1] = P_A^1$ and $\Pr[\mathsf{G}_1 = 1] = P_A^2$. Regarding the probability P_B, note that during $A^H(m_0^*, m_1^*)$'s simulation of game G_1 towards $\mathcal{A}$, the view of $\mathcal{A}$ is completely independent of the value m_0^* $(= x) \leftarrow\$ \{0,1\}^{256}$. Hence, we have $P_B = \frac{1}{2^{256}}$ which leads to

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{2q_H}{2^{128}} \ (= 2q_H\sqrt{P_B}).$$

Game G_2**:** In this game, the $\textsc{Decaps}_{c^*}$ oracle is modified such that $H''(H(c))$ is returned instead of $H'(s, H(c))$ for an invalid ciphertext c, where H'' is a fresh *internal* random oracle not directly accessible to $\mathcal{A}$. Similar to the $\mathsf{G}_0 \to \mathsf{G}_1$ "hop" above, by using Lemma 2 w.r.t. the pseudorandomness of $H'(s, \cdot)$–this time on inputs of the form "$H(c)$"–during decapsulation, it is not hard to obtain:

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_3**:** In this game, we again modify the $\textsc{Decaps}_{c^*}$ oracle such that $H'(\overline{H}(H(c)), H(c))$ is returned instead of $H''(H(c))$ for an invalid ciphertext c, where $\overline{H}$ is another fresh internal random oracle not directly accessible to $\mathcal{A}$. Here we use the generalized OW2H lemma (Lemma 4) to bound the difference in $\mathcal{A}$'s behavior in games G_2 and G_3.

In the context of Lemma 4, note that the oracle algorithm needs to distinguish the pair of random functions $(H''(\cdot), H')$ in G_2 from the pair $(H'(\overline{H}(\cdot), \cdot), H')$ in G_3. But it is not hard to see that this is the same as distinguishing (H'', H') in G_2 from (H'', G') in G_3, where the oracle G' is obtained by *reprogramming* H' on inputs of the form "$(\overline{H}(x), x)$" with $x \in \{0,1\}^{256}$; namely, we have

$$G'(y) = \begin{cases} H''(x) & \text{if } y \text{ is of the form } (\overline{H}(x), x) \text{ with } x \in \{0,1\}^{256} \\ H'(y) & \text{otherwise.} \end{cases}$$

So again in the context of applying Lemma 4, consider an oracle algorithm A which has quantum access to either (H'', H') or (H'', G') such that $A^{H'',H'}$ and $A^{H'',G'}$ simulate G_2 and G_3 respectively towards $\mathcal{A}$, while making $q_{H'}$ oracle queries.[10] Note that the set of differences between the H' and G'

[10] For example, A uses the first oracle H'' to simulate $\textsc{Decaps}_{c^*}$ in Figure 3.9 w.r.t. invalid ciphertexts c; given such a decapsulation query c from $\mathcal{A}$, the algorithm A returns $H''(H(c))$, where the oracle H is sampled independently by A at the games' setup.

oracles is $\mathcal{S} = \{(\overline{H}(x), x) \mid x \in \{0,1\}^{256}\}$. If we then set $\Pr[\mathsf{G}_2 = 1] = P_{\text{left}}$ and $\Pr[\mathsf{G}_3 = 1] = P_{\text{right}}$, from Lemma 4 we have $|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq 2q_{H'}\sqrt{P_{\text{guess}}}$. Regarding P_{guess}, note that during $A^{H'',H'}$'s simulation of G_2 towards the adversary $\mathcal{A}$, the view of $\mathcal{A}$ is completely independent of the (internal) random oracle $\overline{H}$. Hence the probability that measurement of a random H'-oracle query in G_2 will be of the form $(\overline{H}(x), x)$ (with $x \in \{0,1\}^{256}$) is at-most $\frac{1}{2^{256}}$, i.e., $P_{\text{guess}} \leq \frac{1}{2^{256}}$, since $\overline{H}(x)$ will be a fresh uniformly random value in $\{0,1\}^{256}$. Therefore,

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_4**:** In this game, we generate the encapsulated key k^* in the setup as "$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$" instead of "$k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$" where we have $(\overline{k}_0^*, r^*) \leftarrow\!\!\$\ G(m^*, H(pk))$ and $\overline{k}_1^* \leftarrow\!\!\$\ \{0,1\}^{256}$. Here we make use of our analysis of the $\mathsf{FO}_{\text{pre}}^{\text{kyber}}$ transform above.

Consider the game $\overline{\mathsf{G}}_2$ "played" by adversary $\overline{\mathcal{A}}$ in Fig. 3.8 w.r.t. $\overline{\mathsf{Kyber}}.\mathsf{KEM}$. Depending on whether $\overline{\mathcal{A}}$ gets the "real *pre-key*" $\overline{k}_0^*$ or the "random *pre-key*" $\overline{k}_1^*$ from its challenger, it can simulate the game G_3 or G_4 respectively towards $\mathcal{A}$. Namely, $\overline{\mathcal{A}}^{H,H'}(c^*, \overline{k}_b^*)$ computes the encapsulated key k^* as $k^* \leftarrow H'(\overline{k}_b^*, H(c^*))$ (where b is the bit sampled by $\overline{\mathcal{A}}$'s challenger in Fig. 3.8) and sends it to $\mathcal{A}$ during the games' setup. $\overline{\mathcal{A}}^{H,H',\overline{\textsc{Decaps}_{c^*}}}$ also simulates the decapsulation oracle in games G_3 and G_4 (see Fig. 3.9) as follows: given a decapsulation query c from $\mathcal{A}$, $\overline{\mathcal{A}}$ queries its *own* $\overline{\textsc{Decaps}_{c^*}}$ oracle in $\overline{\mathsf{G}}_2$ on c to obtain a key $\overline{k}'$–which can also be the value "$\overline{H}(H(c))$" if c is invalid (see Line 9 in "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.8)–and returns $H'(\overline{k}', H(c))$ to $\mathcal{A}$. Hence, it is not hard to see from this reduction that

$$\begin{aligned}|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| &= \left|\Pr[1 \leftarrow \overline{\mathcal{A}} \mid b = 1] - \Pr[1 \leftarrow \overline{\mathcal{A}} \mid b = 0]\right| \\ &= 2 \cdot \left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right|.\end{aligned}$$

By using Inequality (3.1) above w.r.t. our analysis of $\overline{\mathsf{Kyber.KEM}}$, we obtain[11]

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\overline{\mathsf{Kyber.KEM}}}(\overline{\mathcal{A}}) + \frac{4q_{H'}}{2^{128}} + \frac{2C(q_H + 1)^3}{2^{256}}.$$

Game G_5**:** Here we have the encapsulated key k^* in the setup to be an independent and uniformly random value, i.e., "$k^* \leftarrow\!\$ \{0,1\}^{256}$", instead of deriving it from H' as "$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$". Similar to the $\overline{\mathsf{G}}_0 \to \overline{\mathsf{G}}_1$ hop above, by using Lemma 2 w.r.t. the pseudorandomness of $H'(\overline{k}_1^*, \cdot)$–with "PRF key" $\overline{k}_1^* \leftarrow\!\$ \{0,1\}^{256}$–during setup, it is not hard to obtain:

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_6**:** In this game, we modify the $\textsc{Decaps}_{c^*}$ oracle such that $H''(H(c))$ is returned instead of $H'(\overline{H}(H(c)), H(c))$ for an invalid ciphertext c. In essence, we are reverting the changes introduced in the "$\mathsf{G}_2 \to \mathsf{G}_3$" hop. Hence, by applying a similar reasoning as that hop, we get

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_7**:** In this game, $\textsc{Decaps}_{c^*}$ oracle is modified such that $H'(s, H(c))$ is returned instead of $H''(H(c))$ for an invalid ciphertext c. Again in essence, we are reverting the changes introduced in the "$\mathsf{G}_1 \to \mathsf{G}_2$" hop. Hence, by using a similar reasoning as that hop–namely, pseudorandomness of the oracle $H'(s, \cdot)$ on inputs of the form "$H(c)$"–we obtain

$$|\Pr[\mathsf{G}_7 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_8**:** Here we re-introduce the "$m \leftarrow H(m)$" step during encapsulation (Line 2 in "$\mathsf{Encap(pk)}$", Fig. 3.6) in this game's setup, thereby reverting the changes introduced in the "$\mathsf{G}_0 \to \mathsf{G}_1$" hop. By applying Lemma 3 in a similar way as that hop, we get

$$|\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{2q_H}{2^{128}}.$$

[11] Here we replace the term "$q'_H + q'_D$" in Inequality (3.1) with "q_H". Recall from Footnote 8 of this chapter that $(q'_H + q'_D)$ is the maximum number of times oracle H is queried in $\overline{\mathsf{G}}_2$. But since the decapsulation algorithm of Kyber.KEM involves a single invocation of $H(\cdot)$ for each input ciphertext c (see "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.6), the quantity "q_H" *includes* the number of times H is queried by $\overline{\mathcal{A}}$ to answer decapsulation queries from $\mathcal{A}$ – following our convention w.r.t. counting the number of random oracle queries in security games (see Section 2.3).

Now note that G_8 is the IND-CCA game for Kyber.KEM where the adversary $\mathcal{A}$ gets a "random" encapsulated key k^*, i.e., $k^* \leftarrow\$ \{0,1\}^{256}$ (in contrast to getting the "real" encapsulated key in G_0). Hence, we have

$$2 \cdot \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) = |\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_0 = 1]| \,.$$

By collecting the above bounds, we obtain

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\overline{\mathsf{Kyber.KEM}}}(\overline{\mathcal{A}}) + \frac{7q_{H'} + 2q_H}{2^{128}} + \frac{C(q_H+1)^3}{2^{256}}. \quad (3.2)$$

□

The above result essentially states that the provable IND-CCA security guarantees enjoyed by the implicitly-rejecting $\mathsf{FO}^{\not\perp}_{m}$ – and as an extension, the $\mathsf{FO}^{\mathsf{kyber}}_{\mathsf{pre}}$-derived $\overline{\mathsf{Kyber.KEM}}$ – in the QROM discussed earlier (see Subsection 3.2.2) also apply to the actual Kyber.KEM with minimal loss in tightness. So one can simply "plug in" existing IND-CCA security reductions for $\mathsf{FO}^{\not\perp}_{m}$-derived KEMs – w.r.t. the passive OW-/IND-CPA security of the corresponding base PKE scheme – from the literature in Inequality (3.2) to obtain concrete IND-CCA security bounds for Kyber in the QROM.

For example, one can use the reduction in [5, Theorem 2] to show the existence of an IND-CPA adversary $\mathcal{B}$ against the base Kyber.PKE scheme, with its running time about the same as that of IND-CCA adversary $\overline{\mathcal{A}}$ above, such that[12]

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\overline{\mathsf{Kyber.KEM}}}(\overline{\mathcal{A}}) \leq 2q'_G\sqrt{\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{2q'_{H'}}{2^{128}} + 4q'_G\sqrt{\delta},$$

where q'_G is an upper bound on the number of queries made by $\overline{\mathcal{A}}$ to the QRO G, and δ is the correctness parameter of Kyber.PKE (see Definition 2). Regarding the above properties of Kyber.PKE, it was argued in [11, Theorem 1] that (in the (Q)ROM) Kyber.PKE is tightly IND-CPA secure under the MLWE hardness assumption, since under the MLWE assumption, the public-key and ciphertexts of Kyber.PKE are pseudorandom; also, the δ-correctness property of Kyber.PKE has been rigorously analyzed in [11, 61].

Now an advantage of using the reduction in [5, Theorem 2] is that it mainly uses the "One-Way to Hiding (OW2H) lemma" [36, 37] proof

[12] Technically, [5, Theorem 2] reduces the IND-CCA security of the KEM to the OW-CPA security of the underlying PKE scheme. But recall from Lemma 7 that IND-CPA security of a PKE scheme with a sufficiently large message space also implies its OW-CPA security (also note that in Kyber.PKE, the message space is $\{0,1\}^{256}$).

technique (see Lemmas 3 and 4) which is amenable to *formal verification* as shown by Unruh [56]. On the downside however, the reduction is non-tight in the sense that we incur a square-root advantage loss w.r.t. passive security of the underlying base PKE scheme.

One can instead use the tighter reduction in [52, Corollary 4.7] to essentially show that there exists an IND-CPA adversary $\mathcal{B}'$ against $\mathsf{Kyber.PKE}$, with running time at-most three times that of $\overline{\mathcal{A}}$, such that

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}}) \leq 8q'_G \cdot (q'_G + 1) \left(\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}') + \frac{8(3q'_G + 1)}{2^{256}} \right) + 6(3q'_G + q'_D) \cdot ((8q'_G + 1)\delta + \sqrt{3\eta}) + (4q'_G + 12) \cdot \eta + \frac{4q'_{H'}}{2^{128}},$$

where η is the "injectivity parameter" that was concretely established for a deterministic version of $\mathsf{Kyber.PKE}$[13] in [53], as mentioned in Subsection 3.2.2 above. Note that the reduction no longer incurs a square-root advantage loss w.r.t. IND-CPA security of $\mathsf{Kyber.PKE}$. More specifically, the reduction uses a variant of the OW2H technique known as the "Measure-Rewind-Measure (MRM)" OW2H lemma [52] to achieve tighter IND-CCA security bounds. However, to the best of our knowledge, there is currently no framework to formally verify applications of the MRM OW2H lemma in post-quantum security proofs – in contrast to Unruh's framework [56] w.r.t. the "plain" OW2H lemma. We leave the extension of Unruh's framework to cover the MRM variant as an interesting open problem.

Remark 1. It is worth mentioning that NIST has recently started plans [18, 19] to essentially replace the current tweaked FO transform $\mathsf{FO}^{\mathsf{kyber}}$ (see Fig. 3.6) used in Kyber with the transform $\mathsf{FO}^{\mathsf{kyber}}_{\mathsf{pre}}$ (see Fig. 3.7) above;[14] in other words, $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ would potentially be the new NIST PQC standard over $\mathsf{Kyber.KEM}$. In this context, our above IND-CCA security analysis of

[13] Namely, where encryption randomness r is derived from the hash function G; see Lines 3 and 4 in "$\mathsf{Encap}(\mathsf{pk})$" and "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.7.

[14] Technically, the new transform being considered for Kyber (see [19] for a formal description) slightly differs from $\mathsf{FO}^{\mathsf{kyber}}_{\mathsf{pre}}$ in the way invalid ciphertexts are rejected during decapsulation. Namely, in the new transform, the hash H' (see Line 7 in "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.7) is no longer used for rejection, and instead, hash G is used as follows: for an invalid ciphertext c (which fails the re-encryption check in Line 5, "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.7), we compute $(\overline{k}'', r'') \leftarrow G(s, c)$ and output the key $\overline{k}''$. However in Kyber, the sizes of ciphertexts are more than 512 bits which ensures proper domain separation when computing keys $\overline{k}'$ for valid ciphertexts ("$(\overline{k}', r') \leftarrow G(m', h)$"; Line 3 in "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.7) and keys $\overline{k}''$ for invalid ciphertexts ("$(\overline{k}'', r'') \leftarrow G(s, c)$") – same as in $\mathsf{FO}^{\mathsf{kyber}}_{\mathsf{pre}}$.

Kyber.KEM in the QROM can be seen as a "safety net" in case problems arise in NIST's aforementioned plans (patent issues, for example) and Kyber.KEM is once again picked to be the standard.

More importantly, NIST's decision above showcases the impact our work in this thesis has on the PQC standardization process; mainly, our observations in [25, 26] regarding the divergences between Kyber's variant of the FO transform and the standard FO transforms in the literature, and our subsequent arguments regarding the inapplicability of related proof strategies in the literature to concretely establish IND-CCA security of Kyber in the QROM – contrary to what was claimed in the specification document [11].

Remark 2. We would like to point out that the NIST PQC third-round finalist *Saber* [62] implements the same variant of FO transform as Kyber, i.e., $\mathsf{FO}^{\mathsf{kyber}}$, in its KEM construction. Hence, our above result on provable IND-CCA security of Kyber in the QROM also applies to Saber in a similar fashion (where we would instead need to rely on hardness of solving the so-called *module learning-with-rounding* problem [63] for IND-CPA security of the corresponding base PKE scheme).

3.2.4 *Related Work*

An alternative approach to prove IND-CCA security of Kyber in the QROM was suggested in [64], involving the *compressed oracle* technique introduced in [57]. More specifically, given two random oracles $H_1 : \{0,1\}^m \to \{0,1\}^n$, $H_2 : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n$, and a polynomial-sized stateless classical circuit C which has quantum access to H_1, H_2, it was shown in [57, Section 5] that the "domain extender" $C^{H_1,H_2}(x,y) = H_2(H_1(x),y)$ is *indifferentiable* from a quantum random oracle $H : \{0,1\}^{m+\ell} \to \{0,1\}^n$. Informally, indifferentiability guarantees that any efficient adversary cannot distinguish $\langle (H_1,H_2), C^{H_1,H_2} \rangle$ from $\langle \mathcal{S}^H, H \rangle$ where the simulator $\mathcal{S}$ queries H and simulates the oracles H_1, H_2.

Now note that in Kyber (Fig. 3.6, Line 6 of "$\mathsf{Encap}(\mathsf{pk})$"), the encapsulated keys are generated as "$k \leftarrow H'(\overline{k}, H(c))$" by hashing the "*pre-key*" $\overline{k}$ and a "nested hash" of the ciphertext, i.e., $H(c)$. And as noted above in Subsection 3.2.2 above, this nested hash $H(c)$ creates problems when extending prior QROM security analysis of (implicitly-rejecting) FO transforms in the literature to Kyber. However, since [57, Section 5] essentially shows that $H'(\overline{k}, H(c))$ is indifferentiable from $H''(\overline{k}, c)$, for a fresh random oracle H'', we can "ignore" the nested hash $H(c)$ in our analysis of Kyber; in fact, the resulting variant of the FO transform, where keys are derived as "$k \leftarrow H''(\bar{k}, c)$", is essentially used by FrodoKEM (see Figure 3.3) – and we already established concrete IND-CCA security of this variant in the QROM in Subsection 3.1.3 above. However, we make a couple of remarks regarding this matter:

- At a conceptual level, our IND-CCA security analysis of Kyber above relies on arguably simpler proof techniques than the ones introduced in [57]. Specifically, our reduction from IND-CCA security of Kyber.KEM to that of $\overline{\text{Kyber}}$.KEM in the QROM (Theorem 2) is based on the well-known "OW2H lemma" [36, 37] proof technique. And as mentioned in Subsection 3.2.3 above, Unruh [56] provided a framework for formally verifying security proofs that involve applications of the OW2H lemma in the QROM. Hence, this should make our overall security analysis of Kyber amenable to formal verification, thereby providing further confidence in our positive IND-CCA security results for the new NIST PQC standard.

- Quantitatively, if we rely on the above indifferentiability argument to analyze Kyber instead, then when switching from "$H'(\bar{k}, H(c))$" to "$H''(\bar{k}, c)$" we would incur an additive "indifferentiability" term $O(q^2/2^{n/2})$ (as specified in [57, Section 5]) as an overhead in our IND-CCA security bounds, where q is the number of adversarial QRO queries made to H, H', and $n = 256$ in Kyber. In contrast, our analysis of Kyber (Theorem 2) incurs an additive "collision-resistance (of H)" term $O(q^3/2^n)$ as an overhead – as can be seen in the bounds in Inequality (3.2). Hence, our concrete IND-CCA security analysis of Kyber allows for strictly more number of random oracle queries q when compared to the indifferentiability-based analysis (especially w.r.t. higher security level parameter sets for Kyber when the "correctness" term $O(q\sqrt{\delta})$ is no longer a limiting factor on q – e.g., $\delta = 2^{-164}, 2^{-174}$).

Recently, Chen *et al.* [65] analyzed the concrete IND-CCA security of Kyber in the QROM using another alternative approach; more specifically, it involves using a well-known indistinguishability result between random functions and random permutations in the quantum setting [40]. However, since their reduction needs to efficiently simulate a random *permutation* in the QROM, their resulting IND-CCA security bounds include an additive term $O(\sqrt{q^3/2^{128}})$ which significantly restricts the number of QRO queries

q an adversary can make – this is in contrast to the "collision-resistance" term $O(q^3/2^{256})$ in our obtained bounds in Subsection 3.2.3 above.

In more recent work, Barbosa and Hülsing [66] reduced the IND-CCA security of Kyber.KEM *directly* to OW-CPA security of a deterministic version of the base Kyber.PKE scheme (see Footnote 13 of this chapter) in the QROM – in contrast to our *indirect* approach above which relies on IND-CCA security of an intermediate $\overline{\mathsf{Kyber}}$.KEM. Their reduction roughly follows along similar lines of that in [51] for the $\mathsf{FO}_m^{\not\perp}$ transform. However, their security bounds incur two different additive "collision-resistance" terms when compared to a single such term in our bounds (Inequality (3.2)).

3.3 SUMMARY

In this chapter, we revisited the FO-variants implemented in two NIST PQC KEMs – i.e., Kyber, which was selected by NIST for standardization, and FrodoKEM, which is currently recommended by the German federal agency BSI. We argued how the differences between these FO-variants and the standard (implicitly-rejecting) FO transforms in the literature invalidate the QROM IND-CCA security claims of the above NIST candidates. Subsequently, we re-established the concrete IND-CCA security of Kyber and FrodoKEM in the QROM by carefully accounting for the aforementioned differences in our formal proofs.

Given the importance placed by standards bodies on the above two schemes, we hope that the results in this chapter provide confidence to cryptographic scheme designers in using these schemes in applications requiring IND-CCA security in a post-quantum setting.

4

ANONYMITY AND ROBUST ENHANCEMENTS, PART I: GENERIC RESULTS

Standards bodies worldwide such as NIST, ISO, ETSI and IETF are in the process of standardizing new post-quantum secure public-key encryption schemes and digital signatures. Focusing on the former primitive, a main security target of evaluation for the encryption schemes has been IND-CCA security. This was appropriate as a starting point because it suffices for many important use cases. However, since the post-quantum cryptographic standards are intended to be widely used for decades to come, it is important to study the above encryption schemes' fitness for emerging modern applications where security properties other than IND-CCA are required.

Two important security properties that go beyond IND-CCA security are *anonymity* (or key privacy) and *robustness*. Anonymity was first formalised in the public key setting in [12]. Roughly, a PKE scheme is anonymous if a ciphertext does not leak anything about which public key was used to create it; strong forms of anonymity equip the adversary with a decryption oracle. Anonymous PKE is a fundamental component of several deployed anonymity systems, most notably anonymous cryptocurrencies like Zcash [67]. It is also important in building anonymous broadcast encryption schemes [68, 69], anonymous credential systems [14] and auction protocols [70]. Robustness for PKE, first formalised in [13], goes hand-in-hand with anonymity. Suppose a party equipped with a private key receives a ciphertext for an anonymous PKE scheme. In the absence of other information, how does a party decide that it is the intended receiver of that ciphertext? The standard approach is to perform trial decryption. Robustness provides an assurance that this process does not go wrong – that the receiver is not fooled into accepting a plaintext intended for someone else. Robustness is also important for maintaining consistency in searchable encryption [71] and ensuring auction bid correctness [70]. Various robustness notions for PKE were studied in [13], while stronger notions were introduced in [72]; the symmetric setting was treated in [73–76].

However, there is almost no work – prior to this thesis – that shows how to build anonymous, robust post-quantum PKE schemes. Nor is it known whether the candidate schemes considered for standardization –

in particular, candidates shortlisted by NIST in the final-round of its PQC competition – meet these extended notions. The only directly relevant work is by Mohassel [77], who showed a number of foundational results on anonymity and robustness of hybrid PKE schemes built via the KEM-DEM paradigm. Our work is influenced by Mohassel's general approach; however, Mohassel only considers KEMs that are directly constructed from strongly-secure PKE schemes.[1] This makes the results of [77] inapplicable to NIST candidates, for a few reasons. First, the NIST final-round candidates are all KEMs, not PKE schemes, so there is a basic syntactic mismatch. Second, the base PKE schemes used within the candidate KEMs are only weakly (i.e., OW-/IND-CPA) secure, but [77] relies on the starting PKE having IND-CCA security. Finally, [77] only analyzes *explicit rejection* KEMs, for which decapsulation can fail, but all the NIST final-round candidates except the alternate candidate HQC [78] are implicit rejection KEMs that never output the error symbol $\perp$. This means that such implicit rejection NIST PQC KEMs cannot be even weakly robust, while the constructions of [77] all start from robust KEMs.

One of the negative results of [77] is that even if a KEM enjoys a strong anonymity property, the hybrid PKE scheme that results from applying the standard KEM-DEM paradigm may not be anonymous. This is concerning, since it indicates that if one only focuses on KEMs in the NIST PQC standardization process, rather than the PKE schemes that will inevitably be built from them using the standard KEM-DEM approach, then there is no guarantee that desired security properties will actually carry over. Thus, one must dig into a KEM's internals if the target is to achieve anonymous hybrid PKE.

In fact, all the NIST final-round candidates in the KEM/PKE category are constructed using variants of the Fujisaki-Okamato (FO) transform [1, 2, 79] as described in Chapter 3. The FO transform and variants of it have been heavily analysed in the literature (e.g., in [4–6, 50, 55] to name a few) in the ROM and the QROM, but insofar as we are aware, only with a view to establishing IND-CCA security of the resulting KEMs. Only one prior work [80] studies the relationship between FO transforms and anonymity; it shows that the original FO transform enhances anonymity in the classical ROM. But this result does not tell us whether the modern FO variants used by the NIST PQC KEM candidates also enhance (or even preserve)

[1] Namely, the KEM encapsulation algorithm samples a random message from the PKE scheme's message space and then "PKE-encrypts" it; the random message is then the KEM encapsulated key and the corresponding PKE-encryption is the KEM ciphertext.

anonymity and robustness properties; notably, the results of [80] are also not in the QROM.

anonymity and robustness for the kem-dem paradigm. The first main contribution of this chapter is a modular theory of anonymity and robustness for PKE schemes built via the KEM-DEM paradigm. This extends the work of [77] to general KEMs (instead of those built directly from PKE, see Footnote 1 of this chapter). An interesting aspect that emerges is a fundamental separation between our results for implicit and explicit rejection KEMs. At a high level, KEMs that perform implicit rejection do not in general transfer anonymity and robustness to PKE schemes obtained via the KEM-DEM paradigm from the KEM component, whilst KEMs that offer explicit rejection, and that also satisfy a mild robustness property, do. Our positive result for explicit rejection KEMs relies on a relatively weak anonymity notion for KEMs which we introduce here, called wANO-CCA security. Our negative results for the implicit rejection case are proved through the construction of specific counterexamples and are surprisingly strong. For example, an implicit rejection KEM cannot be robust, but can achieve a strong form of collision freeness (called SCFR-CCA security which we define here). This is in some sense the next best thing to robustness. We show that even this property is not sufficient, by exhibiting an implicit rejection KEM that is anonymous (technically, ANO-CCA secure that we define subsequently), IND-CCA and SCFR-CCA secure, and a DEM that offers authenticated encryption (see Subsection 2.3.3) and satisfies a strong robustness property (so-called XROB security from [73]), but where the hybrid PKE scheme resulting from composing this KEM and DEM is not ANO-CCA secure.

anonymity and robustness from fo transforms. Since almost all the NIST final-round candidates are KEMs of the implicit rejection type and we have a strong negative result there, we must dig deeper if we wish to assure ourselves that anonymity and robustness will be obtained for PKE schemes built from those KEMs. This introduces the second main contribution of this chapter, wherein we analyse how the implicitly-rejecting $\mathsf{FO}^{\not\perp}$ transform of [4] (see Figure 3.2) lifts anonymity and robustness (technically, collision-freeness) properties from a starting weakly-secure PKE scheme, first to the strongly-secure KEM built by the $\mathsf{FO}^{\not\perp}$ transform, and then to the hybrid PKE scheme constructed using the KEM-DEM paradigm. The culmination of this analysis is showing that KEMs and PKE schemes built via FO-type transforms can bypass our negative result for implicit rejection KEMs.

APPLICATION TO NIST PQC CANDIDATES. In the next chapter, we will apply our above generic analysis for implicit-rejection KEMs, and the hybrid PKE schemes derived from them, to specific schemes related to the NIST PQC standardization process; these schemes employ FO-type transforms that can be seen as variants of $\mathrm{FO}^{\not\perp}$. In particular, we focus on the current NIST PQC standard *Kyber* [11], the NIST fourth-round candidate *Classic McEliece* [21], and the NIST third-round alternate candidate *FrodoKEM* [16]; it is worth mentioning that the latter two schemes are also currently recommended by the German federal agency BSI for usage in the post-quantum setting [17].

CHAPTER ORGANISATION. Section 4.1 contains some additional preliminary definitions not covered in previous chapters. Section 4.2 contains our anonymity and robustness definitions for KEMs, which can be seen as an additional contribution of this chapter. Sections 4.3 and 4.4 contain our analysis of the generic KEM-DEM composition with respect to explicit rejection and implicit rejection KEMs respectively. Section 4.5 contains our study of anonymity and robustness enhancement for the $\mathsf{FO}^{\not\perp}$ transform, and the corresponding security properties of hybrid PKE schemes built from $\mathsf{FO}^{\not\perp}$-derived KEMs.

4.1 ADDITIONAL PRELIMINARIES

In this section, we define some cryptographic primitives and security notions that are relevant in this (and the next) chapter (and that are not covered in Chapter 2).

4.1.1 *Public-Key Encryption, Revisited*

In Subsection 2.3.1, we have already seen standard notions of OW-CPA and IND-CPA/-CCA security for PKE schemes. We will now define their anonymity (formally, ANO-CPA/-CCA security) and strong/weak robustness (S/WROB security) properties, as introduced by Bellare *et al.* [12] and Abdalla *et al.* [13] respectively.

```
$\mathsf{ANO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}$
$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$ \{0, 1\}$
$(m, \mathsf{st}) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp}(\mathsf{pk}_0, \mathsf{pk}_1)$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m)$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}}(c^*, \mathsf{st})$
return $[b' = b]$
```

```
$\textsc{Dec}_a(b, c)$
if $b \notin \{0, 1\} \vee c = a$
    then return $\perp$
$m := \mathsf{Dec}(\mathsf{sk}_b, c)$
return $m$
```

```
$\mathsf{SROB\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}$ $\boxed{\mathsf{SCFR\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$
$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp}(\mathsf{pk}_0, \mathsf{pk}_1)$
$m_0 := \textsc{Dec}_\perp(0, c)$
$m_1 := \textsc{Dec}_\perp(1, c)$
return $[m_0 \neq \perp \wedge m_1 \neq \perp]$
$\boxed{\textbf{return } [m_0 = m_1 \neq \perp]}$
```

```
$\mathsf{WROB\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}$ $\boxed{\mathsf{WCFR\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$
$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$(m, b) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp}(\mathsf{pk}_0, \mathsf{pk}_1)$
$c \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m)$
$m' := \textsc{Dec}(1 - b, c)$
return $[m' \neq \perp]$
$\boxed{\textbf{return } [m = m' \neq \perp]}$
```

FIGURE 4.1: Security games for anonymity, robustness and collision freeness of PKE schemes. Here st is some state information maintained by the adversary $\mathcal{A}$.

Definition 16 (Anonymity of PKE [12])**.** *Given a PKE* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *we define the game w.r.t. its* ANO-CCA security *in Figure 4.1 and the* ANO-CCA advantage measure *for adversary* $\mathcal{A}$ *against* PKE *as*

$$\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}) = \left|\Pr[\mathsf{ANO\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}} = 1] - \frac{1}{2}\right|.$$

If we remove the adversaries' access to decryption oracles in the ANO-CCA security game, we obtain the corresponding game for ANO-CPA security*; the* ANO-CPA advantage measure *is defined in the same fashion as that of ANO-CCA.*

Definition 17 (Robustness of PKE [13])**.** *Given a PKE* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *we define the game w.r.t. its* SROB-CCA (resp. WROB-CCA) security *in Figure 4.1 and the* SROB-CCA (resp. WROB-CCA) advantage measure *for adversary* $\mathcal{A}$ *against* PKE *as*

$$\mathbf{Adv}^{\mathsf{(S/W)ROB\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}) = \Pr[\mathsf{(S/W)ROB\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}} = 1].$$

If we remove the adversaries' access to decryption oracles in the above security games, we obtain the corresponding games for CPA-versions of the robustness notions; the SROB-CPA (resp. WROB-CPA) advantage measure *is defined in the same fashion as that of SROB-CCA (resp. WROB-CCA).*

Subsequent to the formalization of robustness properties of PKE schemes in [13], Mohassel [77] introduced a relaxation of PKE robustness called *strong/weak collision freeness (S/WCFR security)*. Roughly speaking, a PKE scheme is collision-free if a ciphertext does not decrypt to the same message under two different secret keys. Mohassel used collision freeness as an intermediate notion to provide generic transformation of any collision-free PKE scheme to a (strongly) robust one. We formally define collision freeness of PKE schemes below.

Definition 18 (Collision freeness of PKE [77])**.** *Given a PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *we define the game w.r.t. its* SCFR-CCA (resp. WCFR-CCA) security *in Figure 4.1 and the* SCFR-CCA (resp. WCFR-CCA) advantage measure *for adversary* $\mathcal{A}$ *against* PKE *as*

$$\mathbf{Adv}^{\mathsf{(S/W)CFR\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}) = \Pr[\mathsf{(S/W)CFR\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}} = 1].$$

If we remove the adversaries' access to decryption oracles in the above security games, we obtain the corresponding games for CPA-versions of the collision freeness notions; the SCFR-CPA (resp. WCFR-CPA) advantage measure *is defined in the same fashion as that of SCFR-CCA (resp. WCFR-CCA).*

```
$\mathsf{FROB}^{\mathcal{A}}_{\mathsf{DEM}}$
$(c, k_0, k_1) \leftarrow \mathcal{A}$
$m_0 := \mathsf{Dec}(k_0, c)$
$m_1 := \mathsf{Dec}(k_1, c)$
$b_m := [m_0 \neq \bot \wedge m_0 \neq \bot]$
$b_k := [k_0 \neq k_1]$
return $[b_m \wedge b_k]$
```

```
$\mathsf{XROB}^{\mathcal{A}}_{\mathsf{DEM}}$
$(m_0, k_0, r_0, k_1, c_1) \leftarrow \mathcal{A}$
$c_0 := \mathsf{Enc}(k_0, m_0; r_0)$
$m_1 := \mathsf{Dec}(k_1, c_1)$
$b_m := [m_0 \neq \bot \wedge m_0 \neq \bot]$
$b_k := [k_0 \neq k_1]$
$b_c := [c_0 = c_1 \neq \bot]$
return $[b_m \wedge b_k \wedge b_c]$
```

Figure 4.2: Security games for robustness of DEMs.

4.1.2 *Data Encapsulation Mechanism, Revisited*

Note that in the previous subsection, we only considered robustness of *asymmetric key* encryption (i.e., PKE). Farshim *et al.* [73] formalized different flavors of robustness for *symmetric key* encryption. We will consider two such notions, namely *full robustness (FROB)* and *mixed robustness (XROB)*. At a high level, in the security games corresponding to these robustness notions, the adversary gets to choose the (symmetric) keys; this is in contrast to the honestly generated (asymmetric) keys in the SROB and WROB notions above. Formal definitions of FROB and XROB security follow.

Definition 19 (Robustness of DEMs [73])**.** *Given DEM* $\mathsf{DEM} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*, we define the game w.r.t. its* FROB (resp. XROB) security *in Figure 4.2 and the* FROB-CCA (resp. XROB) advantage measure *for adversary* $\mathcal{A}$ *against* DEM *as*

$$\mathbf{Adv}^{\mathsf{(F/X)ROB}}_{\mathsf{DEM}}(\mathcal{A}) = \Pr[\mathsf{(F/X)ROB}^{\mathcal{A}}_{\mathsf{DEM}} = 1].$$

4.2 anonymity and robustness of kems

As mentioned earlier, Mohassel [77] studied the anonymity and robustness of KEMs. However, all of his definitions and results apply only to the special case of KEMs that are constructed from PKE schemes in a restricted way: namely KEMs in which the encapsulation algorithm selects a random message for the PKE scheme and encrypts it using the PKE scheme's encryption algorithm. With this limitation, Mohassel provided a number of interesting results (positive and negative) concerning the anonymity and robustness of KEMs and of PKE schemes constructed from them via the KEM-DEM paradigm.

In this section, we bridge the definitional gap left by Mohassel's work by considering fully general definitions for KEM anonymity and robustness. We will then revisit his results on these properties in context of the KEM-DEM paradigm in later sections. As we shall see, how much can be recovered depends in a critical way on the KEM's behaviour with respect to rejection of invalid ciphertexts.

We first define anonymity, or more formally, ANO-CCA security of a KEM $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ via the security game between an adversary and a challenger, as described in Figure 4.3.[2]

Definition 20 ((Weak) Anonymity of KEM)**.** *Given a KEM scheme* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$, *we define the game w.r.t. its* (w)ANO-CCA security *in Figure 4.3 and the* (w)ANO-CCA advantage measure *for adversary $\mathcal{A}$ against* KEM *as*

$$\mathbf{Adv}^{\mathsf{(w)ANO\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) = \left|\Pr[\mathsf{(w)ANO\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}} = 1] - \frac{1}{2}\right|.$$

If we remove the adversaries' access to decryption oracles in the (w)ANO-CCA security game, we obtain the corresponding game for (w)ANO-CPA security; *the* (w)ANO-CPA advantage measure *is defined in the same fashion as that of (w)ANO-CCA.*

In the context of KEM-DEM paradigm for constructing PKE schemes, we will find it sufficient to work with an even weaker notion of anonymity for KEMs, that we refer to as *weak* anonymity or wANO-CPA/-CCA security. Here, the ANO- security game is modified by giving the adversary only the ciphertext c^* in response to its challenge query, instead of (c^*, k^*); see Figure 4.3. We also define the corresponding adversarial advantage measures as above.

We now shift our focus to robustness of KEMs. Specifically, we define weak robustness (WROB) and strong robustness (SROB) security notions for general KEMs; the corresponding security games are described in Fig. 4.3.[3]

[2] Note that the security game differs from so-called "AI-CPA/-CCA" games (roughly speaking, they are a hybrid of ANO- and IND- security games: AI = ANO + IND) defined for *general encryption schemes* in [13], where in the latter, an adversary can have access to multiple public keys (and some corresponding secret keys which will not result in a trivial win for the adversary). Since we are only considering PKE schemes and KEMs in this chapter, it is not hard to show that the two security notions are equivalent up to a factor depending on the number of secret key queries an adversary could make (as already discussed in [13]).

[3] The security game for WROB has a subtle difference from the corresponding WROB game defined for general encryption schemes in [13] (in addition to the fact that, in the latter game,

```
$\mathsf{ANO\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}$ $\boxed{\mathsf{wANO\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}}$
$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$ \{0, 1\}$
$(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}_0, \mathsf{pk}_1, c^*, k^*)$
$\boxed{b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}_0, \mathsf{pk}_1, c^*)}$
return $[b = b']$
```

```
$\textsc{Decaps}_a(b, c)$
if $b \notin \{0, 1\} \vee c = a$
    then return $\perp$
$k := \mathsf{Decap}(\mathsf{sk}_b, c)$
return $k$
```

```
$\mathsf{SROB\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}$ $\boxed{\mathsf{SCFR\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}}$
$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$c \leftarrow \mathcal{A}^{\textsc{Decaps}_\perp}(\mathsf{pk}_0, \mathsf{pk}_1)$
$k_0 := \textsc{Decaps}_\perp(0, c)$
$k_1 := \textsc{Decaps}_\perp(1, c)$
return $[k_0 \neq \perp \wedge k_1 \neq \perp]$
$\boxed{\textbf{return } [k_0 = k_1 \neq \perp]}$
```

```
$\mathsf{WROB\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}$ $\boxed{\mathsf{WCFR\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}}}$
$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$b \leftarrow \mathcal{A}^{\textsc{Decaps}_\perp}(\mathsf{pk}_0, \mathsf{pk}_1)$
$(c, k_b) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$
$k_{1-b} := \textsc{Decaps}(1 - b, c)$
return $[k_{1-b} \neq \perp]$
$\boxed{\textbf{return } [k_b = k_{1-b} \neq \perp]}$
```

FIGURE 4.3: Security games for anonymity, robustness and collision freeness of KEMs.

Definition 21 (Robustness of KEM)**.** *Given KEM* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$*, we define the game w.r.t. its* SROB-CCA (resp. WROB-CCA) security *in Figure 4.3 and the* SROB-CCA (resp. WROB-CCA) advantage measure *for adversary $\mathcal{A}$ against* KEM *as*

$$\mathbf{Adv}^{\mathsf{(S/W)ROB\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) = \Pr[\mathsf{(S/W)ROB\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM}} = 1].$$

If we remove the adversaries' access to decryption oracles in the above security games, we obtain the corresponding games for CPA-versions of the robustness notions; the SROB-CPA (resp. WROB-CPA) advantage measure *is defined in the same fashion as that of SROB-CCA (resp. WROB-CCA).*

Note that these robustness definitions mainly apply for *explicitly rejecting* KEMs that output a special symbol $\perp$ on decapsulation errors (see e.g., KEMs derived from the $\mathsf{FO}^{\perp}$ transform in Fig. 3.2). KEMs that offer only *implicit rejection* – i.e., which never output $\perp$ on decapsulation (e.g., $\mathsf{FO}^{\not\perp}$-derived KEMs in Fig. 3.2) – cannot satisfy even the WROB-CPA notion.

In the following sections, we will revisit the KEM-DEM paradigm wherein we focus on anonymity and robustness of the hybrid PKE scheme when starting with corresponding anonymous and robust properties of the underlying KEM (and DEM). We will first consider the case of explicitly rejecting KEMs in Section 4.3 before turning our attention to (non-robust) implicitly rejecting KEMs in Section 4.4.

4.3 GENERIC KEM-DEM COMPOSITION FOR EXPLICIT REJECTION KEMS

With the above anonymity and robustness notions in hand, it is straightforward to extend Mohassel's result (specifically, [77, Claim 3.3]) concerning anonymity preservation in the KEM-DEM composition from the specific case of KEMs constructed directly from PKEs to fully general (explicitly rejecting) KEMs, with a non-zero decapsulation error probability; in fact, we can also show the robustness of hybrid PKE schemes constructed from robust KEMs via the KEM-DEM paradigm. More formally, we have the following:

an adversary can have access to multiple public keys). The difference is that in our notion, an adversary outputs a bit b that determines which of the two public keys $(\mathsf{pk}_0, \mathsf{pk}_1)$ will be used for encapsulation. This is required because the weak robustness notion is inherently *asymmetric* w.r.t. the two challenge public keys, since one key is used for encapsulation (resp. encryption in case of PKE schemes) and the other for decapsulation (resp. decryption in case of PKE schemes).

Theorem 3. *Let* $\mathsf{PKE}^{\mathrm{hy}} = (\mathsf{KGen}^{\mathrm{hy}}, \mathsf{Enc}^{\mathrm{hy}}, \mathsf{Dec}^{\mathrm{hy}})$ *be a hybrid PKE scheme obtained by composing a KEM* $\mathsf{KEM} = (\mathsf{KGen}^{\mathrm{kem}}, \mathsf{Encap}, \mathsf{Decap})$ *with a DEM* $\mathsf{DEM} = (\mathsf{KGen}^{\mathrm{dem}}, \mathsf{Enc}, \mathsf{Dec})$*. If* KEM *is δ-correct, then:*

1. *For any* ANO-CCA *adversary* $\mathcal{A}_{\mathrm{hy}}$ *against* $\mathsf{PKE}^{\mathrm{hy}}$*, there exist* wANO-CCA *adversary* $\mathcal{A}_{\mathrm{kem}}$*,* IND-CCA *adversary* $\overline{\mathcal{A}}_{\mathrm{kem}}$ *and* WROB-CPA *adversary* $\hat{\mathcal{A}}_{\mathrm{kem}}$ *against* KEM*, and* INT-CTXT *adversary* $\mathcal{A}_{\mathrm{dem}}$ *against* DEM *such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{wANO\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}_{\mathrm{kem}}) + 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\overline{\mathcal{A}}_{\mathrm{kem}}) \\ + \mathbf{Adv}^{\mathsf{WROB\text{-}CPA}}_{\mathsf{KEM}}(\hat{\mathcal{A}}_{\mathrm{kem}}) + \mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) + \delta\,.\end{aligned}$$

   The running times of $\mathcal{A}_{\mathrm{kem}}$*,* $\overline{\mathcal{A}}_{\mathrm{kem}}$ *and* $\mathcal{A}_{\mathrm{dem}}$ *are the same as that of* $\mathcal{A}_{\mathrm{hy}}$*. The running time of* $\hat{\mathcal{A}}_{\mathrm{kem}}$ *is independent (and less than that) of the running time of* $\mathcal{A}_{\mathrm{hy}}$*.*

2. *For any* WROB-ATK *(resp.* SROB-ATK*) adversary* $\mathcal{A}_{\mathrm{hy}}$ *against* $\mathsf{PKE}^{\mathrm{hy}}$*, there exists* WROB-ATK *(resp.* SROB-ATK*) adversary* $\mathcal{A}_{\mathrm{kem}}$ *against* KEM *such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{WROB\text{-}ATK}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{WROB\text{-}ATK}}_{\mathsf{KEM}}(\mathcal{A}_{\mathrm{kem}})\,, \\ \mathbf{Adv}^{\mathsf{SROB\text{-}ATK}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{SROB\text{-}ATK}}_{\mathsf{KEM}}(\mathcal{A}_{\mathrm{kem}})\,,\end{aligned}$$

   where ATK $\in \{\text{CPA}, \text{CCA}\}$ *and the running time of* $\mathcal{A}_{\mathrm{kem}}$ *is that of* $\mathcal{A}_{\mathrm{hy}}$*.*

Proof. (of Theorem 3.1)

Let $\mathcal{A}_{\mathrm{hy}}$ be an adversary in the ANO-CCA game for $\mathsf{PKE}^{\mathrm{hy}}$. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_4$ described in Figure 4.4.

Game G_0**:** The game G_0 is exactly the ANO-CCA game for $\mathsf{PKE}^{\mathrm{hy}}$. Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}})$$

Game G_1**:** In game G_1, we first make some "cosmetic" changes. Namely, the pair (c_0^*, k^*) is generated by running $\mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b *before* the adversary $\mathcal{A}_{\mathrm{hy}}$ gets to choose a message m. This change does not affect $\mathcal{A}_{\mathrm{hy}}$'s view in any way.

Next, we modify the oracle $\textsc{Dec}^{\mathrm{hy}}_{a}(b, \cdot)$ (with $a \in \{\perp, c^*\}$) such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$ (and $c_1 \neq c_1^*$), then the oracle uses k^* to decrypt c_1, instead of first decapsulating c_0^* to recover a session

```
Games $\mathsf{G}_0$ - $\mathsf{G}_4$
$(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}^{\mathrm{kem}}$
$b \leftarrow\!\!\$ \{0,1\}$   // $\mathsf{G}_1 - \mathsf{G}_4$
$(c_0^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$   // $\mathsf{G}_1 - \mathsf{G}_4$
$\hat{k} \leftarrow\!\!\$ \mathcal{K}$   // $\mathsf{G}_3 - \mathsf{G}_4$
$(m, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{\mathrm{DEC}_{\perp}^{\mathrm{hy}}}(\mathsf{pk}_0, \mathsf{pk}_1)$
$b \leftarrow\!\!\$ \{0,1\}$   // $\mathsf{G}_0$
$(c_0^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$   // $\mathsf{G}_0$
$c_1^* \leftarrow \mathsf{Enc}(k^*, m)$   // $\mathsf{G}_0 - \mathsf{G}_2$
$c_1^* \leftarrow \mathsf{Enc}(\hat{k}, m)$   // $\mathsf{G}_3 - \mathsf{G}_4$
$c^* = (c_0^*, c_1^*)$
$b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{\mathrm{DEC}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
return $[b' = b]$
```

```
$\mathrm{DEC}_a^{\mathrm{hy}}(b, c)$   // $c \neq a$
Parse $c = (c_0, c_1)$
if $c_0 = c_0^*$   // $\mathsf{G}_1 - \mathsf{G}_4$
    $k' \leftarrow k^*$   // $\mathsf{G}_1 - \mathsf{G}_2$
    $k' \leftarrow \hat{k}$   // $\mathsf{G}_3$
    return $\perp$   // $\mathsf{G}_4$
else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_0, c_0)$
$m' \leftarrow \mathsf{Dec}(k', c_1)$
return $m'$
```

```
$\mathrm{DEC}_a^{\mathrm{hy}}(1 - b, c)$   // $c \neq a$
Parse $c = (c_0, c_1)$
if $c_0 = c_0^*$   // $\mathsf{G}_2 - \mathsf{G}_4$
    return $\perp$   // $\mathsf{G}_2 - \mathsf{G}_4$
else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_1, c_0)$
$m' \leftarrow \mathsf{Dec}(k', c_1)$
return $m'$
```

FIGURE 4.4: Games G_0 – G_4 for the proof of Theorem 3.

key k'. It is not hard to see that the games G_0 and G_1 are equivalent unless there is a decapsulation error w.r.t. KEM. Therefore, we have

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \delta$$

Game G_2: In game G_2, we modify the oracle $\textsc{Dec}_a^{\mathrm{hy}}(1-b,\cdot)$ (with $a \in \{\perp, c^*\}$) such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$, then the oracle returns $\perp$. Again it is not hard to see that the games G_1 and G_2 are equivalent unless the following event occurs: $\mathsf{Decap}(\mathsf{sk}_{1-b}, c_0^*) = k' \neq \perp$ (and $\mathsf{Dec}(k', c_1) \neq \perp$) where $\mathsf{Encap}(\mathsf{pk}_b) = (c_0^*, k^*)$. And we can bound the probability of this event occurring by the advantage of an adversary $\hat{\mathcal{A}}_{\mathsf{kem}}$ in the WROB-CPA game of KEM. The adversary $\hat{\mathcal{A}}_{\mathsf{kem}}$, upon receiving public keys pk_0 and pk_1, simply samples a bit b uniformly at random, i.e., $b \leftarrow\$ \{0,1\}$, and returns the bit to the WROB-CPA challenger. Hence,

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{WROB\text{-}CPA}}(\hat{\mathcal{A}}_{\mathsf{kem}})$$

Game G_3: In game G_3, we compute c_1^* in the setup as "$c_1^* = \mathsf{Enc}(\hat{k}, m)$", instead of "$c_1^* = \mathsf{Enc}(k^*, m)$" as in G_2, for a uniformly random key $\hat{k}$ (i.e., $\hat{k} \leftarrow\$ \mathcal{K}$, where $\mathcal{K}$ is the encapsulated key space of KEM) that is independent of k^*. We make the appropriate modification in the $\textsc{Dec}_a^{\mathrm{hy}}(b,\cdot)$ oracle as well, i.e., if the decryption query is (c_0, c_1) where $c_0 = c_0^*$, then the oracle uses $\hat{k}$ (instead of k^*) to decrypt c_1.

We now show that the difference in $\mathcal{A}_{\mathrm{hy}}$'s success probabilities in games G_2 and G_3 can be bounded by the advantage of an adversary $\overline{\mathcal{A}}_{\mathsf{kem}}$ in the IND-CCA game of KEM. Upon receiving the input (pk, c^*, k) from its IND-CCA challenger, where $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$ and $k \leftarrow\$ \{k^*, \hat{k}\}$ for a uniformly random key $\hat{k}$ that is independent of k^*, $\overline{\mathcal{A}}_{\mathsf{kem}}$ proceeds as described in Figure 4.5. Note that if k is a "real" (respectively, "random") key, i.e., $k = k^*$ (resp., $k = \hat{k}$), then $\overline{\mathcal{A}}_{\mathsf{kem}}$ perfectly simulates game G_2 (resp., G_3) towards $\mathcal{A}_{\mathrm{hy}}$ (also note that, to answer $\mathcal{A}_{\mathrm{hy}}$'s decryption queries, $\overline{\mathcal{A}}_{\mathsf{kem}}$ never has to make the *forbidden* query $c^* (= c_0^*)$ to its decapsulation oracle $\textsc{Decaps}_{c^*} (= \mathsf{Decap}(\mathsf{sk}_b, \cdot))$. Therefore, we have

$$\begin{aligned} |\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| &= |\Pr[1 \leftarrow \overline{\mathcal{A}}_{\mathsf{kem}}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k) \mid k = \hat{k}] \\ &\quad - \Pr[1 \leftarrow \overline{\mathcal{A}}_{\mathsf{kem}}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k) \mid k = k^*]| \\ &\leq 2\mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\overline{\mathcal{A}}_{\mathsf{kem}}) \end{aligned}$$

Game G_4: In game G_4, we modify the oracle $\textsc{Dec}_a^{\mathrm{hy}}(b,\cdot)$ such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$, then the oracle returns $\perp$.

```
$\overline{\mathcal{A}}_{\mathsf{kem}}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k)$
  $b \leftarrow\!\!\$ \{0,1\}$
  $\mathsf{pk}_b = \mathsf{pk}$
  $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}^{\mathsf{kem}}$
  $c_0^* = c^*$
  $(m, \mathsf{st}) \leftarrow \mathcal{A}_{\mathsf{hy}}^{\textsc{Dec}_{\perp}^{\mathsf{hy}}}(\mathsf{pk}_0, \mathsf{pk}_1)$
  $c_1^* \leftarrow \mathsf{Enc}(k, m)$
  $c^* = (c_0^*, c_1^*)$
  $b' \leftarrow \mathcal{A}_{\mathsf{hy}}^{\textsc{Dec}_{c^*}^{\mathsf{hy}}}(c^*, \mathsf{st})$
  return $[b' = b]$
```

```
$\textsc{Dec}_a^{\mathsf{hy}}(b, c)$    // $c \neq a$
  Parse $c = (c_0, c_1)$
  if $c_0 = c_0^*$ then $k' \leftarrow k$
  else $k' \leftarrow \textsc{Decaps}_{c^*}(c_0)$
  $m' \leftarrow \mathsf{Dec}(k', c_1)$
  return $m'$
```

```
$\textsc{Dec}_a^{\mathsf{hy}}(1-b, c)$    // $c \neq a$
  Parse $c = (c_0, c_1)$
  if $c_0 = c_0^*$ then return $\perp$
  else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_{1-b}, c_0)$
  $m' \leftarrow \mathsf{Dec}(k', c_1)$
  return $m'$
```

Figure 4.5: IND-CCA adversary $\overline{\mathcal{A}}_{\mathsf{kem}}^{\textsc{Decaps}_{c^*}}$ for the proof of Theorem 3. Here the $\textsc{Decaps}_{c^*}$ oracle corresponds to the Decap algorithm of KEM as in the IND-CCA security game for KEMs; see Fig. 2.2.

It is not hard to see that the games G_3 and G_4 are equivalent unless the following event occurs: $\mathcal{A}_{\mathsf{hy}}$ makes a decryption query (c_0^*, c_1) to the oracle $\textsc{Dec}_a^{\mathsf{hy}}(b, \cdot)$ such that $\mathsf{Dec}(\hat{k}, c_1) \neq \perp$, for a uniformly random key $\hat{k}$. And we can bound the probability of this event occurring by the advantage of an adversary $\mathcal{A}_{\mathsf{dem}}$ in the INT-CTXT game of DEM. In the INT-CTXT game, we are implicitly defining $\hat{k}$ to be the random secret key chosen by the challenger. The adversary $\mathcal{A}_{\mathsf{dem}}$ proceeds as described in Figure 4.6. Note that if the aforementioned event occurs, then $\mathcal{A}_{\mathsf{dem}}$ wins its corresponding game; also note that, $\mathcal{A}_{\mathsf{dem}}$ only makes a single encryption query to the one-time AE-secure DEM, namely "$c_1^* = \textsc{Enc}(m)$" (= $\mathsf{Enc}(\hat{k}, m)$; see Line 5 in "$\mathcal{A}_{\mathsf{dem}}^{\textsc{Enc},\textsc{Dec}}$", Fig. 4.6), and then it never makes the forbidden query c_1^* to its decryption oracle $\textsc{Dec}$ (= $\mathsf{Dec}(\hat{k}, \cdot)$). Hence, we have

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Adv}_{\mathsf{DEM}}^{\mathsf{INT\text{-}CTXT}}(\mathcal{A}_{\mathsf{dem}})$$

Finally, we show that $\mathcal{A}_{\mathsf{hy}}$'s success probability in game G_4 can be bounded by the advantage of an adversary $\mathcal{A}_{\mathsf{kem}}$ in the wANO-CCA game of KEM. Upon receiving public keys pk_0 and pk_1 along with the ciphertext c^*, where $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b chosen by the

```
$\mathcal{A}_{\text{dem}}^{\textsc{Enc},\textsc{Dec}}$
    $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}^{\text{kem}}$
    $b \leftarrow\!\$\ \{0,1\}$
    $(c_0^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$
    $(m, \mathsf{st}) \leftarrow \mathcal{A}_{\text{hy}}^{\textsc{Dec}_{\perp}^{\text{hy}}}(\mathsf{pk}_0, \mathsf{pk}_1)$
    $c_1^* \leftarrow \textsc{Enc}(m)$
    $c^* = (c_0^*, c_1^*)$
    $b' \leftarrow \mathcal{A}_{\text{hy}}^{\textsc{Dec}_{c^*}^{\text{hy}}}(c^*, \mathsf{st})$
    return $\perp$
```

```
$\textsc{Dec}_a^{\text{hy}}(b, c)$    // $c \neq a$
    Parse $c = (c_0, c_1)$
    if $c_0 = c_0^*$ then
        query $\textsc{Dec}(c_1)$
        return $\perp$
    else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_0, c_0)$
    $m' \leftarrow \mathsf{Dec}(k', c_1)$
    return $m'$
```

```
$\textsc{Dec}_a^{\text{hy}}(1-b, c)$    // $c \neq a$
    Parse $c = (c_0, c_1)$
    if $c_0 = c_0^*$ then return $\perp$
    else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_1, c_0)$
    $m' \leftarrow \mathsf{Dec}(k', c_1)$
    return $m'$
```

FIGURE 4.6: INT-CTXT adversary $\mathcal{A}_{\text{dem}}^{\textsc{Enc},\textsc{Dec}}$ for the proof of Theorem 3.

challenger, the adversary $\mathcal{A}_{\text{kem}}$ proceeds as described in Figure 4.7. Observe that $\mathcal{A}_{\text{kem}}$ perfectly simulates the game G_4 towards $\mathcal{A}_{\text{hy}}$ (also note that, to answer $\mathcal{A}_{\text{hy}}$'s decryption queries, $\mathcal{A}_{\text{kem}}$ never has to make the *forbidden* query $c^*(= c_0^*)$ to its decapsulation oracle $\textsc{Decaps}_{c^*}$. Therefore, we have $|\Pr[\mathsf{G}_4 = 1] - 1/2| = \mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{wANO\text{-}CCA}}(\mathcal{A}_{\text{kem}})$.

Collecting all of the above bounds, we finally arrive at

$$\begin{aligned}\mathbf{Adv}_{\mathsf{PKE}^{\text{hy}}}^{\mathsf{ANO\text{-}CCA}}(\mathcal{A}_{\text{hy}}) \leq{} & \mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{wANO\text{-}CCA}}(\mathcal{A}_{\text{kem}}) + 2\mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\overline{\mathcal{A}}_{\text{kem}}) \\ & + \mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{WROB\text{-}CPA}}(\hat{\mathcal{A}}_{\text{kem}}) + \mathbf{Adv}_{\mathsf{DEM}}^{\mathsf{INT\text{-}CTXT}}(\mathcal{A}_{\text{dem}}) + \delta\,.\end{aligned}$$

□

Proof. (of Theorem 3.2)

Let $\mathcal{A}_{\text{hy}}$ be an adversary in the WROB-ATK game for PKE^{hy}. Upon receiving two (honestly-generated) public keys pk_0 and pk_1, $\mathcal{A}_{\text{hy}}$ wins the game if it returns a message and a bit, namely (m, b), such that $\mathsf{Dec}^{\text{hy}}(\mathsf{sk}_{1-b}, c) \neq \perp$ where $c(= (c_0, c_1)) \leftarrow \mathsf{Enc}^{\text{hy}}(\mathsf{pk}_b, m)$. Let $(c_0, k_b) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$ and $\mathsf{Decap}(\mathsf{sk}_{1-b}, c_0) = k_{1-b}$. It is easy to see that $k_{1-b} \neq \perp$, since $\mathsf{Dec}^{\text{hy}}(\mathsf{sk}_{1-b}, c) \neq \perp$ implies $\mathsf{Dec}(k_{1-b}, c_1) \neq \perp$. The probability of

```
$\mathcal{A}_{\text{kem}}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}_0, \mathsf{pk}_1, c^*)$
    $\hat{k} \leftarrow\$ \mathcal{K}$
    $c_0^* = c^*$
    $(m, \mathsf{st}) \leftarrow \mathcal{A}_{\text{hy}}^{\textsc{Dec}_{\perp}^{\text{hy}}}(\mathsf{pk}_0, \mathsf{pk}_1)$
    $c_1^* \leftarrow \mathsf{Enc}(\hat{k}, m)$
    $c^* = (c_0^*, c_1^*)$
    $b' \leftarrow \mathcal{A}_{\text{hy}}^{\textsc{Dec}_{c^*}^{\text{hy}}}(c^*, \mathsf{st})$
    return $b'$
```

```
$\textsc{Dec}_a^{\text{hy}}(0, c)$    // $c \neq a$
    Parse $c = (c_0, c_1)$
    if $c_0 = c_0^*$ return $\perp$
    else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_0, c_0)$
    $m' \leftarrow \mathsf{Dec}(k', c_1)$
    return $m'$

$\textsc{Dec}_a^{\text{hy}}(1, c)$    // $c \neq a$
    Parse $c = (c_0, c_1)$
    if $c_0 = c_0^*$ return $\perp$
    else $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_1, c_0)$
    $m' \leftarrow \mathsf{Dec}(k', c_1)$
    return $m'$
```

Figure 4.7: wANO-CCA adversary $\mathcal{A}_{\text{kem}}^{\textsc{Decaps}_{c^*}}$ for the proof of Theorem 3. Here the $\textsc{Decaps}_{c^*}$ oracle corresponds to the Decap algorithm of KEM as in the wANO-CCA security game for KEMs; see Fig. 4.3.

$\mathcal{A}_{\text{hy}}$ winning the game can then be bounded by the advantage of an adversary $\mathcal{A}_{\text{kem}}$ in the WROB-ATK game for KEM. Upon receiving two public keys pk_0 and pk_1 from its WROB-ATK challenger, $\mathcal{A}_{\text{kem}}$ forwards the keys to $\mathcal{A}_{\text{hy}}$ and simulates the WROB-ATK game w.r.t. PKE^{hy} (note that if ATK = CCA, then $\mathcal{A}_{\text{kem}}$ can simulate the $\textsc{Dec}_a^{\text{hy}}$ oracles since it has access to the $\textsc{Decaps}_a$ oracles in its WROB-CCA game). Once $\mathcal{A}_{\text{hy}}$ finally submits the pair (m, b), $\mathcal{A}_{\text{kem}}$ forwards the bit b to the WROB-ATK challenger. Note that a win for $\mathcal{A}_{\text{hy}}$ implies a win for $\mathcal{A}_{\text{kem}}$.

Similarly, let $\mathcal{A}_{\text{hy}}$ be an adversary in the SROB-ATK game for PKE^{hy}. Upon receiving two (honestly-generated) public keys pk_0 and pk_1, $\mathcal{A}_{\text{hy}}$ wins the game if it returns a ciphertext c $(= (c_0, c_1))$ such that $\mathsf{Dec}^{\text{hy}}(\mathsf{sk}_0, c) \neq \perp$ and $\mathsf{Dec}^{\text{hy}}(\mathsf{sk}_1, c) \neq \perp$. Let $\mathsf{Decap}(\mathsf{sk}_0, c_0) = k_0$ and $\mathsf{Decap}(\mathsf{sk}_1, c_0) = k_1$. It is again easy to see that $k_0 \neq \perp$ and $k_1 \neq \perp$ since we have $\mathsf{Dec}(k_0, c_1) \neq \perp$ and $\mathsf{Dec}(k_1, c_1) \neq \perp$. Hence we can bound the winning probability of $\mathcal{A}_{\text{hy}}$ by the advantage of an adversary $\mathcal{A}_{\text{kem}}$ in the SROB-ATK game for KEM. Upon receiving two public keys pk_0 and pk_1 from its SROB-ATK challenger, $\mathcal{A}_{\text{kem}}$ forwards the keys to $\mathcal{A}_{\text{hy}}$ and simulates the SROB-ATK game w.r.t. PKE^{hy} (note that if ATK = CCA, then $\mathcal{A}_{\text{kem}}$ can simulate the

$\textsc{Dec}_a^{\text{hy}}$ oracles since it has access to the $\textsc{Decaps}_a$ oracles in its SROB-CCA game). Once $\mathcal{A}_{\text{hy}}$ submits the final ciphertext $c = (c_0, c_1)$, $\mathcal{A}_{\text{kem}}$ forwards c_0 to the SROB-ATK challenger. Again, a win for $\mathcal{A}_{\text{hy}}$ implies a win for $\mathcal{A}_{\text{kem}}$. □

4.4 GENERIC KEM-DEM COMPOSITION FOR IMPLICIT REJECTION KEMS

Note that Theorem 3 in the previous section is only meaningful for KEMs with explicit rejection, since for implicitly rejecting KEMs, the advantage term $\mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{WROB\text{-}CPA}}(\cdot)$ in the above security bounds can be significantly large (in fact, the advantage will be 1 for reasonable adversaries). In this section, we will now consider the generic KEM-DEM composition – in the context of anonymity and robustness – when the underyling KEM can only do implicit rejection.

4.4.1 *Robustness*

We first consider what can be said about robustness for hybrid PKE schemes built from KEMs offering implicit rejection. We begin with a relaxed notion of robustness, namely *collision freeness* (as introduced for the specific case of KEMs obtained from PKEs in [77]). Informally, a scheme is said to be collision-free if a ciphertext always decrypts to two *different* messages under two different secret keys. We consider two variants, weak (WCFR) and strong collision freeness (SCFR). We formally define both notions via the security games described in Figure 4.3; note that the games have different finalisation steps compared to that corresponding to the robustness (WROB, SROB) notions.

Definition 22 (Collision freeness of KEMs). *Given KEM scheme* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$, *we define the game w.r.t. its* SCFR-CCA (resp. WCFR-CCA) security *in Figure 4.3 and the* SCFR-CCA (resp. WCFR-CCA) advantage measure *for adversary* $\mathcal{A}$ *against* KEM *as*

$$\mathbf{Adv}_{\mathsf{KEM}}^{\mathsf{(S/W)CFR\text{-}CCA}}(\mathcal{A}) = \Pr[\mathsf{(S/W)CFR\text{-}CCA}_{\mathsf{KEM}}^{\mathcal{A}} = 1].$$

If we remove the adversaries' access to decryption oracles in the above security games, we obtain the corresponding games for CPA-versions of the collision freeness notions; the SCFR-CPA (resp. WCFR-CPA) advantage measure *is defined in the same fashion as that of SCFR-CCA (resp. WCFR-CCA).*

Now suppose we have a KEM that is SCFR-CCA (resp. WCFR-CCA) secure and a DEM that is FROB (resp. XROB) secure. (Recall that FROB and XROB are robustness notions for symmetric encryption schemes introduced in [73] and defined in Figure 4.2.) Then we can show that the hybrid PKE scheme obtained by composing these KEM and DEM schemes is SROB-CCA (resp. WROB-CCA) secure. More formally,

Theorem 4. *Let* $\mathsf{PKE}^{\mathsf{hy}} = (\mathsf{KGen}^{\mathsf{hy}}, \mathsf{Enc}^{\mathsf{hy}}, \mathsf{Dec}^{\mathsf{hy}})$ *be a hybrid encryption scheme obtained by composing a KEM* $\mathsf{KEM} = (\mathsf{KGen}^{\mathsf{kem}}, \mathsf{Encap}, \mathsf{Decap})$ *with a DEM* $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}, \mathsf{Dec})$*. Then for any* SROB-CCA *(resp.* WROB-CCA*) adversary* $\mathcal{A}$ *against* $\mathsf{PKE}^{\mathsf{hy}}$*, there exist* SCFR-CCA *(resp.* WCFR-CCA*) adversary* $\mathcal{B}$ *against* KEM *and* FROB *(resp.* XROB*) adversary* $\mathcal{C}$ *against* DEM *such that*

$$\mathbf{Adv}^{\mathsf{SROB\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}_{\mathrm{kem}}) + \mathbf{Adv}^{\mathsf{FROB}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}),$$
$$\mathbf{Adv}^{\mathsf{WROB\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{WCFR\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}_{\mathrm{kem}}) + \mathbf{Adv}^{\mathsf{XROB}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}),$$

where the running times of $\mathcal{A}_{\mathrm{kem}}$ *and* $\mathcal{A}_{\mathrm{dem}}$ *are the same as that of* $\mathcal{A}_{\mathrm{hy}}$*.*

Proof. We only focus on SROB-CCA security in the following. The proof for WROB-CCA security follows similarly via straightforward reductions. Now let $\mathcal{A}_{\mathrm{hy}}$ be an adversary in the SROB-CCA game for $\mathsf{PKE}^{\mathsf{hy}}$. Upon receiving two (honestly-generated) public keys pk_0 and pk_1, $\mathcal{A}_{\mathrm{hy}}$ wins the game if it returns a ciphertext c $(= (c_0, c_1))$ such that $\mathsf{Dec}^{\mathsf{hy}}(\mathsf{sk}_0, c) \neq \perp$ and $\mathsf{Dec}^{\mathsf{hy}}(\mathsf{sk}_1, c) \neq \perp$. Let $\mathsf{Decap}(\mathsf{sk}_0, c_0) = k_0$ and $\mathsf{Decap}(\mathsf{sk}_1, c_0) = k_1$. It is easy to see that $k_0 \neq \perp$ and $k_1 \neq \perp$. Now we consider two (disjoint) sub-events w.r.t. this winning event:

- $\underline{k_0 = k_1}$. It is easy to see that the probability of this winning sub-event can be bounded by the advantage of an adversary $\mathcal{A}_{\mathrm{kem}}$ in the SCFR-CCA game for KEM. Upon receiving two public keys pk_0 and pk_1 from its SCFR-CCA challenger, $\mathcal{A}_{\mathrm{kem}}$ forwards the keys to $\mathcal{A}_{\mathrm{hy}}$ and simulates the SROB-CCA game w.r.t. $\mathsf{PKE}^{\mathsf{hy}}$ (note that $\mathcal{A}_{\mathrm{kem}}$ can simulate the $\textsc{Dec}_a^{\mathrm{hy}}$ oracles since it has access to the $\textsc{Decaps}_a$ oracles in its SCFR-CCA game). Once $\mathcal{A}_{\mathrm{hy}}$ submits the final ciphertext $c = (c_0, c_1)$, $\mathcal{A}_{\mathrm{kem}}$ forwards c_0 to the SCFR-CCA challenger. Note that $k_0 = k_1$ implies a win for $\mathcal{A}_{\mathrm{kem}}$.

- $\underline{k_0 \neq k_1}$. The probability of this winning sub-event can be bounded by the advantage of an adversary $\mathcal{A}_{\mathrm{dem}}$ in the FROB game for DEM. $\mathcal{A}_{\mathrm{dem}}$ generates two key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)$ honestly using $\mathsf{KGen}^{\mathsf{kem}}$ and forwards $(\mathsf{pk}_0, \mathsf{pk}_1)$ to $\mathcal{A}_{\mathrm{hy}}$. $\mathcal{A}_{\mathrm{dem}}$ then simulates the SROB-CCA game w.r.t. $\mathsf{PKE}^{\mathrm{hy}}$ towards $\mathcal{A}_{\mathrm{hy}}$ (again note that $\mathcal{A}_{\mathrm{dem}}$ can simulate the $\textsc{Dec}_a^{\mathrm{hy}}$ oracles since it has access to the corresponding secret keys sk_0, sk_1). Once $\mathcal{A}_{\mathrm{hy}}$ submits the final ciphertext $c = (c_0, c_1)$, $\mathcal{A}_{\mathrm{dem}}$ first computes k_0, k_1 as above and forwards (c_1, k_0, k_1) to the FROB challenger. Note that $\mathcal{A}_{\mathrm{hy}}$ winning implies $\mathsf{Dec}^{\mathrm{hy}}(\mathsf{sk}_i, c) \neq \perp$ which in turn implies $\mathsf{Dec}(k_i, c_1) \neq \perp$. Therefore, the (sub-)event that $k_0 \neq k_1$ implies a win for $\mathcal{A}_{\mathrm{dem}}$.

We conclude the proof by noting that we can do a similar case-distinction as above to argue about WROB-CCA security as well. □

Note that Farshim et al. [73] provide efficient constructions of FROB-secure and XROB-secure DEMs, meaning that the requirements for the above theorem can be easily met. At the same time, they showed that a DEM that achieves the standard AE notion of security is also inherently robust, albeit w.r.t. some weaker notions compared to FROB. Namely, such AE-secure DEMs were shown to satisfy the so-called *semi-full robustness* (SFROB) notion in [73]. The SFROB notion of robustness for DEMs is a (potentially) weaker variant of FROB where, in the corresponding security game, the adversary does not get to choose any keys. Instead, two keys are honestly generated and the adversary is given oracle access to encryption and decryption algorithms under both keys. The adversary is also given access to one of the keys, and the game is won (similar to that of FROB) if the adversary returns a ciphertext that decrypts correctly under both honestly generated keys.

The following theorem shows that a DEM that is only AE-secure – and that lacks the stronger robustness properties from [73] – is incapable of *generically* transforming strongly collision-free implicit rejection KEMs to strongly robust hybrid PKE schemes.

Theorem 5. *Suppose there exists a KEM that is simultaneously* SCFR-CCA, IND-CCA *and* ANO-CCA *secure. Suppose that there exists a* SUF-CMA*-secure MAC scheme and an* IND-CPA *secure symmetric encryption scheme (such schemes can be built assuming only the existence of one-way functions). Suppose also that collision-resistant hash functions exist. Then there exists an implicit rejection KEM that is* SCFR-CCA, IND-CCA *and* ANO-CCA *secure and a DEM that is* AE-secure*, such that the hybrid PKE scheme obtained from their composition is not* SROB-CCA *secure.*

Proof. We focus on the "Encrypt-then-MAC" (EtM) construction of a DEM (see Section 2.3.4). Namely, let $\mathsf{MAC} = (\mathsf{KGen}^{\mathrm{mac}}, \mathsf{Tag}, \mathsf{Vf})$ be an SUF-CMA

MAC with key space $\mathcal{K}$. We construct $\overline{\mathsf{MAC}} = (\mathsf{KGen}^{\mathrm{mac}}, \overline{\mathsf{Tag}}, \overline{\mathsf{Vf}})$ where the only difference from MAC is that a fixed special key $\bar{k}$ is chosen uniformly at random from $\mathcal{K}$ such that the verification of *any* tag under $\bar{k}$ verifies successfully, i.e., $\overline{\mathsf{Vf}}(\bar{k}, \cdot) = 1$. Note that $\overline{\mathsf{MAC}}$ is also SUF-CMA secure because the probability of sampling $\bar{k}$ uniformly at random from $\mathcal{K}$ can be considered to be negligible (here we are assuming that $\mathsf{KGen}^{\mathrm{mac}}$ outputs a uniformly random key from $\mathcal{K}$ which is typically the case in practice). So by composing $\overline{\mathsf{MAC}}$ with an IND-CPA secure symmetric encryption scheme that *never* rejects invalid ciphertexts (i.e., never outputs $\perp$) via the EtM construction, we get an AE-secure $\overline{\mathsf{DEM}}$.

Now let $\mathsf{KEM} = (\mathsf{KGen}^{\mathrm{kem}}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM that is SCFR-CCA, IND-CCA and ANO-CCA secure. Also let H be a collision-resistant hash function with its range being the key space of the IND-CPA symmetric encryption scheme used to obtain $\overline{\mathsf{DEM}}$. We construct $\overline{\mathsf{KEM}} = (\mathsf{KGen}, \mathsf{Encap}, \overline{\mathsf{Decap}})$ where the only difference from KEM is that the ciphertext space is augmented by a special bitstring $\bar{c}$. With respect to $\bar{c}$, the decapsulation algorithm works as follows: $\overline{\mathsf{Decap}}(\mathsf{sk}, \bar{c}) = (H(\mathsf{pk}), \bar{k})$, for any key-pair $(\mathsf{pk}, \mathsf{sk})$ generated by KGen and the fixed $\overline{\mathsf{MAC}}$ key $\bar{k}$ described above. It is not hard to see that $\overline{\mathsf{KEM}}$ is also IND-CCA and ANO-CCA secure. To argue for the SCFR-CCA security of $\overline{\mathsf{KEM}}$, the only additional case to consider is when the adversary returns the final ciphertext $\bar{c}$. Note that $\overline{\mathsf{Decap}}(\mathsf{sk}_0, \bar{c})$ = $\overline{\mathsf{Decap}}(\mathsf{sk}_1, \bar{c})$, or equivalently, $(H(\mathsf{pk}_0), \bar{k}) = (H(\mathsf{pk}_1), \bar{k})$, happens with a negligible probability because of the collision-resistance of H.

Note that the resulting hybrid PKE scheme obtained by composing $\overline{\mathsf{KEM}}$ and $\overline{\mathsf{DEM}}$ is not SROB-CCA secure. This is because an SROB-CCA adversary, upon receiving two public keys $\mathsf{pk}_0, \mathsf{pk}_1$, could simply output the ciphertext $(\bar{c}, (c', t'))$ where (c', t') is an arbitrary $\overline{\mathsf{DEM}}$ ciphertext. The adversary wins the SROB-CCA game because when decrypting $(\bar{c}, (c', t'))$ under sk_i ($i \in \{0, 1\}$) we have $\overline{\mathsf{Decap}}(\mathsf{sk}_i, \bar{c})$ = $(H(\mathsf{pk}_i), \bar{k})$. Since the use of key $\bar{k}$ always leads to successful verification of the $\overline{\mathsf{DEM}}$ ciphertext and the underlying IND-CPA symmetric encryption never rejects, we thus have that the final decryption of $(\bar{c}, (c', t'))$ does *not* return $\perp$ under either of the secret keys sk_0, sk_1.

□

4.4.2 *Anonymity*

Now we turn to the question of what can be said about anonymity for hybrid PKE schemes built from KEMs offering implicit rejection. We prove a negative result that strengthens an analogous result of [77]. That result showed that there exist KEMs that are ANO-CCA (and IND-CCA) secure and XROB-secure DEMs, such that the hybrid PKE scheme resulting from their composition is *not* ANO-CCA secure. Thus anonymity is not preserved in the hybrid construction. However the KEM construction that was used to show this negative result in [77] is not SCFR-CCA secure, which might lead one to think that the strong collision freeness of implicit rejection KEMs might be sufficient to preserve anonymity. Here we show this not to be true.

Before we discuss our negative result in more detail, we first define the notion of *claw-free permutations* as introduced by Goldwasser *et al.* [81] in the context of constructing secure digital signature schemes (also see [9, Section 4.2]). A claw-free pair of permutations [GMR88] is a pair of trapdoor permutations[4] $(\mathcal{F}_1, \mathcal{F}_2)$, where $\mathcal{F}_i = (\mathsf{G}_i, f_i, f_i^{-1})$ with the following properties:

1. We have the key generation algorithms $\mathsf{G}_1 = \mathsf{G}_2$. Hence we denote $\mathsf{G} = \mathsf{G}_1 = \mathsf{G}_2$.
2. For any public key pk, $f_1(\mathsf{pk}, \cdot)$ and $f_2(\mathsf{pk}, \cdot)$ have the same domain and range.
3. Given only pk, the probability that any efficient adversary can find a pair (x_1, x_2) such that $f_1(\mathsf{pk}, x_1) = f_2(\mathsf{pk}, x_2)$ is negligible. Such a pair (x_1, x_2) is called a *claw*. It is straightforward to formalize this property in the form of a *claw-finding* security game (see e.g., Lemma 11), similar to the game-based security notions seen previously.

Theorem 6. *Suppose there exists a KEM that is simultaneously* SROB-CCA, IND-CCA *and* ANO-CCA *secure, a claw-free pair of permutations with domain and range being the encapsulated key space of the KEM, and a collision-resistant hash function. Suppose also that there exists a DEM that is* AE*-secure and* XROB*-secure. Then there exists an implicit rejection KEM that is* SCFR-CCA, IND-CCA *and* ANO-CCA *secure and a DEM that is* AE*-secure and* XROB*-secure, such that the resulting hybrid PKE is not* ANO-CCA *secure.*

Proof. Let $\mathsf{KEM} = (\mathsf{KGen}^{\mathsf{kem}}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM that is IND-CCA, ANO-CCA and SROB-CCA secure. Let $(\mathcal{F}_1, \mathcal{F}_2)$ be a claw-free pair of permutations, with the domain and range being the encapsulated key space of KEM, and let H be a collision-resistant hash function that maps the space of public keys of KEM to the encapsulated key space. We now construct

[4] We refer the reader to [46] for a formal definition of this standard cryptographic primitive.

```
$\overline{\mathsf{Encap}}(\mathsf{pk})$
$(c,k) \leftarrow \mathsf{Encap}(\mathsf{pk})$
$\overline{k} := f_1(\mathsf{PK},k)$
return $(c,\overline{k})$

$\overline{\mathsf{Decap}}(\mathsf{sk},c)$
$k' := \mathsf{Decap}(\mathsf{sk},c)$
if $k' = \perp$ then
    $\overline{k}' := f_2(\mathsf{PK},H(\mathsf{pk}))$
else $\overline{k}' := f_1(\mathsf{PK},k')$
return $\overline{k}'$
```

FIGURE 4.8: $\overline{\mathsf{Encap}}$ and $\overline{\mathsf{Decap}}$ algorithms of $\overline{\mathsf{KEM}}$ for the proof of Theorem 6.

$\overline{\mathsf{KEM}}$ = ($\mathsf{KGen}^{\mathsf{kem}}$, $\overline{\mathsf{Encap}}$, $\overline{\mathsf{Decap}}$) that is IND-CCA, ANO-CCA and SCFR-CCA secure, but when composed with an XROB-secure DEM, does not result in an ANO-CCA secure hybrid PKE scheme.

We first generate public parameters for $\overline{\mathsf{KEM}}$ which are related to the instantiation of $(\mathcal{F}_1, \mathcal{F}_2)$. Recall that $\mathcal{F}_i = (\mathsf{G}_i, f_i, f_i^{-1})$ where $\mathsf{G} = \mathsf{G}_1 = \mathsf{G}_2$ is the key-generator for the pair of claw-free permutations. Hence, we generate the public parameters $f_1(\mathsf{PK}, .)$ and $f_2(\mathsf{PK}, .)$, where PK is the public key of the pair of claw-free permutations. The subsequent key generation algorithm of $\overline{\mathsf{KEM}}$ (which is independent of the generation of public parameters) is the same as that of KEM. The $\overline{\mathsf{Encap}}$ and $\overline{\mathsf{Decap}}$ algorithms of $\overline{\mathsf{KEM}}$ are described in Figure 4.8.

It is not hard to see that $\overline{\mathsf{KEM}}$ is also ANO-CCA secure. To argue about the IND-CCA security of $\overline{\mathsf{KEM}}$ based on the IND-CCA security of KEM, we need to observe in the reduction that when the IND-CCA challenger of KEM returns a uniformly random key k (in the real-or-random experiment), $f_1(\mathsf{PK}, k)$ is a uniformly random key as well, since $f_1(\mathsf{PK}, .)$ is a permutation. To show the SCFR-CCA security of $\overline{\mathsf{KEM}}$, consider an SCFR-CCA adversary that, after receiving two $\overline{\mathsf{KEM}}$ public keys $\mathsf{pk}_0, \mathsf{pk}_1$, wins the corresponding security game by returning a ciphertext c such that $\overline{\mathsf{Decap}}(\mathsf{sk}_0, c)$ = $\overline{\mathsf{Decap}}(\mathsf{sk}_1, c)$. There are 3 cases to consider:

- **Case 1:** If $\mathsf{Decap}(\mathsf{sk}_0, c) \neq \perp$ and $\mathsf{Decap}(\mathsf{sk}_1, c) \neq \perp$, then we can break the SROB-CCA security of KEM via a straightforward reduction.

- **Case 2:** If $\mathsf{Decap}(\mathsf{sk}_0, c) = \perp$ and $\mathsf{Decap}(\mathsf{sk}_1, c) = \perp$, then this would mean that $f_2(\mathsf{PK}, H(\mathsf{pk}_0)) = f_2(\mathsf{PK}, H(\mathsf{pk}_1))$. This would break the collision-resistance of H as $f_2(\mathsf{PK}, .)$ is a permutation, and with high probability, $\mathsf{pk}_0 \neq \mathsf{pk}_1$.

- **Case 3:** Without loss of generality, let $\mathsf{Decap}(\mathsf{sk}_0, c) = k \neq \perp$ and let $\mathsf{Decap}(\mathsf{sk}_1, c) = \perp$. This would mean that $f_1(\mathsf{PK}, k) = f_2(\mathsf{PK}, H(\mathsf{pk}_1))$. But then the pair $(k, H(\mathsf{pk}_1))$ is a claw w.r.t. $f_1(\mathsf{PK}, .)$ and $f_2(\mathsf{PK}, .)$ which breaks the underlying claw-freeness assumption of $(\mathcal{F}_1, \mathcal{F}_2)$.

Now let $\mathsf{DEM} = (\mathsf{KGen}^{\mathsf{dem}}, \mathsf{Enc}, \mathsf{Dec})$ be an AE-secure DEM which is additionally XROB-secure. We describe and then analyse an adversary $\mathcal{A}$ for the ANO-CCA security game against the hybrid PKE scheme resulting from the composition of $\overline{\mathsf{KEM}}$ and DEM.

Upon receiving two public keys pk_0 and pk_1 (along with the public parameters $f_1(\mathsf{PK}, \cdot)$ and $f_2(\mathsf{PK}, \cdot)$), $\mathcal{A}$ selects an arbitrary message m and forwards the challenge message m in the ANO-CCA game. It then receives the ciphertext $c = (c_0, c_1)$ where $(c_0, k) \leftarrow \overline{\mathsf{Encap}}(\mathsf{pk}_b)$ and $c_1 \leftarrow \mathsf{Enc}(k, m)$, for a uniformly random bit $b \leftarrow\$ \{0, 1\}$. Then, $\mathcal{A}$ asks for the decryption of ciphertext $c' = (c_0, c_1')$ w.r.t. sk_0 where $c_1' = \mathsf{Enc}(\hat{k}, m)$ with $\hat{k} = f_2(\mathsf{PK}, H(\mathsf{pk}_0))$. If the response is $\perp$, then the adversary $\mathcal{A}$ outputs 0; else, it outputs 1.

To see why $\mathcal{A}$ breaks ANO-CCA security of the hybrid PKE scheme, consider the following 2 cases:

- $\underline{b = 0}$: In the decryption of $c' = (c_0, c_1')$ w.r.t. sk_0, we have that $\mathsf{Decap}(\mathsf{sk}_0, c_0) = k'$ where $f_1(\mathsf{PK}, k') = k$. Therefore, we have $f_1(\mathsf{PK}, k') = k \neq f_2(\mathsf{PK}, H(\mathsf{pk}_0))$ (i.e., $k \neq \hat{k}$) with a high probability owing to the claw-freeness of $(\mathcal{F}_1, \mathcal{F}_2)$. Since DEM is XROB-secure, we also have $\mathsf{Dec}(k, \mathsf{Enc}(\hat{k}, m)) = \perp$ with a high probability. Hence, the adversary guesses correctly by outputting 0.

- $\underline{b = 1}$: In the decryption of $c' = (c_0, c_1')$ w.r.t. sk_0, we have that $\mathsf{Decap}(\mathsf{sk}_0, c_0) = \perp$ with a high probability because the underlying KEM is SROB-CCA secure (note that $\mathsf{Encap}(\mathsf{pk}_1) = (c_0, k')$ where $f_1(\mathsf{PK}, k') = k$). Because of the way $\overline{\mathsf{KEM}}$ was constructed, we thus have $\overline{\mathsf{Decap}}(\mathsf{sk}_0, c_0) = f_2(\mathsf{PK}, H(\mathsf{pk}_0))(= \hat{k})$. Therefore, we have $\mathsf{Dec}(\hat{k}, \mathsf{Enc}(\hat{k}, m)) = m \neq \perp$. Again, the adversary guesses correctly by outputting 1.

□

The consequence of the above theorem (and its counterexample) is that, for implicit rejection KEMs, we cannot hope to transfer anonymity properties of the KEM to those of the hybrid PKE scheme resulting from the standard KEM-DEM composition in a fully generic manner. To make further progress in this direction, then, we need to look more closely at specific KEM constructions. In the next section, we will look at such KEMs obtained from a standard *implicitly-rejecting* FO transform in the literature, namely $\mathsf{FO}^{\not\perp}$ as seen in Chapter 3.

4.5 ANONYMITY AND ROBUSTNESS OF KEMS OBTAINED FROM $\mathsf{FO}^{\not\perp}$

As discussed in Chapter 3, Fujisaki and Okamoto [1, 2, 79] introduced generic transformations that turn weakly secure PKE schemes into IND-CCA secure KEMs and PKE schemes. Several distinct transforms have emerged, each with slightly different flavours – as analyzed in [4]. One main distinction is whether the constructed KEM offers implicit rejection (e.g., $\mathsf{FO}^{\not\perp}$ in [4]) or explicit rejection ($\mathsf{FO}^{\perp}$). As we have already seen, this distinction is important in considering robustness. And since all NIST PQC final-round candidates in the KEM/PKE category except one alternate candidate offer implicit rejection, we mainly focus on the corresponding implicitly-rejecting $\mathsf{FO}^{\not\perp}$ transform. Also, since we are mainly concerned with the post-quantum setting in this thesis, our analysis that follows will be in the QROM.

Now given a base PKE scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and hash functions G_r and G_k, the KEM $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G_r, G_k]$ is defined in Figure 3.2. As described in [4], the $\mathsf{FO}^{\not\perp}$ transform "implicitly" uses a modular transformation T that converts a OW-CPA/IND-CPA secure PKE scheme PKE into a *deterministic* PKE scheme $\mathsf{PKE}_1 = \mathsf{T}\left[\mathsf{PKE}, G_r\right] = \left(\mathsf{KGen}, \mathsf{Enc}', \mathsf{Dec}'\right)$ that is secure in the presence of so-called *plaintext-checking attacks*.[5]The transformation is described in Figure 4.9.

It was proved in [5] that the $\mathsf{FO}^{\not\perp}$ transform lifts IND-CPA security of PKE to IND-CCA security of $\mathsf{KEM}^{\not\perp}$ in the QROM. We provide some further enhancement results for $\mathsf{FO}^{\not\perp}$. They demonstrate that, provided the starting base PKE scheme PKE and the derived deterministic scheme PKE_1 satisfy some mild security assumptions on anonymity (wANO-CPA[6]) and collision-freeness (SCFR-CPA) respectively, then $\mathsf{FO}^{\not\perp}$ confers strong anonymity (ANO-CCA) and collision-freeness (SCFR-CCA) to the final $\mathsf{KEM}^{\not\perp}$ in the QROM. We first focus on anonymity in the following.

[5] Technically, PKE_1 satisfies One-Wayness under Plaintext Checking Attacks (OW-PCA security). At a high level, the corresponding security game is similar to that of the OW-CPA notion (Figure 2.1) but where the adversary additionally has access to a plaintext checking oracle $\mathrm{Pco}(c, m)$ which outputs 1 if the decryption of ciphertext c returns the message m and outputs 0 otherwise.

[6] The wANO-CPA security notion for PKE is a weaker variant of ANO-CPA where, in the corresponding security game, the challenger encrypts a uniformly random *secret* message under either of the two honestly generated public-keys and *only* provides the resulting ciphertext to the adversary, along with the generated public-keys.

```
$\mathsf{Enc}'(\mathsf{pk}, m)$
    $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; G_r(m))$
    return $c$

$\mathsf{Dec}'(\mathsf{sk}, c)$
    $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
    $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; G_r(m'))$
    if $c = c'$ then
        return $m'$
    else return $\perp$
```

FIGURE 4.9: Deterministic PKE scheme PKE_1 = T[PKE, G_r]; algorithm KGen for PKE_1 is the same as that of PKE.

Theorem 7. *Given* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct and has message space $\mathcal{M}$. Then for any* ANO-CCA *adversary $\mathcal{A}$ against* $\mathsf{KEM}^{\not\perp} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_{G_r} and q_{G_k} queries to the quantum random oracles G_r and G_k respectively, and at most q_D queries to the (classical) decapsulation oracle, there exist* wANO-CPA *adversary $\mathcal{B}$ and* OW-CPA *adversary $\mathcal{B}'$ against* PKE*, and* SCFR-CPA *adversary $\mathcal{B}''$ against* $\mathsf{PKE}_1 = (\mathsf{KGen}, \mathsf{Enc}', \mathsf{Dec}')$ *issuing at most q_{G_r} queries to G_r, such that:*

$$\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{wANO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) + 2(q_{G_r} + q_{G_k})\sqrt{\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}')} \\ + q_D \cdot \mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}'') + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 2q_{G_r}(q_D + 2)\sqrt{2\delta}\,.$$

Moreover, the running times of $\mathcal{B}$, $\mathcal{B}'$ and $\mathcal{B}''$ are the same as that of $\mathcal{A}$.

Proof. Denote $\Omega_{\mathbf{G}_r}$, $\Omega_{\mathbf{G}_k}$, $\Omega_{\mathbf{G}'_k}$ and $\Omega_{\mathbf{G}''_k}$ to be the set of all functions $\mathbf{G}_r : \mathcal{M} \to \mathcal{R}$, $\mathbf{G}_k : \mathcal{C} \to \mathcal{K}$, $\mathbf{G}'_k : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ and $\mathbf{G}''_k : \mathcal{M} \to \mathcal{K}$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{K}$ is the encapsulated key space of $\mathsf{KEM}^{\not\perp}$ and $\mathcal{C}$ is the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$.

Let $\mathcal{A}$ be an adversary in the ANO-CCA game for $\mathsf{KEM}^{\not\perp}$ issuing at most q_D (classical) queries to the oracle $\textsc{Decaps}_{c^*}$, and q_{G_r} and q_{G_k} quantum queries to the random oracles G_r and G_k respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_8$ described in Figure 4.10.

Game G_0**:** The game G_0 is exactly the ANO-CCA game for $\mathsf{KEM}^{\not\perp}$ (= $\mathsf{FO}^{\not\perp}$[PKE, G_r, G_k]). Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A})$$

Games $\mathsf{G}_{0.5} - \mathsf{G}_1$**:** In game $\mathsf{G}_{0.5}$, we modify the decapsulation oracle $\textsc{Decaps}_{c^*}(0, \cdot)$ such that $G^{\mathsf{rej}}_{0k}(c)$ is returned instead of $G_k(s_0, c)$ for an in-

```
Games G_0 - G_8
$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
$G_r \xleftarrow{\$} \Omega_{\mathbf{G}_r}$
$G_r^{\text{good}} \leftarrow \Omega_{\mathbf{G}_r}$   // Sampling distribution
    // described in description of $\mathsf{G}_2$ below.
$G_r := G_r^{\text{good}}$   // $\mathsf{G}_2$ - $\mathsf{G}_5$
$G_{0k}^{\text{acc}}, G_{1k}^{\text{acc}}, G_{0k}^{\text{rej}}, G_{1k}^{\text{rej}} \xleftarrow{\$} \Omega_{\mathbf{G}_k}$
$G_{2k} \xleftarrow{\$} \Omega_{\mathbf{G}'_k}; G_{3k} \xleftarrow{\$} \Omega_{\mathbf{G}''_k}$
$b \xleftarrow{\$} \{0,1\}$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* \leftarrow G_r(m^*)$   // $\mathsf{G}_0 - \mathsf{G}_6$
$r^* \xleftarrow{\$} \mathcal{R}$   // $\mathsf{G}_7 - \mathsf{G}_8$
$c^* := \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
$k^* \leftarrow G_k(m^*, c^*)$   // $\mathsf{G}_0 - \mathsf{G}_6$
$k^* \xleftarrow{\$} \mathcal{K}$   // $\mathsf{G}_7 - \mathsf{G}_8$
$\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
$i \xleftarrow{\$} \{1, \ldots, q_{G_r} + q_{G_k}\}$   // $\mathsf{G}_8$
run $\mathcal{A}^{G_r, G_k, \textsc{Decaps}_{c^*}}(\mathsf{inp})$ until
    $i$-th query to $G_r \times G_{3k}$   // $\mathsf{G}_8$
measure the $i$-th query and let the
    outcome be $m'$   // $\mathsf{G}_8$
return $[m' = m^*]$   // $\mathsf{G}_8$
$b' \leftarrow \mathcal{A}^{G_r, G_k, \textsc{Decaps}_{c^*}}(\mathsf{inp})$
return $[b' = b]$

$G_k(m, c)$
if $c = c^*$ return $G_{3k}(m)$   // $\mathsf{G}_5$ - $\mathsf{G}_8$
if $\mathsf{Enc}(\mathsf{pk}_0, m; G_r(m)) = c$   // $\mathsf{G}_3$ - $\mathsf{G}_8$
    return $G_{0k}^{\text{acc}}(c)$   // $\mathsf{G}_3$ - $\mathsf{G}_8$
if $\mathsf{Enc}(\mathsf{pk}_1, m; G_r(m)) = c$   // $\mathsf{G}_3$ - $\mathsf{G}_8$
    return $G_{1k}^{\text{acc}}(c)$   // $\mathsf{G}_3$ - $\mathsf{G}_8$
return $G_{2k}(m, c)$

$\textsc{Decaps}_a(0, c)$   // $c \neq a$
return $G_{0k}^{\text{acc}}(c)$   // $\mathsf{G}_{3.5}$ - $\mathsf{G}_8$
Parse $\mathsf{sk}'_0 = (\mathsf{sk}_0, s_0)$
$m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
if $\mathsf{Enc}(\mathsf{pk}_0, m', G_r(m')) = c$ then
    return $G_k(m', c)$
else return $G_k(s_0, c)$   // $\mathsf{G}_0$
else return $G_{0k}^{\text{rej}}(c)$   // $\mathsf{G}_{0.5}$ - $\mathsf{G}_3$

$\textsc{Decaps}_a(1, c)$   // $c \neq a$
return $G_{1k}^{\text{acc}}(c)$   // $\mathsf{G}_4$ - $\mathsf{G}_8$
Parse $\mathsf{sk}'_1 = (\mathsf{sk}_1, s_1)$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
if $\mathsf{Enc}(\mathsf{pk}_1, m', G_r(m')) = c$ then
    return $G_k(m', c)$
else return $G_k(s_1, c)$   // $\mathsf{G}_0$ - $\mathsf{G}_{0.5}$
else return $G_{1k}^{\text{rej}}(c)$   // $\mathsf{G}_1$ - $\mathsf{G}_{3.5}$
```

FIGURE 4.10: Games G_0 – G_8 for the proof of Theorem 7.

valid ciphertext c. Then in game G_1, we modify the decapsulation oracle $\textsc{Decaps}_{c^*}(1,\cdot)$ such that $G^{\mathsf{rej}}_{1k}(c)$ is returned instead of $G_k(s_1,c)$ for an invalid ciphertext c. Here the random oracles G^{rej}_{0k} and G^{rej}_{1k} are not directly accessible to $\mathcal{A}$.

We can use Lemma 2 w.r.t. the pseudorandomness of $G_k(s_0,\cdot)$ and $G_k(s_1,\cdot)$, with PRF keys $s_0, s_1 \leftarrow\!\!\$\ \mathcal{M}$ respectively, to obtain the following via straightforward reductions:

$$|\Pr[\mathsf{G}_{0.5}=1]-\Pr[\mathsf{G}_0=1]| \leq \frac{2q_{G_k}}{\sqrt{|\mathcal{M}|}},$$
$$|\Pr[\mathsf{G}_1=1]-\Pr[\mathsf{G}_{0.5}=1]| \leq \frac{2q_{G_k}}{\sqrt{|\mathcal{M}|}}.$$

Game G_2**:** In game G_2, we change the random oracle G_r such that it uniformly samples "good" random coins w.r.t. the key-pairs $(\mathsf{pk}_0,\mathsf{sk}_0)$ and $(\mathsf{pk}_1,\mathsf{sk}_1)$, similar to the "$\mathsf{G}_3 \to \mathsf{G}_4$" game-hop in our proof of Theorem 1 above. Namely, given a PKE key-pair $(\mathsf{pk},\mathsf{sk})$ and a message $m \in \mathcal{M}$, define

$$\mathcal{R}_{\text{good}}((\mathsf{pk},\mathsf{sk}),m) = \{r \in \mathcal{R} \mid \mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},m;r)) = m\}$$

and $\mathcal{R}_{\text{bad}}((\mathsf{pk},\mathsf{sk}),m) = \mathcal{R} \setminus \mathcal{R}_{\text{good}}((\mathsf{pk},\mathsf{sk}),m)$. For notational convenience, let $\mathsf{KP}_0 = (\mathsf{pk}_0,\mathsf{sk}_0)$ and $\mathsf{KP}_1 = (\mathsf{pk}_1,\mathsf{sk}_1)$. Now we define the oracle $G_r^{\text{good}} \leftarrow \Omega_{\mathbf{G}_r}$ such that $G_r^{\text{good}}(m)$ is sampled according to a uniform distribution in $(\mathcal{R}_{\text{good}}(\mathsf{KP}_0,m) \cap \mathcal{R}_{\text{good}}(\mathsf{KP}_1,m))$; in G_2, we are essentially replacing the oracle G_r with G_r^{good}.

At a high level, our following analysis of the current $\mathsf{G}_1 \to \mathsf{G}_2$ game-hop can be seen as an extension of that of the "$\mathsf{G}_1 \to \mathsf{G}_2$" game-hop in the proof of [5, Theorem 1]; the latter analysis only considers a single key-pair, whereas we extend its arguments to two key-pairs. For the sake of completeness, we provide the full analysis below.

Note that the task of distinguishing between G_1 and G_2 is equivalent to that of distinguishing between the oracles G_r and G_r^{good}. To be specific, for any two fixed key-pairs KP_0, KP_1 generated by KGen, we can construct an oracle distinguisher B between G_r and G_r^{good} such that $B^{G_r}(\mathsf{KP}_0,\mathsf{KP}_1)$ simulates G_1, and $B^{G_r^{\text{good}}}(\mathsf{KP}_0,\mathsf{KP}_1)$ simulates G_2. That is, we have

$$\begin{aligned}&\big|\Pr[\mathsf{G}_2=1 \mid \mathsf{KP}_0,\mathsf{KP}_1 \leftarrow \mathsf{KGen}] - \Pr[\mathsf{G}_1=1 \mid \mathsf{KP}_0,\mathsf{KP}_1 \leftarrow \mathsf{KGen}]\big| \\ &\quad= \big|\Pr[1 \leftarrow B^{G_r^{\text{good}}}(\mathsf{KP}_0,\mathsf{KP}_1)] - \Pr[1 \leftarrow B^{G_r}(\mathsf{KP}_0,\mathsf{KP}_1)]\big|\end{aligned}$$

```
$C^N(\mathsf{KP}_0, \mathsf{KP}_1)$
  Pick a $2q_G$-wise function $f$
  $b'' \leftarrow B^{\hat{G}}(\mathsf{KP}_0, \mathsf{KP}_1)$
  return $b''$

$\hat{G}(m)$
  if $N(m) = 0$
    $\hat{G}(m) = \mathsf{Sample}(\mathcal{R}_{\text{good}}(\mathsf{KP}_0, m) \cap \mathcal{R}_{\text{good}}(\mathsf{KP}_1, m); f(m))$
  else
    $\hat{G}(m) = \mathsf{Sample}(\mathcal{R}_{\text{bad}}(\mathsf{KP}_0, m) \cup \mathcal{R}_{\text{bad}}(\mathsf{KP}_1, m); f(m))$
  return $\hat{G}(m)$
```

FIGURE 4.11: Algorithm C^N for the proof of Theorem 7. $\mathsf{Sample}(\mathcal{Y})$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow\!\!\$\ \mathcal{Y}$ and $\mathsf{Sample}(\mathcal{Y}; f(m))$ denotes the deterministic execution of $\mathsf{Sample}(\mathcal{Y})$ using explicit randomness $f(m)$.

Now any such distinguisher between G_r and G_r^{good} can be converted to a distinguisher between N_0 and N_1 where $N_0 : \mathcal{M} \to \{0,1\}$ is an oracle such that $N_0(m)$ is sampled according to the Bernoulli distribution $\mathbf{B}_{\delta(\mathsf{KP}_0, \mathsf{KP}_1, m)}$[7], where

$$\delta(\mathsf{KP}_0, \mathsf{KP}_1, m) = \frac{|\mathcal{R}_{\text{bad}}(\mathsf{KP}_0, m) \cup \mathcal{R}_{\text{bad}}(\mathsf{KP}_1, m)|}{|\mathcal{R}|}$$

and $N_1 : \mathcal{M} \to \{0,1\}$ is an oracle that always outputs 0 for any input m. Specifically, for any distinguisher $B^{\hat{G}}(\mathsf{KP}_0, \mathsf{KP}_1)$ with $\hat{G} \in \{G_r, G_r^{\text{good}}\}$, we can construct a distinguisher $C^N(\mathsf{KP}_0, \mathsf{KP}_1)$ with $N \in \{N_0, N_1\}$ that is described in Figure 4.11.

Note that if $N = N_0$, then $\hat{G} = G_r$, and if $N = N_1$, then $\hat{G} = G_r^{\text{good}}$. Therefore, for any two fixed key-pairs KP_0, KP_1 generated by KGen, we have $\Pr[1 \leftarrow C^{N_0}(\mathsf{KP}_0, \mathsf{KP}_1)] = \Pr[1 \leftarrow B^{G_r}(\mathsf{KP}_0, \mathsf{KP}_1)]$ and also we have

[7] That is, $\Pr[N_0(m) = 1] = \delta(\mathsf{KP}_0, \mathsf{KP}_1, m)$ and $\Pr[N_0(m) = 0] = 1 - \delta(\mathsf{KP}_0, \mathsf{KP}_1, m)$.

$\Pr[1 \leftarrow C^{N_1}(\mathsf{KP}_0, \mathsf{KP}_1)] = \Pr[1 \leftarrow B^{G_r^{\text{good}}}(\mathsf{KP}_0, \mathsf{KP}_1)]$. Hence, from Lemma 5, we get

$$\begin{aligned} &|\Pr[1 \leftarrow B^{G_r^{\text{good}}}(\mathsf{KP}_0, \mathsf{KP}_1)] - \Pr[1 \leftarrow B^{G_r}(\mathsf{KP}_0, \mathsf{KP}_1)]| \\ &= |\Pr[1 \leftarrow C^{N_1}(\mathsf{KP}_0, \mathsf{KP}_1)] - \Pr[1 \leftarrow C^{N_0}(\mathsf{KP}_0, \mathsf{KP}_1)]| \\ &\qquad\qquad\qquad\qquad\qquad \leq 2q_{G_r}\sqrt{\delta(\mathsf{KP}_0, \mathsf{KP}_1)} \end{aligned}$$

where $\delta(\mathsf{KP}_0, \mathsf{KP}_1) = \max_{m \in \mathcal{M}} \delta(\mathsf{KP}_0, \mathsf{KP}_1, m)$. Hence, conditioned on two fixed key-pairs KP_0, KP_1 generated by KGen, we obtain

$$\begin{aligned} &|\Pr[\mathsf{G}_2 = 1 \mid \mathsf{KP}_0, \mathsf{KP}_1 \leftarrow \mathsf{KGen}] - \Pr[\mathsf{G}_1 = 1 \mid \mathsf{KP}_0, \mathsf{KP}_1 \leftarrow \mathsf{KGen}]| \\ &\qquad\qquad\qquad\qquad\qquad \leq 2q_{G_r}\sqrt{\delta(\mathsf{KP}_0, \mathsf{KP}_1)} \end{aligned}$$

Averaging over $\mathsf{KP}_0 \leftarrow \mathsf{KGen}$, $\mathsf{KP}_1 \leftarrow \mathsf{KGen}$, and applying Jensen's inequality w.r.t. the square root function, we get

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq 2q_{G_r}\sqrt{\mathbf{E}[\delta(\mathsf{KP}_0, \mathsf{KP}_1)]}$$

where the expectation is taken over $\mathsf{KP}_0 \leftarrow \mathsf{KGen}$, $\mathsf{KP}_1 \leftarrow \mathsf{KGen}$. From the notion of δ-correctness, note that for a key-pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$, $\mathbf{E}[\delta((\mathsf{pk}, \mathsf{sk}))] = \delta$, where we have $\delta((\mathsf{pk}, \mathsf{sk})) = \max_{m \in \mathcal{M}} \delta((\mathsf{pk}, \mathsf{sk}), m)$ and $\delta((\mathsf{pk}, \mathsf{sk}), m) = \frac{\mathcal{R}_{\text{bad}}(\mathsf{pk}, \mathsf{sk}, m)}{\mathcal{R}}$. We now show that for two key-pairs $\mathsf{KP}_0 = (\mathsf{pk}_0, \mathsf{sk}_0)$, $\mathsf{KP}_1 = (\mathsf{pk}_1, \mathsf{sk}_1)$, we have $\delta(\mathsf{KP}_0, \mathsf{KP}_1) \leq 2\delta$. First note that, for a particular message m, $\delta(\mathsf{KP}_0, \mathsf{KP}_1, m) \leq \delta(\mathsf{KP}_0, m) + \delta(\mathsf{KP}_1, m)$, and hence, $\delta(\mathsf{KP}_0, \mathsf{KP}_1) \leq \delta(\mathsf{KP}_0) + \delta(\mathsf{KP}_1)$. We now have the following

$$\begin{aligned} \mathbf{E}[\delta(\mathsf{KP}_0, \mathsf{KP}_1)] &= \sum_{\substack{\mathsf{KP}_0 \\ \mathsf{KP}_1}} \Pr[\mathsf{KP}_0] \Pr[\mathsf{KP}_1] \delta(\mathsf{KP}_0, \mathsf{KP}_1) \\ &\leq \sum_{\substack{\mathsf{KP}_0 \\ \mathsf{KP}_1}} \Pr[\mathsf{KP}_0] \Pr[\mathsf{KP}_1] (\delta(\mathsf{KP}_0) + \delta(\mathsf{KP}_1)) \\ &= \sum_{\mathsf{KP}_1} \Big(\sum_{\mathsf{KP}_0} \Pr[\mathsf{KP}_0] \delta(\mathsf{KP}_0) \Big) \Pr[\mathsf{KP}_1] \\ &+ \sum_{\mathsf{KP}_0} \Big(\sum_{\mathsf{KP}_1} \Pr[\mathsf{KP}_1] \delta(\mathsf{KP}_1) \Big) \Pr[\mathsf{KP}_0] \\ &= \sum_{\mathsf{KP}_1} \delta \cdot \Pr[\mathsf{KP}_1] + \sum_{\mathsf{KP}_0} \delta \cdot \Pr[\mathsf{KP}_0] = 2\delta \end{aligned}$$

where $\Pr[\mathsf{KP}_i]$ denotes the probability of the fixed key-pair KP_i being generated by KGen. We also used the fact that the key-pairs KP_0, KP_1 are generated independently. Thus, we finally obtain

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq 2q_{G_r}\sqrt{2\delta}$$

Game G_3**:** In game G_3, we implicitly divide the G_k-queries (m, c) into three disjoint categories: (1) $\mathsf{Enc}(\mathsf{pk}_0, m; G_r(m)) = c$, (2) $\mathsf{Enc}(\mathsf{pk}_0, m; G_r(m)) \neq c = \mathsf{Enc}(\mathsf{pk}_1, m; G_r(m))$, and (3) $\mathsf{Enc}(\mathsf{pk}_0, m; G_r(m)) \neq c \wedge \mathsf{Enc}(\mathsf{pk}_1, m; G_r(m)) \neq c$. We then respond to the queries from the respective categories with $G_{0k}^{\mathrm{acc}}(c)$, $G_{1k}^{\mathrm{acc}}(c)$ and $G_{2k}(m, c)$ respectively, where G_{0k}^{acc} and G_{1k}^{acc} are internal random oracles not directly accessible to the adversary $\mathcal{A}$. Because G_r samples "good" random coins, it is not hard to see that the encryption functions $\mathsf{Enc}(\mathsf{pk}_0, .; G_r(\cdot))$ and $\mathsf{Enc}(\mathsf{pk}_1, .; G_r(\cdot))$ are injective, and hence, the output distributions of the G_k-oracle in the games G_2 and G_3 are equivalent. Therefore,

$$\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_2 = 1]$$

Game $\mathsf{G}_{3.5}$**:** In game $\mathsf{G}_{3.5}$, we change the $\textsc{Decaps}_{c^*}(0, \cdot)$ oracle such that there is no need for the secret key sk'_0. Namely, $G_{0k}^{\mathrm{acc}}(c)$ is returned for the decapsulation of ciphertext c w.r.t. sk'_0. Let $m' = \mathsf{Dec}(\mathsf{sk}_0, c)$. Consider the following two cases:

- $\underline{\mathsf{Enc}(\mathsf{pk}_0, m'; G_r(m')) = c}$: In this case, the $\textsc{Decaps}_{c^*}(0, \cdot)$ oracles in games G_3 and $\mathsf{G}_{3.5}$ return the same value $G_{0k}^{\mathrm{acc}}(c)$.

- $\underline{\mathsf{Enc}(\mathsf{pk}_0, m'; G_r(m')) \neq c}$: In game G_3, as the random oracle G_{0k}^{rej} is independent of all other oracles, the output $G_{0k}^{\mathrm{rej}}(c)$ is uniformly random in the adversary $\mathcal{A}$'s view. In game $\mathsf{G}_{3.5}$, the only way $\mathcal{A}$ gets prior access to the function G_{0k}^{acc} is if it made a G_k-query (m'', c) such that $\mathsf{Enc}(\mathsf{pk}_0, m''; G_r(m'')) = c$. But because G_r samples good random coins, we have $\mathsf{Dec}(\mathsf{sk}_0, c) = m'' = m'$ leading to a contradiction of "$\mathsf{Enc}(\mathsf{pk}_0, m'; G_r(m')) \neq c$". Hence, such a prior access is not possible and $G_{0k}^{\mathrm{acc}}(c)$ will also be a uniformly random value in $\mathcal{A}$'s view.

As the output distributions of the $\textsc{Decaps}_{c^*}(0, \cdot)$ oracle in G_3 and $\mathsf{G}_{3.5}$ are the same in both cases, we have

$$\Pr[\mathsf{G}_{3.5} = 1] = \Pr[\mathsf{G}_3 = 1]$$

Game G_4**:** In game G_4, we change the $\textsc{Decaps}_{c^*}(1, \cdot)$ oracle such that $G_{1k}^{\mathrm{acc}}(c)$ is returned for the decapsulation of *any* ciphertext c w.r.t. sk_1. The analysis here follows quite similarly to that of the previous game-hop except that this simulation of the $\textsc{Decaps}_{c^*}(1,\cdot)$ oracle – without the secret key sk_1' – will fail if $\mathcal{A}$ asks for the decapsulation of a ciphertext $\hat{c}$ such that $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$ and $\mathsf{Enc}(\mathsf{pk}_0, m'; G_r(m')) = \mathsf{Enc}(\mathsf{pk}_1, m'; G_r(m')) = \hat{c}$. In this peculiar case, $G_{0k}^{\mathrm{acc}}(\hat{c})$ is returned in G3 and $G_{1k}^{\mathrm{acc}}(\hat{c})$ is returned in G4.

We bound the probability of this peculiar event (i.e., $\mathcal{A}$ asking for the decapsulation of $\hat{c}$ w.r.t. sk_1') by the advantage of an SCFR-CPA adversary $\mathcal{B}''$ against the deterministic scheme $\mathsf{PKE}_1^{\mathrm{good}} = \mathsf{T}[\mathsf{PKE}, G_r^{\mathrm{good}}]$. First note that, because G_r^{good} samples good random coins, for such ciphertexts $\hat{c}$ we have $\mathsf{Dec}(sk_0, \hat{c}) = \mathsf{Dec}(sk_1, \hat{c}) = m'$ and $\mathsf{Enc}(\mathsf{pk}_0, m'; G_r^{\mathrm{good}}(m')) = \mathsf{Enc}(\mathsf{pk}_1, m'; G_r^{\mathrm{good}}(m')) = \hat{c}$. Note that such a $\hat{c}$ corresponds to winning the SCFR-CPA game of $\mathsf{PKE}_1^{\mathrm{good}}$. So we can construct a corresponding SCFR-CPA adversary $\mathcal{B}''$ that has access to the (non-ideal) "good" random oracle G_r^{good}. Upon receiving two public keys pk_0 and pk_1, $\mathcal{B}''$ simulates G_4 for the adversary $\mathcal{A}$ and maintains a list of $\mathcal{A}$'s *classical* queries to the oracle $\textsc{Decaps}_{c^*}(1,\cdot)$ (note that $\mathcal{B}''$ can simulate the decapsulation oracle as in G_4 even with no access to the corresponding secret keys sk_0 and sk_1). Then $\mathcal{B}''$ chooses a ciphertext uniformly at random from the list and forwards it as the final message to the SCFR-CPA challenger of $\mathrm{PKE}_1^{\mathrm{good}}$.

Let Pr[pec] be the probability of this peculiar event, denoted as "pec", occurring. We have the games $\mathsf{G}_{3.5}$ and G_4 to be equivalent unless the event pec occurs. From the construction of the SCFR-CPA adversary $\mathcal{B}''$ above, it is not hard to see that $\mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1^{\mathrm{good}}}(\mathcal{B}'') \geq \frac{1}{q_D} \cdot \Pr[\mathsf{pec}]$. Hence, we have

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_{3.5} = 1]| \leq \Pr[\mathsf{pec}] \leq q_D \cdot \mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1^{\mathrm{good}}}(\mathcal{B}'')$$

Using a similar analysis as the game-hop $\mathsf{G}_1 \to \mathsf{G}_2$, by replacing G_r^{good} with an ideal random oracle G_r w.r.t. the SCFR-CPA adversary $\mathcal{B}''$, we obtain

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_{3.5} = 1]| \leq q_D \cdot (\mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}'') + 2q_{G_r}\sqrt{2\delta})$$

Game G_5**:** In game G_5, we answer G_k-queries of the form (m, c^*) with $G_{3k}(m)$, where G_{3k} is an independent random oracle. Since G_r samples good randomness, there are at most two G_k-queries worth considering, namely (m_0, c^*) and (m_1, c^*), where $\mathsf{Enc}(\mathsf{pk}_0, m_0; G_r(m_0)) = c^*$ and $\mathsf{Enc}(\mathsf{pk}_1, m_1; G_r(m_1)) = c^*$ (for the other G_k-queries (m', c^*), where $m' \notin \{m_0, m_1\}$, we are replacing the oracle outputs $G_{2k}(m', c^*)$ in G_4 with $G_{3k}(m')$ in G_5). W.r.t. these two queries, the G_k oracle would return $G_{0k}^{\mathrm{acc}}(c^*)$, $G_{1k}^{\mathrm{acc}}(c^*)$ respectively in G_4,

```
$A^{G_r \times G_{3k}}(m^*, (r^*, k^*))$
    $(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
    $G^{\mathrm{acc}}_{0k}, G^{\mathrm{acc}}_{1k} \leftarrow\!\!\$\ \Omega_{\mathbf{G}_k}; G_{2k} \leftarrow\!\!\$\ \Omega_{\mathbf{G}'_k}$
    $b \leftarrow\!\!\$\ \{0, 1\}$
    $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
    $b' \leftarrow \mathcal{A}^{G_r, G_k, \textsc{Decaps}_{c^*}}(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
    return $[b' = b]$

$G_k(m, c)$
    if $c = c^*$return $G_{3k}(m)$
    if $\mathsf{Enc}(\mathsf{pk}_0, m; G_r(m)) = c$
        return $G^{\mathrm{acc}}_{0k}(c)$
    if $\mathsf{Enc}(\mathsf{pk}_1, m; G_r(m)) = c$
        return $G^{\mathrm{acc}}_{1k}(c)$
    return $G_{2k}(m, c)$

$\textsc{Decaps}_a(i, c)$    // $i \in \{0, 1\} \wedge c \neq a$
    return $G^{\mathrm{acc}}_{ik}(c)$
```

FIGURE 4.12: Algorithm $A^{G_r \times G_{3k}}$ for the proof of Theorem 7.

and $G_{3k}(m_0)$, $G_{3k}(m_1)$ respectively in G_5. The adversary $\mathcal{A}$'s view would be identical even after this change because the random values $G^{\mathrm{acc}}_{0k}(c^*)$, $G^{\mathrm{acc}}_{1k}(c^*)$ are only accessible to $\mathcal{A}$ via the G_k-oracle in G_4, and in particular, not through the $\textsc{Decaps}_{c^*}$ oracle since c^* is a forbidden decapsulation query. Hence in G_5, we are effectively replacing (at most) two uniformly random values that can only be accessed via the G_k-oracle by $\mathcal{A}$ with two other uniformly random values (the simpler case of $m_0 = m_1$ would follow similarly). Since the output distributions of the G_k-oracle in the games G_4 and G_5 are equivalent, we have

$$\Pr[\mathsf{G}_5 = 1] = \Pr[\mathsf{G}_4 = 1]$$

Game G_6**:** In game G_6, we reset G_r to be an ideal random oracle, i.e., $G_r(m)$ now returns uniformly random coins from $\mathcal{R}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_1 \rightarrow \mathsf{G}_2$, by using a similar analysis, it is not hard to obtain

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq 2q_{G_r}\sqrt{2\delta}$$

Game G_7**:** In the setup of game G_7, we replace the hash evaluations "$r^* \leftarrow G_r(m^*)$" and "$k^* \leftarrow G_k(m^*, c^*)(= G_{3k}(m^*))$" with "$r^* \leftarrow\!\!\$\ \mathcal{R}$" and "$k^* \leftarrow\!\!\$\ \mathcal{K}$" respectively. That is, r^* and k^* are now uniformly random values that are generated independently of the random oracles G_r and G_{3k}. We use Lemma 3 to bound the difference in the success probabilities of $\mathcal{A}$ in G_6 and G_7. Let

A be an oracle algorithm that has quantum access to the random oracle $G_r \times G_{3k}$, where $(G_r \times G_{3k})(m) = (G_r(m), G_{3k}(m))$. Figure 4.12 describes $A^{G_r \times G_{3k}}$'s operation on input $(m^*, (r^*, k^*))$. Note that the algorithm $A^{G_r \times G_{3k}}$ makes at most $q_{G_r} + q_{G_k}$ number of queries to the random oracle $G_r \times G_{3k}$ to respond to $\mathcal{A}$'s oracle queries[8]. With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_6 = 1]$ and $P_A^2 = \Pr[\mathsf{G}_7 = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 3 w.r.t. the algorithm $A^{G_r \times G_{3k}}$; to analyse the corresponding probability P_B in Lemma 3, we define game G_8 (see Fig. 4.10) such that $P_B = \Pr[\mathsf{G}_8 = 1]$. From Lemma 3, we thus have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq 2(q_{G_r} + q_{G_k})\sqrt{\Pr[\mathsf{G}_8 = 1]}.$$

We now bound the success probability of $\mathcal{A}$ in G_7 by the advantage of an adversary $\mathcal{B}$ in the wANO-CPA game of PKE. Upon receiving public keys pk_0 and pk_1 along with the ciphertext c^*, where $c^* := \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ for uniformly random bit $b \leftarrow\$ \{0,1\}$, (secret) message $m^* \leftarrow\$ \mathcal{M}$ and randomness $r^* \leftarrow\$ \mathcal{R}$ chosen by the challenger, $\mathcal{B}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_7.
- Uses a $2q_{G_r}$-wise independent function and four different $2q_{G_k}$-wise independent functions to simulate the random oracles $G_r, G_{0k}^{\text{acc}}, G_{1k}^{\text{acc}}, G_{2k}$ and G_{3k} respectively in $\mathcal{A}$'s view, as noted in Lemma 1. The random oracle G_k is simulated in the same way as in G_7.
- Answers decapsulation queries using the oracles G_{ik}^{acc} $(i \in \{0,1\})$ as in G_7.
- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^* \leftarrow\$ \mathcal{K}$ and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$.
- After obtaining a bit b' from $\mathcal{A}$, forwards b' to its wANO-CPA challenger as the final message.

It is easy to see that $|\Pr[\mathsf{G}_7 = 1] - \frac{1}{2}| = \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{wANO\text{-}CPA}}(\mathcal{B})$. Now we bound the success probability of $\mathcal{A}$ in G_8 by the advantage of an adversary $\mathcal{B}'$ in the OW-CPA game of PKE. Upon receiving a public-key pk along with a ciphertext c^*, where $c^* := \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ for uniformly random (secret) message $m^* \leftarrow\$ \mathcal{M}$ and randomness $r^* \leftarrow\$ \mathcal{R}$ chosen by the challenger, $\mathcal{B}'$ proceeds as follows:

[8] Similar to the reduction in our proof of Theorem 1, if $A^{G_r \times G_{3k}}$ wants to respond to $\mathcal{A}$'s G_k-query, then $A^{G_r \times G_{3k}}$ prepares a uniform superposition of all states in the output register corresponding to G_r (see Footnote 3 of Chapter 3, and also [50]).

- Runs $\mathcal{A}$ as a subroutine as in game G_8.
- Uses three different $2q_{G_k}$-wise independent functions to simulate the random oracles $G_{0k}^{\mathrm{acc}}, G_{1k}^{\mathrm{acc}}$ and G_{2k} respectively, two different $2(q_{G_r} + q_{G_k})$-wise independent functions to simulate the random oracles G_r and G_{3k} respectively in $\mathcal{A}$'s view, as noted in Lemma 1. Also evaluates $\mathcal{A}$'s G_r- and G_k-queries using the oracle $G_r \times G_{3k}$; the random oracle G_k is simulated in the same way as in G_8,
- Answers decapsulation queries using the oracles G_{ik}^{acc} ($i \in \{0,1\}$) as in G_8.
- For $\mathcal{A}$'s challenge query, first samples a uniformly random bit $b \leftarrow\!\!\$ \{0,1\}$ and sets $\mathsf{pk}_b = \mathsf{pk}$. Then generates a key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}$, samples a uniformly random key $k^* \leftarrow\!\!\$ \mathcal{K}$ and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$. (By doing this, note that we have $c^* := \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ in $\mathcal{A}$'s view.)
- Selects $i \leftarrow\!\!\$ \{1, \ldots, q_{G_r} + q_{G_k}\}$, measures the i-th query to oracle $G_r \times G_{3k}$ and returns the outcome m'.

Again, it is not hard to see that $\Pr[\mathsf{G}_8 = 1] \le \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}')$. Hence by collecting all of the above bounds, we arrive at

$$\begin{aligned}\mathbf{Adv}_{\mathsf{KEM}^{\not\perp}}^{\mathsf{ANO\text{-}CCA}}(\mathcal{A}) \le \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{wANO\text{-}CPA}}(\mathcal{B}) + 2(q_{G_r} + q_{G_k})\sqrt{\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}')} \\ + q_D \cdot \mathbf{Adv}_{\mathsf{PKE}_1}^{\mathsf{SCFR\text{-}CPA}}(\mathcal{B}'') + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 2q_{G_r}(q_D + 2)\sqrt{2\delta}\,.\end{aligned}$$

□

Having dealt with anonymity in the above theorem, we now turn to studying robustness. As discussed above, implicit rejection KEMs cannot formally satisfy robust (i.e., SROB, WROB) properties. Fortunately however, we establish the next best thing (according to our definitions), i.e., *strong collision freeness* of the KEMs constructed using $\mathsf{FO}^{\not\perp}$. Towards our result, we require the following *claw-freeness* property of quantum random oracles.

Lemma 11 ([60, Lemma 2.4])**.** *There is a universal constant C (< 648) such that the following holds: Let $\Omega_{\mathbf{H}_0}$ and $\Omega_{\mathbf{H}_1}$ be the set of all functions $\mathbf{H}_0 : \mathcal{X}_0 \to \mathcal{Y}$ and $\mathbf{H}_1 : \mathcal{X}_1 \to \mathcal{Y}$ respectively, such that $|\mathcal{X}_0| \le |\mathcal{X}_1|$. Let $H_0 \leftarrow\!\!\$ \Omega_{\mathbf{H}_0}$ and $H_1 \leftarrow\!\!\$ \Omega_{\mathbf{H}_1}$. For any quantum algorithm A^{H_0,H_1} making q quantum queries to H_0 and H_1, we have*

$$\Pr[H_0(x_0) = H_1(x_1) \mid (x_0, x_1) \leftarrow A^{H_0,H_1}] \le \frac{C(q+1)^3}{|\mathcal{Y}|}.$$

For the following result, we in-fact need a weaker property than the one described in the above lemma; namely, it's hard for an adversary to return a value $x \in \mathcal{X}_0 \cap \mathcal{X}_1$ such that $H_0(x) = H_1(x)$. We leave the derivation of the corresponding upper-bound as an open problem.

Theorem 8. *Given* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct and has message space $\mathcal{M}$. Then for any* SCFR-CCA *adversary $\mathcal{A}$ against* $\mathsf{KEM}^{\not\perp} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_{G_r} and q_{G_k} queries to the quantum random oracles G_r and G_k respectively, and at most q_D queries to the (classical) decapsulation oracle, there exists an* SCFR-CPA *adversary $\mathcal{B}$ against* $\mathsf{PKE}_1 = (\mathsf{KGen}, \mathsf{Enc}', \mathsf{Dec}')$ *issuing at most q_{G_r} queries to G_r such that*

$$\mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \le q_D \cdot \mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{C(q_{G_k}+1)^3}{|\mathcal{K}|} + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 2q_{G_r}(q_D+2)\sqrt{2\delta}\,.$$

Here $\mathcal{K}$ is the encapsulated key space of $\mathsf{KEM}^{\not\perp}$ *and C (< 648) is the constant from Lemma 11. The running time of $\mathcal{B}$ is the same as that of $\mathcal{A}$.*

Proof. Denote $\Omega_{\mathbf{G}_r}, \Omega_{\mathbf{G}_k}, \Omega_{\mathbf{G}'_k}$ to be the set of all functions $\mathbf{G}_r : \mathcal{M} \to \mathcal{R}$, $\mathbf{G}_k : \mathcal{C} \to \mathcal{K}$, $\mathbf{G}'_k : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc and $\mathcal{C}$ is the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$.

Let $\mathcal{A}$ be an adversary in the SCFR-CCA game for $\mathsf{KEM}^{\not\perp}$ issuing at most q_D (classical) queries to the oracle $\textsc{Decaps}_\perp$, and q_{G_r} and q_{G_k} quantum queries to the random oracles G_r and G_k respectively.

The structure of the proof is very similar to that of Theorem 7. Basically we do the same sequence of game-hops as in the proof of Theorem 7 until the point where we can simulate the decapsulation oracles $\textsc{Decaps}_\perp(i,\cdot)$ $(i \in \{0,1\})$ without requiring the corresponding secret keys sk'_i. In the final game-hop, we reset G_r to be an ideal random oracle.

To be specific, we do the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_5$ as described in Figure 4.13. By a similar analysis as that of the proof of Theorem 7 w.r.t. these game-hops, it is not hard to obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_5 = 1]| \le q_D \cdot \mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 2q_{G_r}(q_D+2)\sqrt{2\delta}$$

Note that the game G_0 is exactly the SCFR-CCA game for $\mathsf{KEM}^{\not\perp}$. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A})$$

Games G_0 - G_5

```
$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
$G_r \leftarrow\!\!\$\ \Omega_{\mathbf{G}_r}$
$G_r^{\text{good}} \leftarrow \Omega_{\mathbf{G}_r}$   // Sampling distribution
        // described in the proof below.
$G_r := G_r^{\text{good}}$   // $\mathsf{G}_2$ - $\mathsf{G}_4$
$G_{0k}^{\text{acc}}, G_{1k}^{\text{acc}}, G_{0k}^{\text{rej}}, G_{1k}^{\text{rej}} \leftarrow\!\!\$\ \Omega_{\mathbf{G}_k}$
$G_{2k} \leftarrow\!\!\$\ \Omega_{\mathbf{G}'_k}$
$\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1)$
$c \leftarrow \mathcal{A}^{G_r, G_k, \textsc{Decaps}_\perp}(\mathsf{inp})$
$k_0 := \textsc{Decaps}_\perp(0, c)$
$k_1 := \textsc{Decaps}_\perp(1, c)$
return $[k_0 = k_1 \neq \perp]$
```

$G_k(m, c)$

```
if $\mathsf{Enc}(\mathsf{pk}_0, m; G_r(m)) = c$   // $\mathsf{G}_3$ - $\mathsf{G}_5$
    return $G_{0k}^{\text{acc}}(c)$   // $\mathsf{G}_3$ - $\mathsf{G}_5$
if $\mathsf{Enc}(\mathsf{pk}_1, m; G_r(m)) = c$   // $\mathsf{G}_3$ - $\mathsf{G}_5$
    return $G_{1k}^{\text{acc}}(c)$   // $\mathsf{G}_3$ - $\mathsf{G}_5$
return $G_{2k}(m, c)$
```

$\textsc{Decaps}_a(0, c)$ // $c \neq a$

```
return $G_{0k}^{\text{acc}}(c)$   // $\mathsf{G}_{3.5}$ - $\mathsf{G}_5$
Parse $\mathsf{sk}'_0 = (\mathsf{sk}_0, s_0)$
$m' := \mathsf{Dec}(\mathsf{sk}_0, c)$
if $\mathsf{Enc}(\mathsf{pk}_0, m'; G_r(m')) = c$ then
    return $G_k(m', c)$
else return $G_k(s_0, c)$   // $\mathsf{G}_0$
else return $G_{0k}^{\text{rej}}(c)$   // $\mathsf{G}_{0.5}$ - $\mathsf{G}_3$
```

$\textsc{Decaps}_a(1, c)$ // $c \neq a$

```
return $G_{1k}^{\text{acc}}(c)$   // $\mathsf{G}_4 - \mathsf{G}_5$
Parse $\mathsf{sk}'_1 = (\mathsf{sk}_1, s_1)$
$m' := \mathsf{Dec}(\mathsf{sk}_1, c)$
if $\mathsf{Enc}(\mathsf{pk}_1, m'; G_r(m')) = c$ then
    return $G_k(m', c)$
else return $G_k(s_1, c)$   // $\mathsf{G}_0$ - $\mathsf{G}_{0.5}$
else return $G_{1k}^{\text{rej}}(c)$   // $\mathsf{G}_1$ - $\mathsf{G}_{3.5}$
```

FIGURE 4.13: Games $\mathsf{G}_0 - \mathsf{G}_5$ for the proof of Theorem 8.

Coming to the game G_5, note that the adversary $\mathcal{A}$ wins the game if it outputs a ciphertext c such that $\textsc{Decaps}_\perp(0, c) = \textsc{Decaps}_\perp(1, c)$. Because of the modification of the $\textsc{Decaps}_\perp(i, \cdot)$ oracles, this winning condition translates to $G^{\mathrm{acc}}_{0k}(c) = G^{\mathrm{acc}}_{1k}(c)$, where G^{acc}_{0k} and G^{acc}_{1k} are independent quantum-accessible random oracles. Note that in this case, (c, c) is a *claw* w.r.t. the pair of QROs $G^{\mathrm{acc}}_{0k} : \mathcal{C} \to \mathcal{K}$ and $G^{\mathrm{acc}}_{1k} : \mathcal{C} \to \mathcal{K}$. Hence we can bound the success probability of $\mathcal{A}$ in G_5 by the advantage of an adversary $\mathcal{B}'$ against the *claw-finding* game w.r.t. the instance (G^{acc}_{0k}, G^{acc}_{1k}). $\mathcal{B}'$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_5.

- Uses a $2q_{G_r}$-wise independent function and a $2q_{G_k}$-wise independent function to perfectly simulate the random oracles G_r and G_{2k} in $\mathcal{A}$'s view, as noted in Lemma 1. Also uses the pair of oracles $f_0 : \mathcal{C} \to \mathcal{K}$ and $f_1 : \mathcal{C} \to \mathcal{K}$ – which is the instance of the claw-finding game – to simulate the oracles G^{acc}_{0k} and G^{acc}_{1k} respectively.

- Answers decapsulation queries using the oracles $f_i(\cdot)$ $(i \in \{0, 1\})$ as in G_4.

- After obtaining a ciphertext c from $\mathcal{A}$, forwards (c, c) as the claw w.r.t. (f_0, f_1).

Note that $\mathcal{B}'$ makes at most q_{G_k} queries to the pair (f_0, f_1). It is easy to see that $\Pr[\mathsf{G}_5 = 1] \leq \frac{C(q_H+1)^3}{|\mathcal{K}|}$ from Lemma 11. Hence, we finally get

$$\mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq q_D \cdot \mathbf{Adv}^{\mathsf{SCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{C(q_{G_k}+1)^3}{|\mathcal{K}|} + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 2q_{G_r}(q_D+2)\sqrt{2\delta}\,.$$

□

From Theorems 7 and 8, we see that by applying the $\mathsf{FO}^{\not\perp}$ transformation to weakly secure (i.e., OW-CPA) and weakly anonymous (i.e., wANO-CPA) PKE schemes, with an additional assumption of strong collision freeness against chosen plaintext attacks of the deterministic version of the underlying PKE scheme (PKE_1 = $\mathsf{T}[\mathsf{PKE}, G_r]$), not only do we obtain strongly secure KEMs (i.e., IND-CCA security) but also KEMs that are strongly anonymous (i.e., ANO-CCA) and are strongly collision-free against chosen ciphertext attacks (SCFR-CCA) in the QROM.

At the same time, we showed a negative result in Theorem 6. It essentially shows that starting with a KEM that is IND-CCA, ANO-CCA and SCFR-CCA secure does not *generically* result in a strongly anonymous (ANO-CCA) hybrid PKE scheme via the KEM-DEM composition. Nonetheless, we are able to show the following positive result for KEMs obtained via the $\mathsf{FO}^{\not\perp}$ transform. We only need a weak additional property of the underlying PKE scheme, namely that it be γ-spread (see Definition 3).

Theorem 9. *Let* $\mathsf{PKE}^{\mathrm{hy}} = (\mathsf{KGen}^{\mathrm{hy}}, \mathsf{Enc}^{\mathrm{hy}}, \mathsf{Dec}^{\mathrm{hy}})$ *be a hybrid PKE scheme obtained by composing* $\mathsf{KEM}^{\not\perp} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *with a one-time secure AE scheme* $\mathsf{DEM} = (\mathsf{KGen}^{\mathrm{dem}}, \mathsf{Enc}^{\mathrm{dem}}, \mathsf{Dec}^{\mathrm{dem}})$*. Suppose the base* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *underlying* $\mathsf{KEM}^{\not\perp}$ *is δ-correct and γ-spread (with message space $\mathcal{M}$). Then for any* ANO-CCA *adversary* $\mathcal{A}_{\mathrm{hy}}$ *against* $\mathsf{PKE}^{\mathrm{hy}}$ *issuing at most* q_{G_r} *and* q_{G_k} *queries to the quantum random oracles* G_r *and* G_k *respectively, there exist* ANO-CCA *adversary* $\mathcal{A}_{\mathrm{kem}}$ *and* IND-CCA *adversary* $\overline{\mathcal{A}}_{\mathrm{kem}}$ *against* $\mathsf{KEM}^{\not\perp}$*,* WCFR-CPA *adversary* $\mathcal{B}$ *against* $\mathsf{PKE}_1 = (\mathsf{KGen}, \mathsf{Enc}', \mathsf{Dec}')$*, and* INT-CTXT *adversary* $\mathcal{A}_{\mathrm{dem}}$ *against* DEM *such that:*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}_{\mathrm{kem}}) + 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\overline{\mathcal{A}}_{\mathrm{kem}}) + 2^{-\gamma} \\ + \mathbf{Adv}^{\mathsf{WCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + 2\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 4q_{G_r}\sqrt{\delta}\,.\end{aligned}$$

Moreover, the running times of $\mathcal{A}_{\mathrm{kem}}$*,* $\overline{\mathcal{A}}_{\mathrm{kem}}$ *and* $\mathcal{A}_{\mathrm{dem}}$ *are the same as that of* $\mathcal{A}_{\mathrm{hy}}$*. The running time of* $\mathcal{B}$ *is independent (and less than that) of the running time of* $\mathcal{A}_{\mathrm{hy}}$*.*

Proof. The structure of the proof is quite similar to that of Theorem 3.1, except for some initial game-hops. Here we will focus on these hops.

Denote $\Omega_{\mathbf{G}_r}$, $\Omega_{\mathbf{G}_k}$ and $\Omega_{\mathbf{G}'_k}$ to be the set of all functions $\mathbf{G}_r : \mathcal{M} \to \mathcal{R}$, $\mathbf{G}_k : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ and $\mathbf{G}'_k : \mathcal{C} \to \mathcal{K}$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{K}$ is the encapsulated key space of $\mathsf{KEM}^{\not\perp}$ and $\mathcal{C}$ is the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$. Let $\mathcal{A}_{\mathrm{hy}}$ be an adversary in the ANO-CCA game for $\mathsf{PKE}^{\mathrm{hy}}$ issuing at most q_{G_r} and q_{G_k} quantum queries to the random oracles G_r and G_k respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_6$ described in Figure 4.14.

Game G_0**:** The game G_0 is equivalent to the ANO-CCA game for $\mathsf{PKE}^{\mathrm{hy}}$ (the only "cosmetic" change is that the uniform random bit b is sampled before the adversary $\mathcal{A}_{\mathrm{hy}}$ gets to choose a message m^{hy}). Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}).$$

```
Games G_0 - G_5
    $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
    $s_0 \leftarrow\$ \mathcal{M}; s_1 \leftarrow\$ \mathcal{M}$
    $\mathsf{sk}'_0 := (\mathsf{sk}_0, s_0), \mathsf{sk}'_1 := (\mathsf{sk}_1, s_1)$
    $G_r \leftarrow\$ \Omega_{\mathbf{G}_r}; G_k \leftarrow\$ \Omega_{\mathbf{G}_k}; G'_k \leftarrow\$ \Omega_{\mathbf{G}'_k}$
    $b \leftarrow\$ \{0,1\}$
    $G_r^{\text{good}} \leftarrow \Omega_{\mathbf{G}_r}$   // Sampling distribution
        // described in description of $\mathsf{G}_{0.3}$ above.
    $G_r := G_r^{\text{good}}$   // $\mathsf{G}_{0.3} - \mathsf{G}_{0.6}$
    $m^* \leftarrow\$ \mathcal{M}$   // $\mathsf{G}_{0.3} - \mathsf{G}_5$
    $c_0^* := \mathsf{Enc}(\mathsf{pk}_b, m^*; G_r(m^*))$   // $\mathsf{G}_{0.3} - \mathsf{G}_5$
    $k^* \leftarrow G_k(m^*, c_0^*)$   // $\mathsf{G}_{0.3} - \mathsf{G}_5$
    $k^{\mathsf{rej}} \leftarrow G_k(s_{1-b}, c_0^*)$   // $\mathsf{G}_2$
    $k^{\mathsf{rej}} \leftarrow G'_k(c_0^*)$   // $\mathsf{G}_3 - \mathsf{G}_4$
    $(m^{\text{hy}}, \mathsf{st}) \leftarrow \mathcal{A}_{\text{hy}}^{G_r, G_k, \textsc{Dec}_{\perp}^{\text{hy}}}(\mathsf{pk}_0, \mathsf{pk}_1)$
    $m^* \leftarrow\$ \mathcal{M}$   // $\mathsf{G}_0$
    $c_0^* := \mathsf{Enc}(\mathsf{pk}_b, m^*; G_r(m^*))$   // $\mathsf{G}_0$
    $k^* \leftarrow G_k(m^*, c_0^*)$   // $\mathsf{G}_0$
    $c_1^* \leftarrow \mathsf{Enc}^{\text{dem}}(k^*, m^{\text{hy}})$   // $\mathsf{G}_0 - \mathsf{G}_5$
    $c^* = (c_0^*, c_1^*)$
    $b' \leftarrow \mathcal{A}_{\text{hy}}^{G_r, G_k, \textsc{Dec}_{c^*}^{\text{hy}}}(c^*, \mathsf{st})$
    return $[b' = b]$
```

```
$\textsc{Dec}_a^{\text{hy}}(b, c)$   // $c \neq a$
    Parse $c = (c_0, c_1)$
    Parse $\mathsf{sk}'_b = (\mathsf{sk}_b, s_b)$
    if $c_0 = c_0^*$   // $\mathsf{G}_{0.6} - \mathsf{G}_5$
        $k' := k^*$   // $\mathsf{G}_{0.6} - \mathsf{G}_5$
    else   // $\mathsf{G}_{0.6} - \mathsf{G}_5$
        $m' := \mathsf{Dec}(\mathsf{sk}_b, c_0)$
        if $\mathsf{Enc}(\mathsf{pk}_b, m'; G_r(m')) = c_0$
            $k' \leftarrow G_k(m', c_0)$
        else $k' \leftarrow G_k(s_b, c_0)$
    $m^{\text{hy}'} := \mathsf{Dec}^{\text{dem}}(k', c_1)$
    return $m^{\text{hy}'}$
```

```
$\textsc{Dec}_a^{\text{hy}}(1 - b, c)$   // $c \neq a$
    Parse $c = (c_0, c_1)$
    Parse $\mathsf{sk}'_{1-b} = (\mathsf{sk}_{1-b}, s_{1-b})$
    if $c_0 = c_0^*$   // $\mathsf{G}_2 - \mathsf{G}_5$
        $k' := k^{\mathsf{rej}}$   // $\mathsf{G}_2 - \mathsf{G}_3$
        return $\perp$   // $\mathsf{G}_4 - \mathsf{G}_5$
    else   // $\mathsf{G}_{0.6} - \mathsf{G}_5$
        $m' := \mathsf{Dec}(\mathsf{sk}_{1-b}, c_0)$
        if $\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G_r(m')) = c_0$
            $k' \leftarrow G_k(m', c_0)$
        else $k' \leftarrow G'_k(c_0)$   // $\mathsf{G}_3 - \mathsf{G}_4$
        else $k' \leftarrow G_k(s_{1-b}, c_0)$
    $m^{\text{hy}'} := \mathsf{Dec}^{\text{dem}}(k', c_1)$
    return $m^{\text{hy}'}$
```

FIGURE 4.14: Games G_0 – G_5 for the proof of Theorem 9.

Game $\mathsf{G}_{0.3}$**:** In game $\mathsf{G}_{0.3}$, we first make some "cosmetic" changes. Namely, the pair (c_0^*, k^*) resulting from running $\mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b is generated *before* the adversary $\mathcal{A}_{\text{hy}}$ gets to choose a message m^{hy}. This change does not affect $\mathcal{A}_{\text{hy}}$'s view in any way.

Next, we change the random oracle G_r such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}_b, \mathsf{sk}_b)$, as seen in the proof of Theorem 7. Specifically, define the oracle $G_r^{\text{good}} \leftarrow \Omega_{\mathbf{G}_r}$ such that $G_r^{\text{good}}(m)$ is sampled according to a uniform distribution in $\mathcal{R}_{\text{good}}((\mathsf{pk}_b, \mathsf{sk}_b), m)$. Hence in $\mathsf{G}_{0.3}$, we replace the oracle G_r with G_r^{good}. By using a similar analysis as the game-hop "$\mathsf{G}_1 \to \mathsf{G}_2$" in the proof of Theorem 7 (in fact, the analysis would be simpler in this case since we have to consider a single key-pair $(\mathsf{pk}_b, \mathsf{sk}_b)$ instead of two), it is not hard to obtain

$$|\Pr[\mathsf{G}_{0.3} = 1] - \Pr[\mathsf{G}_0 = 1]| \leq 2q_{G_r}\sqrt{\delta}.$$

Game $\mathsf{G}_{0.6}$**:** In game $\mathsf{G}_{0.6}$, we modify the oracle $\textsc{Dec}_a^{\text{hy}}(b, \cdot)$ (with $a \in \{\bot, c^*\}$) such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$ (and $c_1 \neq c_1^*$), then the oracle uses k^* to decrypt c_1, instead of first decapsulating c_0^* to recover a session key k'. It is not hard to see that the games G_0 and G_1 are equivalent since G_r samples good random coins, and hence, there is no decapsulation error w.r.t. $\mathsf{KEM}^{\not\perp}$. Therefore, we have

$$\Pr[\mathsf{G}_{0.6} = 1] = \Pr[\mathsf{G}_{0.3} = 1].$$

Game G_1**:** In game G_1, we reset G_r to be an ideal random oracle, i.e., $G_r(m)$ now returns uniformly random coins from $\mathcal{R}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_0 \to \mathsf{G}_{0.3}$, by using a similar analysis, it is not hard to obtain

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_{0.6} = 1]| \leq 2q_{G_r}\sqrt{\delta}.$$

Game G_2**:** In game G_2, we modify the oracle $\textsc{Dec}_a^{\text{hy}}(1-b, \cdot)$ such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$, then the oracle uses $k^{\mathsf{rej}}(= G_k(s_{1-b}, c_0^*))$ to decrypt c_1. Here k^{rej} is the key returned if $\mathsf{Decap}(\mathsf{sk}'_{1-b}, c_0^*)$ would have resulted in an "implicit rejection". Thus, it is not hard to see that the games G_1 and G_2 are equivalent unless c_0^* is not (implicitly) rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation, or in other words, if the following event occurs: $\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G_r(m')) = c_0^*$ where $\mathsf{Enc}(\mathsf{pk}_b, m^*; G_r(m^*)) = c_0^*$ and $\mathsf{Dec}(\mathsf{sk}_{1-b}, c_0^*) = m'$ (for $m^* \leftarrow\$ \mathcal{M}$).

There are two sub-events to consider w.r.t. the above event:

1. $\underline{m' \neq m^*}$: In this case, the random oracle G_r on a new query m' will return uniformly random coins $r \leftarrow\!\!\$\ \mathcal{R}$. Since PKE is γ-spread, for the key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b})$ and message m', we have the re-encryption check, namely "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r) = c_0^*$", to hold with probability $\leq 2^{-\gamma}$, for uniformly random r.

2. $\underline{m' = m^*}$: In this case, we can bound the probability of the sub-event occurring by the advantage of an adversary $\mathcal{B}$ in the WCFR-CPA game of PKE_1 $(= \mathsf{T}[\mathsf{PKE}, G_r])$. The adversary $\mathcal{B}$, upon receiving public-keys pk_0 and pk_1, simply samples a bit b and message m^* uniformly at random, i.e., $b \leftarrow\!\!\$\ \{0,1\}$ and $m^* \leftarrow\!\!\$\ \mathcal{M}$, and returns (m, b) to the WCFR-CPA challenger (note that only a single query is made to G_r on m^* in the security experiment).

Hence,

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathbf{Adv}^{\mathsf{WCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + 2^{-\gamma}.$$

Note that for the ANO-CCA security of $\mathsf{KEM}^{\not\perp}$, we anyway rely on the SCFR-CPA security of the deterministic PKE_1.

Game G_3**:** In game G_3, we modify the decryption oracle $\textsc{Dec}_a^{\mathrm{hy}}(1-b, \cdot)$ such that the key $G'_k(c_0)$ is used to decrypt the DEM ciphertext c_1 instead of $G_k(s_{1-b}, c_0)$ where the KEM ciphertext c_0 was implicitly rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation; G'_k is an internal random oracle not directly accessible by the adversary $\mathcal{A}_{\mathrm{hy}}$. We also generate the key k^{rej} as "$k^{\mathsf{rej}} \leftarrow G'_k(c_0^*)$" (instead of "$k^{\mathsf{rej}} \leftarrow G_k(s_{1-b}, c_0^*)$").

Here we can use Lemma 2 w.r.t. pseudorandomness of the QRO $G_k(s_{1-b}, \cdot)$, with PRF key $s_{1-b} \leftarrow\!\!\$\ \mathcal{M}$, to obtain the following via a straightforward reduction:

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_{G_k}}{\sqrt{|\mathcal{M}|}}.$$

Game G_4**:** In game G_4, we modify the oracle $\textsc{Dec}_a^{\mathrm{hy}}(1-b, \cdot)$ such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$, then the oracle returns $\perp$. It is not hard to see that the games G_3 and G_4 are equivalent unless the following event occurs: $\mathcal{A}_{\mathrm{hy}}$ makes a decryption query (c_0^*, c_1) to the oracle $\textsc{Dec}_a^{\mathrm{hy}}(1-b, \cdot)$ such that $\mathsf{Dec}^{\mathrm{dem}}(k^{\mathsf{rej}}, c_1) \neq \perp$. And we can bound the probability of this event occurring by the advantage of an INT-CTXT adversary $\mathcal{A}_{\mathrm{dem}}$ against DEM, again via a straightforward reduction.

Note that in games G_3 and G_4, the internal random oracle G'_k is never queried on c_0^* (particularly, in the $\textsc{Dec}_a^{\mathrm{hy}}(1-b, \cdot)$ oracle) except for defining

$k^{\mathsf{rej}}(\leftarrow G'_k(c_0^*))$ in the setup. This is equivalent to having k^{rej} be a uniformly random key independent of the oracle G'_k, i.e., $k^{\mathsf{rej}} \leftarrow\!\!\$\ \mathcal{K}$. Hence in the INT-CTXT game of DEM, we can implicitly define k^{rej} to be the random secret key chosen by the challenger. The adversary $\mathcal{A}_{\text{dem}}$ then proceeds by first sampling a bit $b \leftarrow\!\!\$\ \{0,1\}$ and locally generating the key pairs $(\mathsf{pk}_b, \mathsf{sk}_b), (\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}$. It then simulates the game G_4 towards $\mathcal{A}_{\text{hy}}$ by generating the intermediate values (e.g., s_b, m^*, c_0^*, k^*, etc.) as in Fig. 4.14 and simulating the decryption oracles $\textsc{Dec}_a^{\text{hy}}$ oracles using the secret keys sk_b, sk_{1-b}.[9] The main thing that is relevant for the reduction is, when $\mathcal{A}_{\text{hy}}$ makes a query (c_0^*, c_1) to $\textsc{Dec}_a^{\text{hy}}(1-b, \cdot)$, $\mathcal{A}_{\text{dem}}$ forwards c_1 to its own decryption oracle in the INT-CTXT game w.r.t. DEM. If the aforementioned event occurs (i.e., $\mathsf{Dec}^{\text{dem}}(k^{\mathsf{rej}}, c_1) \neq \perp$), then $\mathcal{A}_{\text{dem}}$ wins its game (also note that, $\mathcal{A}_{\text{dem}}$ has no need to make any encryption oracle queries in the INT-CTXT game in order to simulate G_4 towards $\mathcal{A}_{\text{hy}}$). Hence, we have

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Adv}_{\mathsf{DEM}}^{\mathsf{INT\text{-}CTXT}}(\mathcal{A}_{\text{dem}}).$$

Game G_5 In game G_5, we (re-)modify the decryption oracle $\textsc{Dec}_a^{\text{hy}}(1-b, \cdot)$ such that the key $G_k(s_{1-b}, c_0)$ is used to decrypt the DEM ciphertext c_1 instead of $G'_k(c_0)$ where the KEM ciphertext c_0 was implicitly rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation. In a sense, we are reverting the changes introduced in the $\mathsf{G}_2 \to \mathsf{G}_3$ hop. Hence, by using a similar analysis as that hop (and note that now, the key k^{rej} is not used anymore), it is not hard to obtain

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2q_{G_k}}{\sqrt{|\mathcal{M}|}}$$

Compared to the proof of Theorem 3.1, we have effectively used the sequence of games $\mathsf{G}_0 - \mathsf{G}_5$ to arrive at a point where we modified the oracle $\textsc{Dec}_a^{\text{hy}}(1-b, \cdot)$ such that if the decryption query is (c_0^*, c_1), the oracle returns $\perp$; this particular point is the *hybrid* game "G_2" in the proof of

[9] $\mathcal{A}_{\text{dem}}$ uses a $2q_{G_r}$-wise and a $2q_{G_k}$-wise independent function to simulate the QROs G_r and G_k respectively. It also simulates the internal random oracle G'_k *classically* towards $\mathcal{A}_{\text{hy}}$, e.g., via "lazy sampling", since G'_k is only used to process classical queries c_0 in the game G_4; recall that we consider only *classical* decryption queries in the QROM.

Theorem 3.1. Now doing a similar sequence of game-hops from that point on, namely "$\mathsf{G}_2 \rightarrow \mathsf{G}_4$", in the current setting starting from G_5, we arrive at

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}_{\mathrm{kem}}) + 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}^{\not\perp}}(\overline{\mathcal{A}}_{\mathrm{kem}}) + 2^{-\gamma} \\ + \mathbf{Adv}^{\mathsf{WCFR\text{-}CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + 2\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) + \frac{4q_{G_k}}{\sqrt{|\mathcal{M}|}} + 4q_{G_r}\sqrt{\delta}\,.\end{aligned}$$

□

4.5.1 *Extension to* $\mathsf{FO}^{\not\perp}_m$

In this section, we have mainly focused on the $\mathsf{FO}^{\not\perp}$ transform. Coming to $\mathsf{FO}^{\not\perp}_m$, the only difference between this transform and $\mathsf{FO}^{\not\perp}$ is that, in the former, the encapsulated keys are computed as "$\overline{k} \leftarrow G_k(m)$" – in contrast to "$\overline{k} \leftarrow G_k(m,c)$" done in the latter (see Figures 3.1 and 3.2). But this is a significant difference in the context of extending our anonymity and robustness enhancing results of $\mathsf{FO}^{\not\perp}$ above to $\mathsf{FO}^{\not\perp}_m$. The reason, at a very high level, is that in our security proofs on $\mathsf{FO}^{\not\perp}$ – e.g., Theorem 7 on ANO-CCA security – we needed to simulate *two* different decapsulation oracles without possessing the corresponding secret keys. And crucially, we used the pair (m,c) as a "handle" in our proofs to answer the decapsulation queries, made out to different secret-keys, in a consistent way. The lack of this feature in $\mathsf{FO}^{\not\perp}_m$ leads us to believe that our results above do not extend to this transform in a straightforward manner.

However, in subsequent work, Xagawa [60] was able to establish anonymity (and robustness) enhancing properties of $\mathsf{FO}^{\not\perp}_m$ in the QROM using an alternative proof strategy. Instead of *directly* analyzing the ANO-CCA security of $\mathsf{FO}^{\not\perp}_m$-derived KEMs and hybrid PKE schemes, he considered a stronger security notion called *strong pseudorandomness* (or, SPR-CCA security). A KEM[10] is said to be SPR-CCA secure if, roughly speaking, an adversary cannot distinguish a *real* ciphertext/encapsulated-key pair (c^*, k^*) from a *random* pair (c', k') where c' is a random ciphertext and k' is a random key (see Subsection 5.3.1 for a formal definition of SPR-CCA security where we also need to consider a *simulator* to specify what we mean by a "random" ciphertext c'). Moreover, it was shown in [60] that SPR-CCA security straightforwardly implies ANO-CCA security.

Now the key insight used by Xagawa in [60] is that since SPR-CCA security is a "single key-pair" notion (i.e., a single key-pair is generated by

[10] The analogous definition for PKE schemes follows straightforwardly.

the challenger in its security game) unlike the "double key-pair" ANO-CCA notion, security proofs corresponding to the former notion would involve simulating a *single* decapsulation oracle – as opposed to *two* decapsulation oracles w.r.t. the latter notion – which in turn precludes the use of any "handle" as discussed above. Hence as a consequence, he was also able to establish anonymity and robustness of *NTRU* [23] – a third-round NIST PQC finalist – in the QROM, since the scheme uses $\mathsf{FO}_m^{\not\perp}$ in its KEM construction. Looking ahead, we will use Xagawa's framework to establish post-quantum anonymity and robustness of Kyber in the next chapter.

4.6 SUMMARY

In this chapter, we provided a generic analysis of anonymity and robustness – two important "beyond IND-CCA" properties – for PKE schemes built via the KEM-DEM paradigm. We first presented general security definitions of anonymity and robustness for the KEM primitive, which we hope will be of independent interest. Then we divided our above generic analysis depending on whether the underlying KEM, in the KEM-DEM composition, offered implicit rejection or explicit rejection. We showed that implicit rejection KEMs, in general, do not transfer their anonymity and robustness to the corresponding KEM-DEM hybrid PKE schemes; on the other hand, explicit rejection KEMs do transfer the above properties.

Shifting our attention to NIST's PQC standardization process, we note that most candidate KEMs considered there offer implicit rejection while using variants of the Fujisaki-Okamoto transform in their construction. Hence, we analyzed an implicitly-rejecting variant – namely, the $\mathsf{FO}^{\not\perp}$ transform – with respect to the anonymity and robustness properties, and showed that $\mathsf{FO}^{\not\perp}$ does confer these properties to the constructed KEM; notably, our analysis is in the QROM, in line with the post-quantum setting considered in this thesis. We also showed that such a class of implicit rejection KEMs obtained via the $\mathsf{FO}^{\not\perp}$ transform does transfer anonymity and robustness to the hybrid PKE schemes in the KEM-DEM composition – thereby overcoming the above generic impossibility result.

5

ANONYMITY AND ROBUST ENHANCEMENTS, PART II: APPLICATION TO NIST PQC CANDIDATES

After motivating the importance of anonymity and robustness for public-key encryption schemes in the previous chapter, we now apply our above generic analysis for implicit rejection KEMs to specific schemes related to the NIST PQC standardization process which employ FO-type transforms that are variants of $\mathsf{FO}^{\not\perp}$. In particular, we focus on the NIST fourth-round candidate *Classic McEliece* [21], the NIST third-round alternate candidate *FrodoKEM* [16], and the current NIST PQC standard *Kyber* [11]. It is worth pointing out that the first two schemes are currently also recommended by the German federal agency BSI for usage in the post-quantum setting [17].

To provide an overview of our results, for Classic McEliece, we show that the hybrid PKE resulting from applying the standard KEM-DEM paradigm is not strongly robust (i.e., SROB secure). In fact, we can show that, for any plaintext m, it is possible to construct a single ciphertext c such that c always decrypts to m under *any* Classic McEliece private key. The construction of c does not even need the public key! The complete details of our "robustness attack" is presented in Section 5.1. But we stress that our attack does not indicate any problem with IND-CCA security of Classic McEliece, but it does expose its limitations as a general-purpose KEM for the broad set of applications that can be envisaged for NIST public key algorithms. Since our $\mathsf{FO}^{\not\perp}$-related results in Chapter 4 on anonymity of KEMs and PKE schemes built from them depend on robustness properties, Classic McEliece's limitations in this regard present a barrier to establishing its anonymity using our techniques (but do not preclude other proof techniques – e.g., the ones used by Xagawa to *directly* prove anonymity of Classic McEliece in [60]).

Coming to FrodoKEM, the news is better. We provide positive results on anonymity and robustness properties of its KEM and the hybrid PKE schemes derived from it. Towards these results, we have to adapt our generic analysis on $\mathsf{FO}^{\not\perp}$ to the actual transform used by FrodoKEM. To be more specific, FrodoKEM uses an FO-type transform – namely, $\mathsf{FO}^{\mathrm{frodo}}$ – which differs significantly from $\mathsf{FO}^{\not\perp}$. And as argued in Chapter 3, these differences invalidate direct application of known IND-CCA security results on the standard $\mathsf{FO}^{\not\perp}$ transform to $\mathsf{FO}^{\mathrm{frodo}}$ in the QROM. Despite this, we

```
$\mathsf{KGen}'$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$s \leftarrow\$ \mathbb{F}_2^n$
$\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}, s)$
return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Encap}(\mathsf{pk})$
$m \leftarrow \mathsf{FixedWeight}()$
$c := \mathsf{Enc}(\mathsf{pk}, m)$
$k \leftarrow H_1(m, c)$
return $(c, k)$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
if $m' \neq \perp$ then
    return $H_1(m', c)$
else return $H_0(s, c)$
```

FIGURE 5.1: The $\mathsf{FO}^{\mathsf{cm}}$ transform used by Classic McEliece (CM). Here we have $(\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ describing the CM KEM and $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ describing the base PKE scheme (also referred to as "the one-way function" in [21]). The algorithm $\mathsf{FixedWeight}()$ outputs a uniformly random n-bit vector $m \in \mathbb{F}_2^n$ with a fixed Hamming weight t (see [21] for a formal description); here t is a CM parameter. We also have hash functions H_0 and H_1 with 256-bit outputs.

were able to re-establish post-quantum IND-CCA security of FrodoKEM in that chapter using an alternative approach. As will be seen in Section 5.2, a similar approach also allows us to establish anonymity and robustness properties for FrodoKEM in the QROM.

Finally, for Kyber, we were also able to establish post-quantum anonymity and robustness of the new NIST PQC standard as well as the hybrid PKE schemes derived from it. To prove anonymity in particular, instead of adapting our generic analysis on $\mathsf{FO}^{\not\perp}$ in the previous chapter to Kyber, we work with Xagawa's proof techniques that were used to establish anonymity of Classic McEliece, NTRU [23] and other important NIST PQC KEMs in [60]. To provide some more context, in the same paper [60], Xagawa was unable to extend his techniques to show anonymity of Kyber because of the fact that the FO-type transform used by Kyber (i.e., $\mathsf{FO}^{\mathsf{kyber}}$) hashes some additional intermediate values in its internal computations when compared to the standard FO transforms of [4]. At a high level, we salvage Xagawa's proof strategy by adapting our "wrapper-based" approach of Chapter 3 which was used to establish IND-CCA security of Kyber in the QROM. The formal details are presented in Section 5.3. Coming to robustness, our positive analysis of Kyber is quite similar to that of FrodoKEM, which in-turn is based on our generic analysis of "$\mathsf{FO}^{\not\perp}$-derived" KEMs in Chapter 4.

5.1 CLASSIC MCELIECE

Classic McEliece (CM) is a code-based KEM which relies on one-wayness (i.e., OW-CPA security) of the so-called *McEliece cryptosystem* [82] for its post-quantum IND-CCA security. As defined in its fourth round NIST PQC specification [21], CM applies a slight variant of the $\mathsf{FO}^{\not\perp}$ transform, which we call $\mathsf{FO}^{\mathrm{cm}}$, to its starting *deterministic* base PKE scheme (see Fig. 5.1). The main difference between $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\mathrm{cm}}$ is that in the latter transform, there is no re-encryption check during decapsulation (see Lines 4 and 5 in "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 3.2). However, the base PKE scheme of CM *implicitly* performs this re-encryption check in its decryption routine (see Line 4 in [21, Section 4.4]). In particular, this means that our generic results on anonymity and robustness conferred by $\mathsf{FO}^{\not\perp}$ – namely, Theorems 7 and 8 – apply to $\mathsf{FO}^{\mathrm{cm}}$ in a straightforward manner.

Therefore, the only thing that would remain to be analyzed is whether the base PKE scheme used by CM satisfies the pre-requisite security properties of Theorems 7 and 8: namely, the notions of wANO-CPA and SCFR-CPA. As we show next, the base PKE scheme used by CM fails to be collision-free in a striking way that rules out the application of these results. This failure also propagates to the hybrid PKE schemes built from CM KEM via the standard KEM-DEM paradigm.

5.1.1 *Specification of the Base PKE Scheme*

As mentioned above, the base PKE scheme used by CM is deterministic. At a high level, to encrypt a message $m \in \mathbb{F}_2^n$, the scheme first encodes m as a binary column vector e of length n and fixed Hamming weight t. Then it computes ciphertext $c = He \in \mathbb{F}_2^{(n-k)}$ where we have the matrix $H = (I_{n-k} \,|\, T) \in \mathbb{F}_2^{(n-k)\times n}$ with $T \in \mathbb{F}_2^{(n-k)\times k}$ essentially being the public key;[1] also k is another CM parameter. More specifically, H is the parity check matrix of an error correcting code whose error correcting capacity is at least t. Decryption is done by using the private key to rewrite matrix H in such a way that efficient decoding can be performed to recover e with perfect correctness. In fact, the base PKE scheme of CM is closely related to the Niederreiter variant of the McEliece PKE scheme [83].

[1] Here I_{n-k} denotes the identity matrix of dimension $(n-k) \times (n-k)$.

5.1.2 *Collision-Freeness of the Base PKE Scheme*

Recall that we would require the base PKE scheme of CM to satisfy the SCFR-CPA property in order to make use of our generic results concerning the $\mathsf{FO}^{\not\perp}$ transform. This property is crucial in the CPA $\rightarrow$ CCA security proofs where we have to simulate the decapsulation oracles under two different secret keys without access to the keys. As we will show now, the base PKE scheme is not SCFR-CPA secure, nor even WCFR-CPA secure. In fact, we can go further and exhibit a strong robustness failure of the base PKE scheme, and explain how it leads to robustness failures in the CM KEM and hybrid PKE schemes built from it.

Consider any error vector e with Hamming weight t in which the t 1's in e are concentrated in the first $n-k$ bit positions of e (in all the parameter sets used in Classic McEliece, $n-k = mt \geq t$, for a positive integer m, so this is always possible). We call such an e *concentrated*. Note that any concentrated e can be written as $e = \binom{e_{n-k}}{0_k}$ with e_{n-k} of length $n-k$ and 0_k being the vector of k zeros. Since encryption is done by computing $c = He$, and H is of the form $(I_{n-k} \mid T)$, it is easy to see that c is a fixed vector independent of the T component of H: namely, $He = e_{n-k}$ which depends only on the first $n-k$ bit positions of e.

Note that this property holds independent of the public key of the base PKE scheme (i.e., the matrix T). Thus there is a class of messages of the base PKE scheme (of size $\binom{n-k}{t}$) for which the resulting ciphertext c can be predicted as a function of the message *without even knowing the public key*. By correctness of the base PKE scheme of CM, such ciphertexts must decrypt to the selected message *under any private key of the scheme*. Hence, it is immediate that this property can be used to violate SCFR-CPA and WCFR-CPA security of the base PKE scheme.

5.1.3 *Robustness of Classic McEliece and Corresponding Hybrid PKE Schemes*

The base PKE scheme is used to construct the CM KEM according to procedure described in Figure 5.1. This means that the CM KEM encapsulations are also of the form $c = He$; meanwhile the encapsulated keys are set as $k = H_1(e, c)$ where H_1 is a hash function. The CM KEM performs implicit rejection, so one cannot hope for robustness. However, one might hope for some form of collision-freeness. Our analysis above shows that the CM KEM does not provide even this, since when e is concentrated, $c = He$

decapsulates to $H_1(e, c)$ under any CM private key. Hence, technically the CM KEM is not SCFR-CPA secure.[2]

Finally, one might ask about the robustness of hybrid PKE schemes built by combining the CM KEM with a DEM using the standard KEM-DEM paradigm. Again, such a PKE scheme cannot be strongly collision free (and therefore not strongly robust either), since it is trivial using our observations to construct a hybrid PKE ciphertext that decrypts correctly under *any* CM private key to *any* fixed choice of message m (without even knowing the public key). To see this, simply consider hybrid ciphertexts of the form $(c_{\text{kem}}, c_{\text{dem}}) = (He, \mathsf{Enc}^{\text{dem}}(k, m; r))$ where $e \in \mathbb{F}_2^n$ is concentrated, $k = H_1(e, c_{\text{kem}})$ is the symmetric key encapsulated by the CM KEM component $c_{\text{kem}} = He$ of the hybrid ciphertext, and r is some fixed randomness for the DEM scheme $(\mathsf{KGen}^{\text{dem}}, \mathsf{Enc}^{\text{dem}}, \mathsf{Dec}^{\text{dem}})$. It is not hard to see that such hybrid ciphertexts decrypt to the freely chosen message m under any CM private key.

Robustness could plausibly be conferred on this hybrid PKE scheme by including a hash of the public key in the key derivation step. However CM public keys are large, so this would have a negative effect on performance. Robustness is *not* conferred in general by replacing the DEM with an AEAD[3] scheme and including the hash of the public key in the associated data to create a "labelled DEM". This is easy to see by adapting the counter-example construction used in the proof of Theorem 5.

5.1.4 *Related Work*

The analysis above shows that we cannot hope to establish anonymity or robustness of the CM KEM or hybrid PKE schemes built from it via the standard KEM-DEM paradigm using the sequence of results in the previous chapter. But this does not rule out more direct approaches to proving anonymity. In fact, Xagawa [60] was able to establish anonymity (i.e., ANO-CCA security) of Classic McEliece[4] and its corresponding hybrid

[2] But it is plausible that the CM KEM is WCFR-CPA secure since in an honest execution of "$\mathsf{Encap}(\mathsf{pk})$" (see Fig. 5.1), the probability that $\mathsf{FixedWeight}()$ outputs a concentrated e is quite low; more precisely, the probability is $\binom{n-k}{t} / \binom{n}{t}$. However, we consider a formal WCFR-CPA security analysis of CM to be beyond the scope of this chapter.

[3] Authenticated Encryption with Associated Data; see [46] for a formal definition of this primitive.

[4] Technically, Xagawa analyzed the NIST PQC third-round specification of CM which included an additional "plaintext confirmation" hash in the ciphertext. Namely, encapsulations of third-round CM KEM were of the form $c = (He, H_2(e))$ where $H_2(e)$ is the plaintext confirmation hash; decapsulation then involved recomputing this hash and checking it against the input

PKE schemes in the QROM using the *strong pseudorandomness* framework discussed in Subsection 4.5.1. He also proposed a modification to Classic McEliece which makes it strongly collision free (i.e., SCFR-CCA secure) in the QROM.

But it is worth pointing out that Xagawa relied on assumptions (namely, the so-called *"modified PR-key"* and *"modified Decisional Syndrome Decoding"* assumptions [60]) which are technically different than the ones which CM relies on for its IND-CCA security. This is in contrast to our subsequent anonymity analysis of FrodoKEM and Kyber wherein we rely on the *same* assumptions as used by these schemes for their post-quantum IND-CCA security: namely, the LWE and MLWE hardness assumptions respectively.

Finally, coming back to Classic McEliece, we are in a situation where the hybrid PKE schemes derived from it are anonymous (as shown by Xagawa [60]) but *not* robust (as shown in this section). Since we argued earlier on in Chapter 4 that robustness is important to ensure basic communication correctness w.r.t. anonymous PKE schemes (i.e., to prevent any ambiguity between senders and receivers), it would be interesting to find further applications for the CM hybrid PKE schemes where a lack of robustness does not hurt.

5.2 FRODOKEM

As seen in Section 3.1, FrodoKEM uses the "$\mathsf{FO}^{\mathrm{frodo}}$" transform on its base FrodoPKE scheme (see Figure 3.3) which differs significantly from the standard $\mathsf{FO}^{\not\perp}$ transform. These significant deviations not only act as a barrier to applying generic results in the literature on IND-CCA enhancement of $\mathsf{FO}^{\not\perp}$ to $\mathsf{FO}^{\mathrm{frodo}}$ in the QROM, as argued in that section, but also act as an obstacle to applying our generic results in Section 4.5 on anonymity and SCFR enhancement of $\mathsf{FO}^{\not\perp}$ to FrodoKEM's variant of the FO transform.

Fortunately, our approach to repairing FrodoKEM's IND-CCA security proof in Section 3.1 also allows us to derive proofs of anonymity and SCFR enhancement for $\mathsf{FO}^{\mathrm{frodo}}$ with similar tightness in the QROM. As will be seen below, in contrast to our generic analysis of $\mathsf{FO}^{\not\perp}$ in Section 4.5 (Theorems 7 and 8) wherein we had to rely on SCFR-CPA security of the base PKE scheme – specifically, its deterministic version – we instead rely on hardness of the *claw-finding* problem in a quantum setting (see Lemma 11).

ciphertext. However, it is easy to see that Xagawa's third-round analysis also extends to the current fourth-round CM specification in a straightforward fashion.

5.2.1 *Anonymity of FrodoPKE*

In our following anonymity (i.e., ANO-CCA security) analysis of FrodoKEM, and the hybrid PKE schemes derived from it, we rely on certain weak security properties of the base FrodoPKE scheme – i.e., ANO-CPA security[5] and IND-CPA security. As noted in Subsection 3.1.3 above, IND-CPA security of FrodoPKE scheme was rigorously established in [16, Subsection 5.1.4]. Upon a close inspection of the corresponding security proof, which is the same as that of [84, Theorem 3.2], it basically proves a stronger property of FrodoPKE $= (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ while relying on the LWE hardness assumption: namely, given $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$ and *any* valid message m, the distribution $\{(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m))\}$ is computationally indistinguishable from $\{(\mathsf{pk}, c^*)\}$ where c^* is a uniformly random value from FrodoPKE's ciphertext space which is independent of pk. It is then not hard to see how we can use this property to also establish ANO-CPA security of FrodoPKE. Namely, in the ANO-CPA security game w.r.t. FrodoPKE, given two honestly-generated public keys $\mathsf{pk}_0, \mathsf{pk}_1$ and a message m chosen by an adversary, it cannot distinguish $(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{Enc}(\mathsf{pk}_0, m))$ from $(\mathsf{pk}_0, \mathsf{pk}_1, c^*)$ where c^* is a uniformly random ciphertext independent of pk_0 and pk_1. Similarly, using the above property, the adversary also cannot distinguish $(\mathsf{pk}_0, \mathsf{pk}_1, c^*)$ from $(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{Enc}(\mathsf{pk}_1, m))$. It follows that the adversary cannot distinguish between the encryptions of m under pk_0 and pk_1, thereby establishing the ANO-CPA security of FrodoPKE. Hence, we have:[6]

Lemma 12 (informal)**.** FrodoPKE *is* ANO-CPA *secure, assuming hardness of the* LWE *problem.*

5.2.2 *Anonymity and Collision-Freeness of FrodoKEM*

We now formally establish the stronger properties of ANO-CCA and SCFR-CCA security for FrodoKEM $= \mathsf{FO}^{\text{frodo}}[\text{FrodoPKE}, G, H, H']$ (see Figure 3.3) in the QROM. Below we define $\mathbf{Coll}^{H}_{\text{FrodoPKE}}$ as probability of the event "$H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$" where pk_0 and pk_1 are two honestly-generated

[5] Technically, the weaker notion of wANO-CPA security (see Footnote 6 of Chapter 4) would have sufficed – as seen in our generic ANO-CCA security analysis of $\mathsf{FO}^{\not\perp}$-based KEMs (Theorem 7). But we go on to prove the stronger ANO-CPA security for FrodoPKE. It would be interesting to find applications for FrodoPKE where its wANO-CPA security does not suffice by ANO-CPA does.

[6] It is not hard to derive concrete security bounds on the ANO-CPA security of FrodoPKE using the results in [16, Subsection 5.1.4]. However we consider it out-of-scope of this chapter since we are mainly concerned with "CPA→CCA" style enhancement results.

FrodoPKE public-keys. Given the space of FrodoPKE's public keys is sufficiently large, if the hash function H is sufficiently collision-resistant, then $\mathbf{Coll}^{H}_{\text{FrodoPKE}}$ can be considered to be negligible.[7]

Theorem 10. *Given* FrodoPKE = $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct and γ-spread, for any* ANO-CCA *adversary $\mathcal{A}$ against* FrodoKEM = $(\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_G and $q_{H'}$ queries to the quantum random oracles G and H' respectively, there exist an* ANO-CPA *adversary $\mathcal{B}$ and* IND-CPA *adversary $\mathcal{B}'$ against* FrodoPKE *such that*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\text{FrodoKEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CPA}}_{\text{FrodoPKE}}(\mathcal{B}) + 2(q_G + q_{H'})\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\text{FrodoPKE}}(\mathcal{B}') + \frac{1}{2^{256}}}$$
$$+ \mathbf{Coll}^{H}_{\text{FrodoPKE}} + \frac{C(q_G+1)^3+1}{2^{256}} + \frac{4q_{H'}}{2^{128}} + 2^{-\gamma} + 8q_G\sqrt{\delta},$$

where C (< 648) is the constant from Lemma 11, and the running times of $\mathcal{B}$ and $\mathcal{B}'$ are the same as that of $\mathcal{A}$.

Proof. The structure of the proof is similar to that of Theorem 7. Denote $\Omega_{\mathbf{G}_2}$, $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$ and $\Omega_{\mathbf{H}'}$ to be the set of all functions $\mathbf{G}_2 : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{G} : \{0,1\}^{256} \to \{0,1\}^{256}$, $\mathbf{H} : \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ and $\mathbf{H}' : \mathcal{C} \to \{0,1\}^{256}$ respectively, where $\mathcal{C}$ is the ciphertext space of FrodoPKE/FrodoKEM.

Let $\mathcal{A}$ be an adversary in the ANO-CCA game for FrodoKEM issuing at most q_G and $q_{H'}$ quantum queries to the random oracles G and H' respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_{12}$ described in Figures 5.2 and 5.3.

Game G_0: The game G_0 is exactly the ANO-CCA game for FrodoKEM. Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\text{FrodoKEM}}(\mathcal{A}).$$

Game G_1: In game G_1, we modify the decapsulation oracles $\textsc{Decaps}_{c^*}(0,\cdot)$ (resp., $\textsc{Decaps}_{c^*}(1,\cdot)$) such that $H_0^{\text{rej}}(c)$ (resp., $H_1^{\text{rej}}(c)$) is returned instead of $H'(s_0, c)$ (resp., $H'(s_1, c)$) for an invalid ciphertext c. Since this change is similar to the sequence of game-hops "$\mathsf{G}_0 \to \mathsf{G}_{0.5} \to \mathsf{G}_1$" in the proof of Theorem 7, using Lemma 2 w.r.t. the pseudorandomness of $H'(s_0,\cdot)$ and $H'(s_1,\cdot)$ with PRF keys $s_0, s_1 \leftarrow\!\!\$ \{0,1\}^{256}$ respectively, it is not hard to obtain

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{4q_{H'}}{\sqrt{2^{256}}}.$$

[7] Alternatively, we could model H as a random oracle and then have $\mathbf{Coll}^{H}_{\text{FrodoPKE}} \leq \frac{1}{2^{256}}$. However, H is not modelled as a random oracle in the original (IND-CCA) security analysis of FrodoKEM in its NIST PQC specification document [16]. Hence, we don't do this either in our following anonymity and robustness analysis.

Games $\mathsf{G}_0 - \mathsf{G}_{8.5}$

```
$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
$G_2 \leftarrow\!\!\$\ \Omega_{\mathbf{G}_2}; G_{0r}, G_{1r} \leftarrow\!\!\$\ \Omega_{\mathbf{G}}$
$G_{0r}^{\text{good}} \leftarrow \Omega_{\mathbf{G}}; G_{0r} := G_{0r}^{\text{good}}$   // $\mathsf{G}_{6.5}$ - $\mathsf{G}_{8.5}$
$G_{1r}^{\text{good}} \leftarrow \Omega_{\mathbf{G}}; G_{1r} := G_{1r}^{\text{good}}$   // $\mathsf{G}_7$ - $\mathsf{G}_{8.5}$
$G_{0k}, G_{1k} \leftarrow\!\!\$\ \Omega_{\mathbf{G}}$   // $\mathsf{G}_0$ - $\mathsf{G}_8$
$G_{0k}, G_{1k} \leftarrow\!\!\$\ \Omega_{\mathbf{poly}}$   // $\mathsf{G}_{8.5}$
$H_2 \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H_0^{\mathsf{rej}}, H_1^{\mathsf{rej}} \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
$H_3 \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H_0^{\text{acc}}, H_1^{\text{acc}} \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
$b \leftarrow\!\!\$\ \{0,1\}$
$m^* \leftarrow\!\!\$\ \{0,1\}^{256}$
$(\bar{k}^*, r^*) \leftarrow G(m^*, H(\mathsf{pk}_b))$   // $\mathsf{G}_0 - \mathsf{G}_2$
$r^* \leftarrow G_{br}(m^*)$   // $\mathsf{G}_3 - \mathsf{G}_{8.5}$
$\bar{k}^* \leftarrow G_{bk}(m^*)$   // $\mathsf{G}_3 - \mathsf{G}_7$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
$k^* \leftarrow H(\bar{k}^*, c^*)$   // $\mathsf{G}_0 - \mathsf{G}_7$
$k^* \leftarrow H_3(m^*)$   // $\mathsf{G}_8 - \mathsf{G}_{8.5}$
$\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
$b' \leftarrow \mathcal{A}^{G,H',\textsc{Decaps}_{c^*}}(\mathsf{inp})$
return $[b' = b]$
```

$\textsc{Decaps}_a(0, c)$ // $c \neq a$

```
return $H_0^{\text{acc}}(c)$   // $\mathsf{G}_{4.5}$ - $\mathsf{G}_{8.5}$
Parse $\mathsf{sk}'_0 = (\mathsf{sk}_0, \mathsf{pk}_0, h_0, s_0)$
$m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
$(\bar{k}', r') \leftarrow G(m', h_0)$   // $\mathsf{G}_0 - \mathsf{G}_2$
$(\bar{k}', r') \leftarrow (G_{0k} \times G_{0r})(m')$   // $\mathsf{G}_3 - \mathsf{G}_4$
if $\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c$ then
  return $H'(\bar{k}', c)$
else return $H'(s_0, c)$   // $\mathsf{G}_0$
else return $H_0^{\mathsf{rej}}(c)$   // $\mathsf{G}_1$ - $\mathsf{G}_4$
```

$G(m, h)$

```
if $h = H(\mathsf{pk}_0)$ then   // $\mathsf{G}_2$-$\mathsf{G}_{8.5}$
  $(\bar{k}, r) \leftarrow (G_{0k} \times G_{0r})(m)$   // $\mathsf{G}_2$-$\mathsf{G}_{8.5}$
elseif $h = H(\mathsf{pk}_1)$ then   // $\mathsf{G}_2$-$\mathsf{G}_{8.5}$
  $(\bar{k}, r) \leftarrow (G_{1k} \times G_{1r})(m)$   // $\mathsf{G}_2$-$\mathsf{G}_{8.5}$
else $(\bar{k}, r) \leftarrow G_2(m, h)$
return $(\bar{k}, r)$
```

$H'(\bar{k}, c)$

```
$m' = \mathsf{Dec}(\mathsf{sk}_0, c)$   // $\mathsf{G}_4 - \mathsf{G}_{8.5}$
if $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c \wedge$
 $G_{0k}(m') = \bar{k}$   // $\mathsf{G}_4 - \mathsf{G}_{8.5}$
  if $c = c^*$   // $\mathsf{G}_6 - \mathsf{G}_{8.5}$
    return $H_3(m')$   // $\mathsf{G}_6 - \mathsf{G}_{8.5}$
  return $H_0^{\text{acc}}(c)$   // $\mathsf{G}_4 - \mathsf{G}_{8.5}$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c)$   // $\mathsf{G}_4 - \mathsf{G}_{8.5}$
if $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c \wedge$
 $G_{1k}(m') = \bar{k}$   // $\mathsf{G}_4 - \mathsf{G}_{8.5}$
  if $c = c^*$   // $\mathsf{G}_6 - \mathsf{G}_{8.5}$
    return $H_3(m')$   // $\mathsf{G}_6 - \mathsf{G}_{8.5}$
  return $H_1^{\text{acc}}(c)$   // $\mathsf{G}_4 - \mathsf{G}_{8.5}$
return $H_2(\bar{k}, c)$
```

$\textsc{Decaps}_a(1, c)$ // $c \neq a$

```
return $H_1^{\text{acc}}(c)$   // $\mathsf{G}_5$ - $\mathsf{G}_{8.5}$
Parse $\mathsf{sk}'_1 = (\mathsf{sk}_1, \mathsf{pk}_1, h_1, s_1)$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
$(\bar{k}', r') \leftarrow G(m', h_1)$   // $\mathsf{G}_0 - \mathsf{G}_2$
$(\bar{k}', r') \leftarrow (G_{1k} \times G_{1r})(m')$   // $\mathsf{G}_3$-$\mathsf{G}_{4.5}$
if $\mathsf{Enc}(\mathsf{pk}_1, m'; r') = c$ then
  return $H'(\bar{k}', c)$
else return $H'(s_1, c)$   // $\mathsf{G}_0$
else return $H_1^{\mathsf{rej}}(c)$   // $\mathsf{G}_1$ - $\mathsf{G}_{4.5}$
```

FIGURE 5.2: Games G_0 – $\mathsf{G}_{8.5}$ for the proof of Theorem 10. The sampling distributions w.r.t. "$G_{0r}^{\text{good}} \leftarrow \Omega_{\mathbf{G}}$" and "$G_{1r}^{\text{good}} \leftarrow \Omega_{\mathbf{G}}$" are described in descriptions of $\mathsf{G}_{6.5}$ and G_7 below.

```
Games $\mathsf{G}_9 - \mathsf{G}_{12}$
$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
$G_2 \leftarrow\$ \Omega_{\mathbf{G}_2}; G_{0r}, G_{1r} \leftarrow\$ \Omega_{\mathbf{G}}$
$G_{0r}^{\text{good}} \leftarrow \Omega_{\mathbf{G}}; G_{0r} := G_{0r}^{\text{good}}$    // $\mathsf{G}_9$
$G_{1r}^{\text{good}} \leftarrow \Omega_{\mathbf{G}}; G_{1r} := G_{1r}^{\text{good}}$    // $\mathsf{G}_9$
$G_{0k}, G_{1k} \leftarrow\$ \Omega_{\mathbf{poly}}$
$H_2 \leftarrow\$ \Omega_{\mathbf{H}}; H_3 \leftarrow\$ \Omega_{\mathbf{G}}$
$H_0^{\text{acc}}, H_1^{\text{acc}} \leftarrow\$ \Omega_{\mathbf{H}'}$
$b \leftarrow\$ \{0,1\}$
$m^* \leftarrow\$ \{0,1\}^{256}$
$r^* \leftarrow G_{br}(m^*)$    // $\mathsf{G}_9 - \mathsf{G}_{10}$
$r^* \leftarrow\$ \{0,1\}^{256}$    // $\mathsf{G}_{11} - \mathsf{G}_{12}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
$k^* \leftarrow H_3(m^*)$    // $\mathsf{G}_9 - \mathsf{G}_{10}$
$k^* \leftarrow\$ \{0,1\}^{256}$    // $\mathsf{G}_{11} - \mathsf{G}_{12}$
$\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
$i \leftarrow\$ \{1, \dots, q_G + q_{H'}\}$    // $\mathsf{G}_{12}$
run $\mathcal{A}^{G, H', \textsc{Decaps}_{c^*}}(\mathsf{inp})$ until
  $i$-th query to $G_{br} \times H_3$    // $\mathsf{G}_{12}$
measure the $i$-th query and let the
  outcome be $m'$    // $\mathsf{G}_{12}$
return $[m' = m^*]$    // $\mathsf{G}_{12}$
$b' \leftarrow \mathcal{A}^{G, H', \textsc{Decaps}_{c^*}}(\mathsf{inp})$
return $[b' = b]$
```

```
$\textsc{Decaps}_a(i, c)$    // $i \in \{0,1\}, c \neq a$
return $H_i^{\text{acc}}(c)$
```

```
$G(m, h)$
if $h = H(\mathsf{pk}_0)$ then
  $r \leftarrow G_{0r}(m)$
  $\bar{k} \leftarrow G_{0k}(m)$
elseif $h = H(\mathsf{pk}_1)$ then
  $r \leftarrow G_{1r}(m)$
  $\bar{k} \leftarrow G_{1k}(m)$
else $(\bar{k}, r) \leftarrow G_2(m, h)$
return $(\bar{k}, r)$
```

```
$H'(\bar{k}, c)$
Compute set of roots $S_0$
  of polynomial $G_{0k}(x) - \bar{k}$
if $\exists m' \in S_0$ s.t.
  $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c$
  if $c = c^*$ then
    return $H_3(m')$
  return $H_0^{\text{acc}}(c)$
Compute set of roots $S_1$
  of polynomial $G_{1k}(x) - \bar{k}$
if $\exists m' \in S_1$ s.t.
  $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c$
  if $c = c^*$ then
    return $H_3(m')$
  return $H_1^{\text{acc}}(c)$
return $H_2(\bar{k}, c)$
```

FIGURE 5.3: Games G_9 – G_{12} for the proof of Theorem 10.

Game G_2**:** In game G_2, we implicitly divide the G-queries into at-most three categories: (1) query is of the form (m, h) with $h = H(\mathsf{pk}_0)$, (2) query is of the form (m, h) with $h = H(\mathsf{pk}_1)$ and (3) the remaining queries. We then respond to queries from the respective categories with $(G_{0k}(m), G_{0r}(m))$, $(G_{1k}(m), G_{1r}(m))$ and $G_2(m, h)$ respectively, where G_{ik}, G_{ir} (for $i \in \{0, 1\}$) are internal random oracles[8]; note that we say "at most" three categories because of the (unlikely) possibility that $H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$. It is not hard to verify that the output distributions of the G-oracle in games G_1 and G_2 are equivalent. Therefore,

$$\Pr[\mathsf{G}_2 = 1] = \Pr[\mathsf{G}_1 = 1].$$

Game G_3**:** In game G_3, we make the following changes w.r.t. the G-oracle evaluation. First, we generate the values $\overline{k}^*, r^*$ in setup of the game as "$\overline{k}^* \leftarrow G_{bk}(m^*)$" and "$r^* \leftarrow G_{br}(m^*)$" (effectively replacing the step "$(\overline{k}^*, r^*) \leftarrow G(m^*, H(\mathsf{pk}_b))$" in G_2). We then similarly generate the values $\overline{k}', r'$ w.r.t. the decapsulation oracles $\textsc{Decaps}_{c^*}(i, \cdot)$ $(i \in \{0, 1\})$ as "$\overline{k}' \leftarrow G_{ik}(m')$" and "$r' \leftarrow G_{ir}(m')$" (replacing the step "$(\overline{k}', r') \leftarrow G(m', h_i)$" in G_2, where $h_i = H(\mathsf{pk}_i)$ because we assume honest generation of the key-pair $(\mathsf{pk}, \mathsf{sk}')$ at setup).

Let "bad" denote the event where the public keys pk_0 and pk_1 generated honestly in the setup satisfy "$H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$". It is not hard to see that the games G_2 and G_3 are equivalent unless the event bad happens. Hence, from our definition of the probability $\mathbf{Coll}^H_{\text{FrodoPKE}}$ above, we have

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \le \Pr[\mathsf{bad}] \le \mathbf{Coll}^H_{\text{FrodoPKE}}.$$

Game G_4**:** In game G_4, we implicitly divide the H'-queries into three disjoint categories: (1) query is of the form $(\overline{k}, c)$ which satisfies $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ and $G_{0k}(m) = \overline{k}$, where $m = \mathsf{Dec}(\mathsf{sk}_0, c)$, (2) query is of the form $(\overline{k}, c)$ which does not fall under "category (1)", while at the same time, satisfies $\mathsf{Enc}(\mathsf{pk}_1, m; G_{1r}(m)) = c$ and $G_{1k}(m) = \overline{k}$, where $m = \mathsf{Dec}(\mathsf{sk}_1, c)$, and (3) the remaining queries. We then respond to queries from the respective categories with $H_0^{\text{acc}}(c)$, $H_1^{\text{acc}}(c)$ and $H_2(\overline{k}, c)$, where H_0^{acc} and H_1^{acc} are internal random oracles not directly accessible to the adversary $\mathcal{A}$.

Focusing on H'-queries in "category (1)", note that it is not possible for two distinct queries $(\overline{k}', c)$ and $(\overline{k}'', c)$ to result in the same output $H_0^{\text{acc}}(c)$. Note that $\mathsf{Dec}(\mathsf{sk}_0, \cdot)$ and $G_{0k}(\cdot)$ are deterministic functions. Hence

[8] In Figure 5.2, we define the random oracle $(G_{ik} \times G_{ir}) \in \Omega_{\mathbf{G}}$ as $(G_{ik} \times G_{ir})(m) = (G_{ik}(m), G_{ir}(m))$.

w.r.t. the queries $(\overline{k}', c)$ and $(\overline{k}'', c)$, there can only exist a unique value m such that $m = \mathsf{Dec}(\mathsf{sk}_0, c)$. At the same time, $G_{0k}(m)$ can take at most one value. The same reasoning applies to "category (2)" as well, and hence, the output distributions of the H'-oracle in the games G_3 and G_4 are equivalent. Therefore,

$$\Pr[\mathsf{G}_4 = 1] = \Pr[\mathsf{G}_3 = 1].$$

Game $\mathsf{G}_{4.5}$**:** In game $\mathsf{G}_{4.5}$, we change the $\textsc{Decaps}_{c^*}(0, \cdot)$ oracle such that there is no need for the secret key sk'_0. Namely, $H_0^{\mathrm{acc}}(c)$ is returned for the decapsulation of ciphertext c w.r.t. sk'_0. Let $m' = \mathsf{Dec}(\mathsf{sk}_0, c)$, $r' = G_{0r}(m')$ and $\overline{k}' = G_{0k}(m')$. Now consider the following two cases:

1. $\underline{\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c}$: In this case, the $\textsc{Decaps}_{c^*}(0, \cdot)$ oracle returns $H'(\overline{k}', c)$ in game G_4 and $H_0^{\mathrm{acc}}(c)$ in game $\mathsf{G}_{4.5}$. Hence, it is not hard to see that we have $H'(\overline{k}', c) = H_0^{\mathrm{acc}}(c)$ in G_4, since the query $(\overline{k}', c)$ falls under "category (1)" w.r.t. oracle H'. Therefore, $\textsc{Decaps}_{c^*}(0, \cdot)$ oracles of games G_4 and $\mathsf{G}_{4.5}$ return the same value $H_0^{\mathrm{acc}}(c)$.

2. $\underline{\mathsf{Enc}(\mathsf{pk}_0, m'; r') \neq c}$: In this case, the $\textsc{Decaps}_{c^*}(0, \cdot)$ oracle returns $H_0^{\mathrm{rej}}(c)$ in game G_4 and $H_0^{\mathrm{acc}}(c)$ in game $\mathsf{G}_{4.5}$. In game G_4, as the random function H_0^{rej} is independent of all other oracles, the output $H_0^{\mathrm{rej}}(c)$ is uniformly random in the adversary $\mathcal{A}$'s view. In game $\mathsf{G}_{4.5}$, the only way $\mathcal{A}$ gets prior access to the value $H_0^{\mathrm{acc}}(c)$ is if it made a H'-query $(\overline{k}'', c)$ such that $\mathsf{Enc}(\mathsf{pk}_0, m''; G_{0r}(m'')) = c$ (and $G_{0k}(m'') = \overline{k}''$), where $m'' = \mathsf{Dec}(\mathsf{sk}_0, c)$. But since $\mathsf{Dec}(\mathsf{sk}_0, \cdot)$ is a deterministic function, we have $m'' = m'$ leading to a contradiction of "$\mathsf{Enc}(\mathsf{pk}_0, m'; r') \neq c$". Therefore, such a prior access is not possible and $H_0^{\mathrm{acc}}(c)$ will also be a uniformly random value in $\mathcal{A}$'s view.

As the output distributions of $\textsc{Decaps}_{c^*}(0, \cdot)$ oracle in G_4 and $\mathsf{G}_{4.5}$ are the same in both cases, we have

$$\Pr[\mathsf{G}_{4.5} = 1] = \Pr[\mathsf{G}_4 = 1].$$

Game G_5**:** In game G_5, we change the $\textsc{Decaps}_{c^*}(1, \cdot)$ oracle such that $H_1^{\mathrm{acc}}(c)$ is returned for the decapsulation of *any* ciphertext c w.r.t. sk'_1. The analysis here follows quite similarly to that of the previous game-hop except that this simulation of the $\textsc{Decaps}_{c^*}(1, \cdot)$ oracle – without the secret key sk'_1 – will fail (w.r.t. case 1 in the above game-hop) if $\mathcal{A}$ asks for the decapsulation of a ciphertext $\hat{c}$ such that $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = \hat{c} =$

$\mathsf{Enc}(\mathsf{pk}_0, m''; G_{0r}(m''))$ and $G_{1k}(m') = \overline{k}' = G_{0k}(m'')$, where $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$ and $m'' = \mathsf{Dec}(\mathsf{sk}_0, \hat{c})$. In this peculiar case, $H_0^{\mathrm{acc}}(\hat{c})$ is returned in $\mathsf{G}_{4.5}$ and $H_1^{\mathrm{acc}}(\hat{c})$ is returned in G_5.

We bound the probability of this peculiar event (i.e., $\mathcal{A}$ asking for the decapsulation of such an above ciphertext $\hat{c}$ w.r.t. sk_1') by the advantage of an adversary $\hat{\mathcal{B}}$ against the *claw-finding* game w.r.t. the pair of quantum random oracles (G_{0k}, G_{1k}). Because note that the pair (m'', m) is a *claw* with $G_{0k}(m'') = G_{1k}(m')$, where $m'' = \mathsf{Dec}(\mathsf{sk}_0, \hat{c})$ and $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$. More formally, $\hat{\mathcal{B}}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game $\mathsf{G}_{4.5}$, by creating the appropriate setup (starting with the generation of two honest key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0')$ and $(\mathsf{pk}_1, \mathsf{sk}_1')$).

- Uses three different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_{0r} and G_{1r} respectively, four different $2q_{H'}$-wise independent functions to simulate the random oracles H_0^{acc}, H_1^{acc}, H_1^{rej} and H_2 respectively in $\mathcal{A}$'s view, as noted in Lemma 1. Also uses the pair of oracles $f_0 : \{0,1\}^{256} \to \{0,1\}^{256}$ and $f_1 : \{0,1\}^{256} \to \{0,1\}^{256}$ – which is the instance of the claw-finding game – to simulate the oracles G_{0k} and G_{1k} respectively.

- Answers decapsulation queries the same way as in $\mathsf{G}_{4.5}$. Particularly, w.r.t. any query $\hat{c}$ made by $\mathcal{A}$ to the DECAPS$(1, \cdot)$ oracle, checks if the query satisfies the above described peculiar event. If so, returns the pair (m'', m') as a claw w.r.t. (f_0, f_1), where $m'' = \mathsf{Dec}(\mathsf{sk}_0, \hat{c})$ and $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$.

Note that $\hat{\mathcal{B}}$ makes at most q_G queries to the pair (f_0, f_1). Let Pr[pec] be the probability of this peculiar event, denoted as "pec", occurring. We have the games $\mathsf{G}_{4.5}$ and G_5 to be equivalent unless the event pec occurs. From the construction of our claw-finding adversary $\hat{\mathcal{B}}$ above, it is not hard to see that $\Pr[\mathsf{pec}] \leq \frac{C(q_G+1)^3}{2^{256}}$ from Lemma 11. Hence, we have

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_{4.5} = 1]| \leq \Pr[\mathsf{pec}] \leq \frac{C(q_G + 1)^3}{2^{256}}.$$

Game G_6**:** In game G_6, we make a further modification to the evaluation of "category (1) and (2)" H'-queries (as introduced in the "$\mathsf{G}_3 \to \mathsf{G}_4$" game-hop) of the form $(\overline{k}, c^*)$ as follows, where c^* is the challenge ciphertext computed in the setup: respond to the corresponding "category (1)" queries with $H_3(m)$, where $m = \mathsf{Dec}(\mathsf{sk}_0, c)$, and the corresponding "category (2)" queries with $H_3(m)$, where $m = \mathsf{Dec}(\mathsf{sk}_1, c)$. Here H_3 is an internal independent random oracle.

Let $m_0 = \mathsf{Dec}(\mathsf{sk}_0, c^*)$ and $m_1 = \mathsf{Dec}(\mathsf{sk}_1, c^*)$ which additionally satisfy $\mathsf{Enc}(\mathsf{pk}_0, m_0; G_{0r}(m_0)) = c^*$ and $\mathsf{Enc}(\mathsf{pk}_1, m_1; G_{1r}(m_1)) = c^*$. So to analyze this change to oracle H', there are only two H'-queries worth considering: namely, "category (1)" query $(\overline{k}_0, c^*)$ and "category (2)" query $(\overline{k}_1, c^*)$ where $\overline{k}_0 = G_{0k}(m_0)$ and $\overline{k}_1 = G_{1k}(m_1)$. With respect to these two queries, the H' oracle would return $H_0^{\mathrm{acc}}(c^*)$, $H_1^{\mathrm{acc}}(c^*)$ respectively in G_5, and $H_3(m_0)$, $H_3(m_1)$ respectively in G_6. Conditional on $m_0 \neq m_1$, the adversary $\mathcal{A}$'s view would be identical even after this change because the random values $H_0^{\mathrm{acc}}(c^*)$, $H_1^{\mathrm{acc}}(c^*)$ are only accessible to $\mathcal{A}$ via the H'-oracle in G_5, and in particular, not through the $\textsc{Decaps}_{c^*}(i, \cdot)$ oracles since c^* is a forbidden decapsulation query. Hence in G_6, we are effectively replacing two uniformly random values that can only be accessed via the H'-oracle by $\mathcal{A}$ with two other uniformly random values. Hence, the output distributions of the H'-oracle in games G_5 and G_6 are equivalent unless we have $m_0 = m_1$, or in other words, the following event occurs w.r.t. two honest FrodoPKE key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)$: $\mathsf{Dec}(\mathsf{sk}_0, c^*) = \mathsf{Dec}(\mathsf{sk}_1, c^*) = m'$ *and* $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = \mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c^*$, where for $m^* \leftarrow\!\$ \{0,1\}^{256}$ and $b \leftarrow\!\$ \{0,1\}$, we have $c^* = \mathsf{Enc}(\mathsf{pk}_b, m^*; G_{br}(m^*))$ (note that we are not assuming the correctness of FrodoPKE, i.e., m^* may or may not be equal to m').

We can bound the probability of the above event by considering the *sub-event* "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G_{(1-b)r}(m')) = c^*$". Note that in the context of an experiment describing the above event at the setup, we have $G_{(1-b)r}(m')$ resulting in uniformly random coins $r' \leftarrow\!\$ \{0,1\}^{256}$, since G_{br} is used to compute the ciphertext c^* and $G_{(1-b)r}$ is a random oracle independent to G_{br}. Since FrodoPKE is γ-spread, for the key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b})$ and message m', we have the condition "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c^*$" to hold with probability $\leq 2^{-\gamma}$ for uniformly random r'. Hence, we have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq 2^{-\gamma}.$$

Game $\mathsf{G}_{6.5}$**:** In game $\mathsf{G}_{6.5}$, we change the random oracle G_{0r} such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}_0, \mathsf{sk}_0)$, as seen in the proof of Theorem 7. Specifically, define the oracle $G_{0r}^{\mathrm{good}} \leftarrow \Omega_{\mathbf{G}}$ such that $G_{0r}^{\mathrm{good}}(m)$ such that $G_{0r}^{\mathrm{good}}(m)$ is sampled according to a uniform distribution in $\mathcal{R}_{\mathrm{good}}((\mathsf{pk}_0, \mathsf{sk}_0), m)$. Hence in $\mathsf{G}_{6.5}$, we replace the oracle G_{0r}

with G_{0r}^{good}. By using a similar analysis as the game-hop "$\mathsf{G}_1 \to \mathsf{G}_2$" in the proof of Theorem 7 (in fact, the analysis would be simpler in this case since we have to consider a single key-pair $(\mathsf{pk}_0, \mathsf{sk}_0)$ instead of two), it is not hard to obtain

$$|\Pr[\mathsf{G}_{6.5} = 1] - \Pr[\mathsf{G}_6 = 1]| \leq 2q_G\sqrt{\delta}.$$

Game G_7**:** In game G_7, we now change the random oracle G_{1r} such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}_1, \mathsf{sk}_1)$. The analysis in this case would be similar (and simpler when compared) to the game-hop "$\mathsf{G}_1 \to \mathsf{G}_2$" in the proof of Theorem 7. But a thing worth noting is that the distinguisher $B^{\hat{G}}$ (for $\hat{G} \in \{G_{1r}, G_{1r}^{\text{good}}\}$) – as was used in the "$\mathsf{G}_1 \to \mathsf{G}_2$" game-hop in the proof of Theorem 7 – will have a single key-pair $(\mathsf{pk}_1, \mathsf{sk}_1)$ as input, and will need to simulate $\mathcal{A}$'s view in the games $\mathsf{G}_{6.5}$ and G_7. But since the distinguisher $B^{\hat{G}}$ can be *unbounded*, it can simulate the "non-ideal" random oracle G_{0r}^{good} that is used in $\mathsf{G}_{6.5}$ and G_7. Again, it is not hard to obtain

$$|\Pr[\mathsf{G}_7 = 1] - \Pr[\mathsf{G}_{6.5} = 1]| \leq 2q_G\sqrt{\delta}.$$

Game G_8**:** In the setup of game G_8, we generate the value k^* as "$k^* \leftarrow H_3(m^*)$" (as opposed to "$k^* \leftarrow H(\overline{k}^*, c^*)$" in G_7). Also $\overline{k}^*$ is not generated in the setup (i.e., we are removing the step "$\overline{k}^* \leftarrow G_{bk}(m^*)$" in G_7) as the value is not required anymore in the game. Note that G_8 is equivalent to G_7 w.r.t. this change unless the following event occurs: for $b = 1$ if we have $c^* = \mathsf{Enc}(\mathsf{pk}_1, m^*; G_{1r}(m^*))$ and $\overline{k}^* \leftarrow G_{1k}(m^*)$ (for $m^* \leftarrow\$ \{0,1\}^{256}$) in the setup, then $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c^*$ and $G_{0k}(m') = \overline{k}^*$, where $\mathsf{Dec}(\mathsf{sk}_0, c^*) = m'$. Note that in this case, the value k^* computed in setup of the games will be equal to $H_3(m')(= H(\overline{k}^*, c^*))$ in G_7 and $H_3(m^*)$ in G_8.

We can bound the probability of such an event by considering the *sub-event* "$G_{0k}(m') = G_{1k}(m^*)$ $(= \overline{k}^*)$". More formally, consider a (hypothetical) experiment which describes the above event at the setup as follows. First, it generates (honestly) two FrodoPKE key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$. Then it uniformly at random samples a message $m^* \leftarrow\$ \{0,1\}^{256}$ and computes $c^* = \mathsf{Enc}(\mathsf{pk}_1, m^*; G_{1r}(m^*))$, $\overline{k}^* \leftarrow G_{1k}(m^*)$; one thing worth noting here is that the hypothetical experiment can simulate the "non-ideal" random oracle G_{1r}, which only samples "good" random coins, with an unbounded running time. Then it computes $m' = \mathsf{Dec}(\mathsf{sk}_0, c^*)$ and finally checks if "$G_{0k}(m') = \overline{k}^*$". Note that in the context of this experiment, since this is the first invocation of the oracle G_{0k} (independent to G_{1k}), $G_{0k}(m')$ results in a uniformly random value $\overline{k}' \leftarrow\$ \{0,1\}^{256}$. Therefore, the probability of this sub-event, or "$G_{0k}(m') = \overline{k}^*$", happening is at most $1/2^{256}$. Hence, it is not hard to see that

$$|\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{1}{2^{256}}.$$

Game $\mathsf{G}_{8.5}$**:** In game $\mathsf{G}_{8.5}$, we replace the random oracles G_{ik} ($i \in \{0,1\}$) with $2q_G$-wise independent functions, following Lemma 1. Random polynomials of degree $2q_G - 1$ over the finite field representation of the message space $\{0,1\}^{256}$ are $2q_G$-wise independent. Let $\Omega_{\mathbf{poly}}$ be the set of all such polynomials. We are then replacing the step "$G_{0k}, G_{1k} \leftarrow\$ \Omega_{\mathbf{G}}$" with "$G_{0k}, G_{1k} \leftarrow\$ \Omega_{\mathbf{poly}}$" in $\mathsf{G}_{8.5}$. From Lemma 1, as this change is indistinguishable when the oracles G_{0k}, G_{1k} are queried at most q_G times, we have

$$\Pr[\mathsf{G}_{8.5} = 1] = \Pr[\mathsf{G}_8 = 1].$$

Game G_9**:** In game G_9, we change the H'-oracle such that there is no need for secret keys sk_0, sk_1. Namely, we implicitly divide the H'-queries into three disjoint categories: (1) query is of the form $(\overline{k}, c)$ such that there exists $m \in \{0,1\}^{256}$ which is a root of the polynomial $G_{0k}(x) - \overline{k}$ (recall that G_{0k} and G_{1k} are now polynomials) *and* $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$, (2) query is of the form $(\overline{k}, c)$ such that it does not fall under "category (1)", while at the same time, there exists $m \in \{0,1\}^{256}$ which is a root of the polynomial $G_{1k}(x) - \overline{k}$ *and* $\mathsf{Enc}(\mathsf{pk}_1, m; G_{1r}(m)) = c$, and (3) the remaining queries. We then respond to queries from the respective categories as follows: (1) return $H_3(m)$ if $c = c^*$, otherwise return $H_0^{\mathrm{acc}}(c)$, (2) return $H_3(m)$ if $c = c^*$, otherwise return $H_1^{\mathrm{acc}}(c)$, and (3) return $H_2(\overline{k}, c)$.

It is not hard to see that the input-output behavior of oracle H' in games $\mathsf{G}_{8.5}$ and G_9 is identical. For example, w.r.t. a query $(\overline{k}, c)$ if the oracle H' in $\mathsf{G}_{8.5}$ returns $H_0^{\mathrm{acc}}(c)$, then we have $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ ($\neq c^*$) and $G_{0k}(m) = \overline{k}$, where $m = \mathsf{Dec}(\mathsf{sk}_0, c)$. This implies that m is the *only* root of the polynomial $G_{0k}(x) - \overline{k}$ which satisfies $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ (note that there cannot exist some other root m' ($\neq m$) of $G_{0k}(x) - \overline{k}$ satisfying $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c$ because, as G_{0r} samples "good" random coins, we must then have $\mathsf{Dec}(\mathsf{sk}_0, c) = m' = m$, a contradiction), and hence on the same input $(\overline{k}, c)$, oracle H' in G_9 outputs the value $H_0^{\mathrm{acc}}(c)$ as well. In the other direction, w.r.t. a query $(\overline{k}, c)$ if the oracle H' in G_9 returns $H_0^{\mathrm{acc}}(c)$, then there exists a root m of the polynomial $G_{0k}(x) - \overline{k}$ such that it *uniquely* satisfies $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ ($\neq c^*$). Since G_{0r} samples "good" random coins, we must have $\mathsf{Dec}(\mathsf{sk}_0, c) = m$ with m satisfying $G_{0k}(m) = \overline{k}$ and

$\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$. Therefore, on the same input $(\bar{k}, c)$, oracle H' in $\mathsf{G}_{8.5}$ outputs the value $H_0^{\text{acc}}(c)$ as well. A similar reasoning applies to the outputs $H_1^{\text{acc}}(c)$ and $H_2(\bar{k}, c)$ w.r.t. H'-queries $(\bar{k}, c)$, and also to queries of the form $(\bar{k}, c^*)$, which finally leads to the equivalence of oracles H' in $\mathsf{G}_{8.5}$ and G_9. We thus have

$$\Pr[\mathsf{G}_9 = 1] = \Pr[\mathsf{G}_{8.5} = 1].$$

Game G_{10}: In game G_{10}, we reset the random oracles G_{ir} (for $i \in \{0, 1\}$) so that they return uniformly random coins from $\{0, 1\}^{256}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_6 \to \mathsf{G}_7$, by using a similar analysis we obtain

$$|\Pr[\mathsf{G}_{10} = 1] - \Pr[\mathsf{G}_9 = 1]| \leq 4q_G\sqrt{\delta}.$$

Game G_{11}: In the setup of game G_{11}, we generate the values r^* and k^* such that they are uniformly random values independent of any oracles, i.e., we replace the step "$r^* \leftarrow G_{br}(m^*)$" with "$r^* \leftarrow\!\!\$ \{0, 1\}^{256}$" and "$k^* \leftarrow H_3(m^*)$" with "$k^* \leftarrow\!\!\$ \{0, 1\}^{256}$". We use Lemma 3 to bound the difference in the success probabilities of $\mathcal{A}$ in G_{10} and G_{11}. Let A be an oracle algorithm that has quantum access to the random oracle $G_r \times H_3$, where $(G_r \times H_3)(m) = (G_r(m), H_3(m))$. Figure 5.4 describes $A^{G_r \times H_3}$'s operation on input $(m^*, (r^*, k^*))$. Note that the algorithm $A^{G_r \times H_3}$ makes at most $q_G + q_{H'}$ number of queries to the random oracle $G_r \times H_3$ to respond to $\mathcal{A}$'s G-oracle and H-oracle queries.[9] With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_{10} = 1]$ and $P_A^2 = \Pr[\mathsf{G}_{11} = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 3 w.r.t. the algorithm $A^{G_r \times H_3}$. To analyze the corresponding probability P_B in Lemma 3, we define game G_{12} (see Fig. 5.3) such that $P_B = \Pr[\mathsf{G}_{12} = 1]$. From Lemma 3, we thus have

$$|\Pr[\mathsf{G}_{10} = 1] - \Pr[\mathsf{G}_{11} = 1]| \leq 2(q_G + q_{H'})\sqrt{\Pr[\mathsf{G}_{12} = 1]}.$$

We now bound the success probability of $\mathcal{A}$ in G_{11} by the advantage of an adversary $\mathcal{B}$ in the ANO-CPA game of FrodoPKE. Upon receiving public-keys pk_0 and pk_1, $\mathcal{B}$ submits a uniformly random message $m^* \leftarrow\!\!\$ \{0, 1\}^{256}$ to the ANO-CPA challenger. It then receives a ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ for uniformly random bit $b (\leftarrow\!\!\$ \{0, 1\})$ and randomness $r^* (\leftarrow\!\!\$ \{0, 1\}^{256})$ chosen by the challenger. $\mathcal{B}$ then proceeds as follows:

[9] See Footnote 3 of Chapter 3 for details on how $A^{G_r \times H_3}$ can respond to the adversary $\mathcal{A}$'s hash queries.

```
$A^{G_r \times H_3}(m^*, (r^*, k^*))$
    $(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
    $G_2 \leftarrow\!\!\$\ \Omega_{\mathbf{G}_2}; G_{0k}, G_{1k} \leftarrow\!\!\$\ \Omega_{\mathbf{poly}}$
    $H_2 \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H_0^{\mathrm{acc}}, H_1^{\mathrm{acc}} \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
    $b \leftarrow\!\!\$\ \{0,1\}$
    $G_{br} := G_r; G_{(1-b)r} \leftarrow\!\!\$\ \Omega_{\mathbf{G}}$
    $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
    $\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
    $b' \leftarrow \mathcal{A}^{G, H', \textsc{Decaps}_{c^*}}(\mathsf{inp})$
    return $[b' = b]$
```

```
$G(m, h)$
    if $h = H(\mathsf{pk}_0)$ then
        $r \leftarrow G_{0r}(m)$
        $\overline{k} \leftarrow G_{0k}(m)$
    elseif $h = H(\mathsf{pk}_1)$ then
        $r \leftarrow G_{1r}(m)$
        $\overline{k} \leftarrow G_{1k}(m)$
    else $(\overline{k}, r) \leftarrow G_2(m, h)$
    return $(\overline{k}, r)$
```

```
$H(\overline{k}, c)$
    Compute set of roots $S_0$
     of polynomial $G_{0k}(x) - \overline{k}$
    if $\exists m' \in S_0$ s.t.
     $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c$
        if $c = c^*$ then
            return $H_3(m')$
        return $H_0^{\mathrm{acc}}(c)$
    Compute set of roots $S_1$
     of polynomial $G_{1k}(x) - \overline{k}$
    if $\exists m' \in S_1$ s.t.
     $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c$
        if $c = c^*$ then
            return $H_3(m')$
        return $H_1^{\mathrm{acc}}(c)$
    return $H_2(\overline{k}, c)$
```

```
$\textsc{Decaps}_a(i, c)$    // $i \in \{0,1\}, c \neq a$
    return $H_i^{\mathrm{acc}}(c)$
```

Figure 5.4: Algorithm $A^{G_r \times H_3}$ for the proof of Theorem 10.

- Runs $\mathcal{A}$ as a subroutine as in game G_{11}.
- Uses five different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_{0r}, G_{1r}, G_{0k} and G_{1k} respectively, four different $2q_{H'}$-wise independent functions to simulate the random oracles H_0^{acc}, H_1^{acc}, H_2 and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 1. The random oracles G and H are simulated in the same way as in G_{11}.
- Answers decapsulation queries using the oracles H_i^{acc} ($i \in \{0,1\}$) as in G_{11}.
- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^* \leftarrow\$ \{0,1\}^{256}$ and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$.
- After obtaining a bit b' from $\mathcal{A}$, forwards b' to its ANO-CPA challenger as the final message.

It is easy to see that $|\Pr[\mathsf{G}_{11} = 1] - \frac{1}{2}| = \mathbf{Adv}^{\text{ANO-CPA}}_{\text{FrodoPKE}}(\mathcal{B})$. Now we bound the success probability of $\mathcal{A}$ in G_{12} by the advantage of an adversary $\mathcal{B}''$ in the OW-CPA game of FrodoPKE. Upon receiving a public-key pk along with a ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ for uniformly random (secret) message $m^* (\leftarrow\$ \{0,1\}^{256})$ and randomness $r^* (\leftarrow\$ \{0,1\}^{256})$ chosen by the challenger, $\mathcal{B}''$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_{12} (e.g., starting with sampling a uniformly random bit $b \leftarrow\$ \{0,1\}$).
- Uses four different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, $G_{(1-b)r}$, G_{0k} and G_{1k} respectively, three different $2q_{H'}$-wise independent functions to simulate the random oracles H_0^{acc}, H_1^{acc} and H_2 respectively, and two different $2(q_G + q_{H'})$-wise independent functions to simulate the random oracles G_{br} and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 1. Also evaluates $\mathcal{A}$'s G- and H-queries using the oracle $G_{br} \times H_3$; the random oracles G and H are simulated in the same way as in G_{12}.
- Answers decapsulation queries using the oracles H_i^{acc} ($i \in \{0,1\}$) as in G_{12}.
- For $\mathcal{A}$'s challenge query, first sets $\mathsf{pk}_b = \mathsf{pk}$. Then generates a key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}$, samples a uniformly random key $k^* \leftarrow\$ \{0,1\}^{256}$ and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$. (By doing this, note that we have $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ in $\mathcal{A}$'s view.)

- Selects $i \leftarrow\!\!\$ \{1, \ldots, q_G + q_{H'}\}$, measures the i-th query to oracle $G_{br} \times H_3$ and returns the outcome m'.

Again, it is not hard to see that $\Pr[\mathsf{G}_{12} = 1] \leq \mathbf{Adv}^{\text{OW-CPA}}_{\text{FrodoPKE}}(\mathcal{B}'')$. From Lemma 7, since we know that IND-CPA security of a PKE scheme with a sufficiently large message space also implies its OW-CPA security, corresponding to adversary $\mathcal{B}''$, there exists an IND-CPA adversary $\mathcal{B}'$ against FrodoPKE such that

$$\mathbf{Adv}^{\text{OW-CPA}}_{\text{FrodoPKE}}(\mathcal{B}'') \leq \mathbf{Adv}^{\text{IND-CPA}}_{\text{FrodoPKE}}(\mathcal{B}') + \frac{1}{2^{256}},$$

where the running time of $\mathcal{B}'$ is that of $\mathcal{B}''$, and $\{0,1\}^{256}$ is the message space of FrodoPKE.

Hence by collecting all of the above bounds, we finally arrive at

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\text{FrodoKEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CPA}}_{\text{FrodoPKE}}(\mathcal{B}) + 2(q_G + q_{H'})\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\text{FrodoPKE}}(\mathcal{B}') + \frac{1}{2^{256}}} + \mathbf{Coll}^{H}_{\text{FrodoPKE}} + \frac{C(q_G+1)^3+1}{2^{256}} + \frac{4q_{H'}}{2^{128}} + 2^{-\gamma} + 8q_G\sqrt{\delta}.$$

□

Regarding the above δ-correctness and γ-spreadness of FrodoPKE assumed in Theorem 10 above, the former property was concretely analyzed in [16, Subsection 2.2.7]. The latter property was also recently analyzed in [8] wherein the authors provided concrete values of γ for different levels of security specified by NIST in the PQC standardization process.[10]

Theorem 11. *For any* SCFR-CCA *adversary $\mathcal{A}$ against the scheme* FrodoKEM $=$ $(\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_G and $q_{H'}$ queries to the quantum random oracles G and H', we have*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\text{FrodoKEM}}(\mathcal{A}) \leq \mathbf{Coll}^{H}_{\text{FrodoPKE}} + \frac{C(q_G+1)^3}{2^{256}} + \frac{C(q_{H'}+1)^3}{2^{256}} + \frac{4q_{H'}}{2^{128}},$$

where C (< 648) is the constant from Lemma 11.

Proof. Denote $\Omega_{\mathbf{G}_2}$, $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$ and $\Omega_{\mathbf{H}'}$ to be the set of all functions $\mathbf{G}_2 : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{G} : \{0,1\}^{256} \to \{0,1\}^{256}$, $\mathbf{H} : \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ and $\mathbf{H}' : \mathcal{C} \to \{0,1\}^{256}$ respectively, where $\mathcal{C}$ is the ciphertext space of

[10] As mentioned in Subsection 3.1.1, we are currently considering "Level 5" security level parameters for FrodoKEM. Nevertheless, our anonymity and robustness analysis of FrodoKEM can be extended to other parameter sets in a straightforward manner.

```
$\mathsf{Games}\ \mathsf{G}_0 - \mathsf{G}_5$
  $(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
  $G_2 \leftarrow\!\!\$\ \Omega_{\mathbf{G}_2}; G_{0r}, G_{1r} \leftarrow\!\!\$\ \Omega_{\mathbf{G}}$
  $G_{0k}, G_{1k} \leftarrow\!\!\$\ \Omega_{\mathbf{G}}$
  $H_2 \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H_0^{\mathsf{rej}}, H_1^{\mathsf{rej}} \leftarrow\!\!\$\ \Omega_{H'}$
  $H_0^{\mathsf{acc}}, H_1^{\mathsf{acc}} \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
  $\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1)$
  $c \leftarrow \mathcal{A}^{G, H', \textsc{Decaps}_\perp}(\mathsf{inp})$
  $k_0 := \textsc{Decaps}_\perp(0, c)$
  $k_1 := \textsc{Decaps}_\perp(1, c)$
  return $[k_0 = k_1 \neq \perp]$
```

```
$\textsc{Decaps}_a(0, c)$   // $c \neq a$
  return $H_0^{\mathsf{acc}}(c)$   // $\mathsf{G}_{4.5}$ - $\mathsf{G}_5$
  Parse $\mathsf{sk}'_0 = (\mathsf{sk}_0, \mathsf{pk}_0, h_0, s_0)$
  $m' := \mathsf{Dec}(\mathsf{sk}_0, c)$
  $(\overline{k}', r') \leftarrow G(m', h_0)$   // $\mathsf{G}_0 - \mathsf{G}_2$
  $r' \leftarrow G_{0r}(m')$   // $\mathsf{G}_3 - \mathsf{G}_4$
  $\overline{k}' \leftarrow G_{0k}(m')$   // $\mathsf{G}_3 - \mathsf{G}_4$
  if $\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c$ then
    return $H'(\overline{k}', c)$
  else return $H'(s_0, c)$   // $\mathsf{G}_0$
  else return $H_0^{\mathsf{rej}}(c)$   // $\mathsf{G}_1$ - $\mathsf{G}_4$
```

```
$G(m, h)$
  if $h = H(\mathsf{pk}_0)$ then   // $\mathsf{G}_2$-$\mathsf{G}_5$
    $r \leftarrow G_{0r}(m)$   // $\mathsf{G}_2$ - $\mathsf{G}_5$
    $\overline{k} \leftarrow G_{0k}(m)$   // $\mathsf{G}_2$ - $\mathsf{G}_5$
  elseif $h = H(\mathsf{pk}_1)$ then   // $\mathsf{G}_2$-$\mathsf{G}_5$
    $r \leftarrow G_{1r}(m)$   // $\mathsf{G}_2$ - $\mathsf{G}_5$
    $\overline{k} \leftarrow G_{1k}(m)$   // $\mathsf{G}_2$ - $\mathsf{G}_5$
  else $(\overline{k}, r) \leftarrow G_2(m, h)$
  return $(\overline{k}, r)$
```

```
$H'(\overline{k}, c)$
  $m' := \mathsf{Dec}(\mathsf{sk}_0, c)$   // $\mathsf{G}_4 - \mathsf{G}_5$
  if $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c \wedge$
   $G_{0k}(m') = \overline{k}$   // $\mathsf{G}_4 - \mathsf{G}_5$
    return $H_0^{\mathsf{acc}}(c)$   // $\mathsf{G}_4 - \mathsf{G}_5$
  $m' := \mathsf{Dec}(\mathsf{sk}_1, c)$   // $\mathsf{G}_4 - \mathsf{G}_5$
  if $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c \wedge$
   $G_{1k}(m') = \overline{k}$   // $\mathsf{G}_4 - \mathsf{G}_5$
    return $H_1^{\mathsf{acc}}(c)$   // $\mathsf{G}_4 - \mathsf{G}_5$
  return $H_2(\overline{k}, c)$
```

```
$\textsc{Decaps}_a(1, c)$   // $c \neq a$
  return $H_1^{\mathsf{acc}}(c)$   // $\mathsf{G}_5$
  Parse $\mathsf{sk}'_1 = (\mathsf{sk}_1, \mathsf{pk}_1, h_1, s_1)$
  $m' := \mathsf{Dec}(\mathsf{sk}_1, c)$
  $(\overline{k}', r') \leftarrow G(m', h_1)$   // $\mathsf{G}_0 - \mathsf{G}_2$
  $r' \leftarrow G_{1r}(m')$   // $\mathsf{G}_3 - \mathsf{G}_{4.5}$
  $\overline{k}' \leftarrow G_{1k}(m')$   // $\mathsf{G}_3 - \mathsf{G}_{4.5}$
  if $\mathsf{Enc}(\mathsf{pk}_1, m'; r') = c$ then
    return $H'(\overline{k}', c)$
  else return $H'(s_1, c)$   // $\mathsf{G}_0$
  else return $H_1^{\mathsf{rej}}(c)$   // $\mathsf{G}_1$ - $\mathsf{G}_{4.5}$
```

FIGURE 5.5: Games $\mathsf{G}_0 - \mathsf{G}_5$ for the proof of Theorem 11.

FrodoPKE/FrodoKEM. Let $\mathcal{A}$ be an adversary in the SCFR-CCA game for FrodoKEM issuing at most q_G and $q_{H'}$ quantum queries to the random oracles G and H' respectively.

The structure of the proof is similar to that of Theorem 10. Basically we do a similar sequence of game-hops as in the proof of Theorem 10 until the point where we can simulate the decapsulation oracles $\textsc{Decaps}_\perp(i,\cdot)$ $(i \in \{0,1\})$ without requiring the corresponding secret keys sk'_i.

To be specific, we do the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_5$ as described in Figure 5.5. By a similar analysis as that in the proof of Theorem 10 w.r.t. these game-hops, it is not hard to obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \mathbf{Coll}^H_{\text{FrodoPKE}} + \frac{C(q_G+1)^3}{2^{256}} + \frac{4q_{H'}}{2^{128}}.$$

Note that the game G_0 is exactly the SCFR-CCA game for FrodoKEM. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\text{FrodoKEM}}(\mathcal{A}).$$

Coming to game G_5, note that the adversary $\mathcal{A}$ wins the game if it outputs a ciphertext c such that $\textsc{Decaps}_\perp(0,c) = \textsc{Decaps}_\perp(1,c)$. Because of the modification of $\textsc{Decaps}_\perp(i,\cdot)$ oracles, this winning condition translates to $H_0^{\text{acc}}(c) = H_1^{\text{acc}}(c)$, where H_0^{acc} and H_1^{acc} are independent quantum random oracles. Note that in this case, (c,c) is a *claw* w.r.t. the pair of QROs $H_0^{\text{acc}} : \mathcal{C} \to \{0,1\}^{256}$ and $H_1^{\text{acc}} : \mathcal{C} \to \{0,1\}^{256}$. Hence we can bound the success probability of $\mathcal{A}$ in G_5 by the advantage of an adversary $\mathcal{B}$ against the *claw-finding* game w.r.t. the instance (H_0^{acc}, H_1^{acc}). $\mathcal{B}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_5, by creating the appropriate setup (starting with the generation of two honest key-pairs $(\mathsf{pk}_0, \mathsf{sk}'_0)$ and $(\mathsf{pk}_1, \mathsf{sk}'_1)$).

- Uses five different $2q_G$-wise independent functions to perfectly simulate random oracles G_2, G_{0r}, G_{1r}, G_{0k} and G_{1k} respectively, and a $2q_{H'}$-wise independent function to simulate random oracle H_2 in $\mathcal{A}$'s view, as noted in Lemma 1. Also uses the pair of oracles $f_0 : \mathcal{C} \to \{0,1\}^{256}$ and $f_1 : \mathcal{C} \to \{0,1\}^{256}$ – which is the instance of the claw-finding game – to simulate oracles H_0^{acc} and H_1^{acc} respectively.

- The random oracles G and H' are simulated in the same way as in G_5 (e.g., note that H' can be simulated as the claw-finding adversary $\mathcal{D}$ possesses the secret keys sk_0 and sk_1).

- Answers decapsulation queries using the oracles $f_i(\cdot)$ $(i \in \{0, 1\})$ as in G_5.
- After obtaining a ciphertext c from $\mathcal{A}$, forwards (c, c) as the claw w.r.t. (f_0, f_1).

Note that $\mathcal{B}$ makes at most $q_{H'}$ queries to the pair (f_0, f_1). It is easy to see that $\Pr[\mathsf{G}_5 = 1] \leq \frac{C(q_{H'}+1)^3}{2^{256}}$ from Lemma 11. Hence, we finally get

$$\mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathrm{FrodoKEM}}(\mathcal{A}) \leq \mathbf{Coll}^{H}_{\mathrm{FrodoPKE}} + \frac{C(q_G+1)^3}{2^{256}} + \frac{C(q_{H'}+1)^3}{2^{256}} + \frac{4q_{H'}}{2^{128}}.$$

□

5.2.3 *Anonymity and Robustness of Hybrid PKE Derived from FrodoKEM*

Regarding hybrid PKE schemes obtained from FrodoKEM via the KEM-DEM composition, we additionally show that such PKE schemes satisfy the stronger ANO-CCA notion of anonymity, in a similar vein to Theorem 9 w.r.t. $\mathsf{FO}^{\not\perp}$-based KEMs.

Theorem 12. *Let* $\mathrm{FrodoKEM}^{\mathrm{hy}} = (\mathsf{KGen}^{\mathrm{hy}}, \mathsf{Enc}^{\mathrm{hy}}, \mathsf{Dec}^{\mathrm{hy}})$ *be a hybrid PKE scheme obtained by composing* $\mathrm{FrodoKEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *with a one-time secure AE scheme* $\mathsf{DEM} = (\mathsf{KGen}^{\mathrm{dem}}, \mathsf{Enc}^{\mathrm{dem}}, \mathsf{Dec}^{\mathrm{dem}})$*. Given the base scheme* $\mathrm{FrodoPKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *underlying* FrodoKEM *is δ-correct and γ-spread, then for any* ANO-CCA *adversary $\mathcal{A}_{\mathrm{hy}}$ against* $\mathrm{FrodoKEM}^{\mathrm{hy}}$ *issuing at most q_G and $q_{H'}$ queries to the quantum random oracles G and H' respectively, there exist* ANO-CCA *adversary $\mathcal{A}_{\mathrm{kem}}$ and* IND-CCA *adversary $\overline{\mathcal{A}}_{\mathrm{kem}}$ against* FrodoKEM *and* INT-CTXT *adversary $\mathcal{A}_{\mathrm{dem}}$ against* DEM *such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathrm{FrodoKEM}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathrm{FrodoKEM}}(\mathcal{A}_{\mathrm{kem}}) + 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathrm{FrodoKEM}}(\overline{\mathcal{A}}_{\mathrm{kem}}) + 2^{-\gamma} \\ + 2\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) + 2\mathbf{Coll}^{H}_{\mathrm{FrodoPKE}} + \frac{4q_{H'}}{2^{128}} + 8q_G\sqrt{\delta},\end{aligned}$$

where $\mathbf{Coll}^{H}_{\mathrm{FrodoPKE}}$ is probability of the event "$H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$" with pk_0 and pk_1 being two honestly-generated public keys of FrodoPKE*. Also the running times of $\mathcal{A}_{\mathrm{kem}}$, $\overline{\mathcal{A}}_{\mathrm{kem}}$ and $\mathcal{A}_{\mathrm{dem}}$ are the same as that of $\mathcal{A}_{\mathrm{hy}}$.*

Proof. The proof is similar to that of Theorem 9, except for some initial game-hops. Here we will focus on these hops.

Games G_0 - G_3

```
$(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}$
$s_0 \leftarrow\!\!\$ \{0,1\}^{256}; s_1 \leftarrow\!\!\$ \{0,1\}^{256}$
$\mathsf{sk}_0' = (\mathsf{sk}_0, \mathsf{pk}_0, H(\mathsf{pk}_0), s_0)$
$\mathsf{sk}_1' = (\mathsf{sk}_1, \mathsf{pk}_1, H(\mathsf{pk}_1), s_1)$
$G_2 \leftarrow\!\!\$ \Omega_{\mathbf{G}_2}; H' \leftarrow\!\!\$ \Omega_{\mathbf{H}}$
$G_{0r}, G_{1r} \leftarrow\!\!\$ \Omega_{\mathbf{G}}$   // $\mathsf{G}_1 - \mathsf{G}_3$
$G_{0k}, G_{1k} \leftarrow\!\!\$ \Omega_{\mathbf{G}}$   // $\mathsf{G}_1 - \mathsf{G}_3$
$b \leftarrow\!\!\$ \{0,1\}$
$m^* \leftarrow\!\!\$ \{0,1\}^{256}$
$(\overline{k}^*, r^*) \leftarrow\!\!\$ G(m^*, H(\mathsf{pk}_b))$   // $\mathsf{G}_0 - \mathsf{G}_1$
$r^* \leftarrow G_{br}(m^*)$   // $\mathsf{G}_2 - \mathsf{G}_3$
$\overline{k}^* \leftarrow G_{bk}(m^*)$   // $\mathsf{G}_2 - \mathsf{G}_3$
$c_0^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
$k^* \leftarrow H'(\overline{k}^*, c_0^*)$
$k^{\mathsf{rej}} \leftarrow H'(s_{1-b}, c_0^*)$   // $\mathsf{G}_3$
$(m^{\mathrm{hy}}, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{G, H', \textsc{Dec}_{\perp}^{\mathrm{hy}}}(\mathsf{pk}_0, \mathsf{pk}_1)$
$c_1^* \leftarrow \mathsf{Enc}^{\mathsf{dem}}(k^*, m^{\mathrm{hy}})$
$c^* = (c_0^*, c_1^*)$
$b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{G, H', \textsc{Dec}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
return $[b' = b]$
```

$G(m, h)$

```
if $h = H(\mathsf{pk}_0)$ then   // $\mathsf{G}_1 - \mathsf{G}_3$
    $r \leftarrow G_{0r}(m)$   // $\mathsf{G}_1 - \mathsf{G}_3$
    $\overline{k} \leftarrow G_{0k}(m)$   // $\mathsf{G}_1 - \mathsf{G}_3$
elseif $h = H(\mathsf{pk}_1)$ then   // $\mathsf{G}_1 - \mathsf{G}_3$
    $r \leftarrow G_{1r}(m)$   // $\mathsf{G}_1 - \mathsf{G}_3$
    $\overline{k} \leftarrow G_{1k}(m)$   // $\mathsf{G}_1 - \mathsf{G}_3$
else $(\overline{k}, r) \leftarrow G_2(m, h)$
return $(\overline{k}, r)$
```

$\textsc{Dec}_a^{\mathrm{hy}}(b, c)$ // $c \neq a$

```
Parse $c = (c_0, c_1)$
Parse $\mathsf{sk}_b' = (\mathsf{sk}_b, \mathsf{pk}_b, h_b, s_b)$
$m' := \mathsf{Dec}(\mathsf{sk}_b, c_0)$
$(\overline{k}', r') \leftarrow G(m', h_b)$   // $\mathsf{G}_0 - \mathsf{G}_1$
$r' \leftarrow G_{br}(m')$   // $\mathsf{G}_2 - \mathsf{G}_3$
$\overline{k}' \leftarrow G_{bk}(m')$   // $\mathsf{G}_2 - \mathsf{G}_3$
if $\mathsf{Enc}(\mathsf{pk}_b, m'; r') = c_0$
    $k' \leftarrow H'(\overline{k}', c_0)$
else $k' \leftarrow H'(s_b, c_0)$
$m^{\mathrm{hy}'} := \mathsf{Dec}^{\mathsf{dem}}(k', c_1)$
return $m^{\mathrm{hy}'}$
```

$\textsc{Dec}_a^{\mathrm{hy}}(1 - b, c)$ // $c \neq a$

```
Parse $c = (c_0, c_1)$ and
  $\mathsf{sk}_{1-b}' = (\mathsf{sk}_{1-b}, \mathsf{pk}_{1-b}, h_{1-b}, s_{1-b})$
if $c_0 = c_0^*$ then   // $\mathsf{G}_3$
    $k' := k^{\mathsf{rej}}$   // $\mathsf{G}_3$
else   // $\mathsf{G}_3$
    $m' := \mathsf{Dec}(\mathsf{sk}_{1-b}, c_0)$
    $(\overline{k}', r') \leftarrow G(m', h_{1-b})$
    $r' \leftarrow G_{(1-b)r}(m')$   // $\mathsf{G}_2 - \mathsf{G}_3$
    $\overline{k}' \leftarrow G_{(1-b)k}(m')$   // $\mathsf{G}_2 - \mathsf{G}_3$
    if $\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c_0$
        $k' \leftarrow H'(\overline{k}', c_0)$
    else $k' \leftarrow H'(s_{1-b}, c_0)$
$m^{\mathrm{hy}'} := \mathsf{Dec}^{\mathsf{dem}}(k', c_1)$
return $m^{\mathrm{hy}'}$
```

FIGURE 5.6: Games G_0 – G_3 for the proof of Theorem 12.

Denote $\Omega_{\mathbf{G}_2}$, $\Omega_{\mathbf{G}}$ and $\Omega_{\mathbf{H}}$ to be the set of all functions $\mathbf{G}_2 : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{G} : \{0,1\}^{256} \to \{0,1\}^{256}$ and $\mathbf{H} : \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ respectively. Let $\mathcal{A}_{\text{hy}}$ be an adversary in the ANO-CCA game for FrodoKEM$^{\text{hy}}$ issuing at most q_G and $q_{H'}$ quantum queries to the random oracles G and H respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_3$ described in Figure 5.6.

Game G_0**:** The game G_0 is equivalent to the ANO-CCA game for FrodoKEM$^{\text{hy}}$, except for some "cosmetic" changes. Namely, the pair (c_0^*, k^*) resulting from running $\mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b is generated *before* the adversary $\mathcal{A}_{\text{hy}}$ gets to choose a message m^{hy}. This change does not affect $\mathcal{A}_{\text{hy}}$'s view in any way. Hence,

$$\left| \Pr[\mathsf{G}_0 = 1] - \frac{1}{2} \right| = \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\text{FrodoKEM}^{\text{hy}}}(\mathcal{A}_{\text{hy}}).$$

Game G_1**:** In game G_1, we implicitly divide the G-queries into at most three categories: (1) query is of the form (m, h) with $h = H(\mathsf{pk}_0)$, (2) query is of the form (m, h) with $h = H(\mathsf{pk}_1)$, and (3) the remaining queries. We then respond to the queries from the respective categories with $(G_{0k}(m), G_{0r}(m))$, $(G_{1k}(m), G_{1r}(m))$ and $G_2(m, h)$ respectively, where G_{ik}, G_{ir} (for $i \in \{0,1\}$) are internal random oracles; note that we say "at most" three categories because of the (unlikely) possibility that $H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$. It is not hard to verify that output distributions of the G-oracle in games G_0 and G_1 are equivalent. Therefore,

$$\Pr[\mathsf{G}_1 = 1] = \Pr[\mathsf{G}_0 = 1].$$

Game G_2**:** In game G_2, we make the following changes w.r.t. the G-oracle evaluation. First, we generate the values $\overline{k}^*, r^*$ in setup of the game as "$\overline{k}^* \leftarrow G_{bk}(m^*)$" and "$r^* \leftarrow G_{br}(m^*)$" (effectively replacing the step "$(\overline{k}^*, r^*) \leftarrow G(m^*, H(\mathsf{pk}_b))$" in G_1). We then similarly generate the values $\overline{k}', r'$ w.r.t. the decryption oracles $\textsc{Dec}_a^{\text{hy}}(i, \cdot)$ ($i \in \{0,1\}$) as "$\overline{k}' \leftarrow G_{ik}(m')$" and "$r' \leftarrow G_{ir}(m')$" (replacing the step "$(\overline{k}', r') \leftarrow G(m', h_i)$" in G_1, where $h = H(\mathsf{pk}_i)$).

Let "bad" denote the event where the public keys pk_0 and pk_1 generated honestly in the setup satisfy "$H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$". It is not hard to see that the games G_1 and G_2 are equivalent unless the event bad happens. As seen in the proof of Theorem 10 (specifically, the "$\mathsf{G}_2 \to \mathsf{G}_3$" hop), we also have the probability of the event bad occurring to be $\Pr[\mathsf{bad}] \leq \mathbf{Coll}^{H}_{\text{FrodoPKE}}$. Hence, we get

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \Pr[\mathsf{bad}] \leq \mathbf{Coll}^{H}_{\mathrm{FrodoPKE}}.$$

Game G_3**:** In game G_3, we modify the oracle $\textsc{Dec}^{\mathrm{hy}}_a(1-b,\cdot)$ such that if the decryption query is (c_0, c_1) where $c_0 = c_0^*$, then the oracle uses key $k^{\mathsf{rej}}(= H'(s_{1-b}, c_0^*))$ to decrypt c_1. Here k^{rej} is the key returned if $\mathsf{Decap}(\mathsf{sk}'_{1-b}, c_0^*)$ would have resulted in an "implicit rejection". Thus, it is not hard to see that the games G_2 and G_3 are equivalent unless c_0^* is not (implicitly) rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b},\cdot)$ operation, or in other words, if the following event occurs: "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c_0^*$" where for a uniformly random message $m^* \leftarrow\$ \{0,1\}^{256}$ we have $(\overline{k}^*, r^*) \leftarrow\$ (G_{bk}(m^*), G_{br}(m^*))$, $\mathsf{Enc}(\mathsf{pk}_b, m^*; r^*) = c_0^*$, $\mathsf{Dec}(\mathsf{sk}_{1-b}, c_0^*) = m'$ and $(\overline{k}', r') \leftarrow\$ (G_{(1-b)k}(m'), G_{(1-b)r}(m'))$.

The analysis that follows is quite similar to the "$\mathsf{G}_5 \to \mathsf{G}_6$" game-hop in the proof of Theorem 10. Note that in the context of an experiment describing the above event at the setup, we have $G_{(1-b)r}(m')$ resulting in uniformly random coins $r' \leftarrow\$ \{0,1\}^{256}$, since G_{br} is used to compute the ciphertext c_1^* and $G_{(1-b)r}$ is a random oracle independent to G_{br}. Since FrodoPKE is γ-spread, for the key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b})$ and message m', we have the condition "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c_0^*$" to hold with probability $\leq 2^{-\gamma}$ for uniformly random r'. Hence, we have

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq 2^{-\gamma}.$$

The rest of the proof follows very similarly to that of Theorem 9, where we then do the game-hop "$\mathsf{G}_0 \to \mathsf{G}_1$" of Theorem 9, skip the "$\mathsf{G}_1 \to \mathsf{G}_2$" hop – since effectively this is covered by our above $\mathsf{G}_0 \to \mathsf{G}_3$ hop – and proceed from "G_3" of Theorem 9 from then on, and so on.

Hence, it is not hard to finally arrive at

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathrm{FrodoKEM}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}) \leq{}& \mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathrm{FrodoKEM}}(\mathcal{A}_{\mathrm{kem}}) + 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathrm{FrodoKEM}}(\overline{\mathcal{A}}_{\mathrm{kem}}) + 2^{-\gamma} \\ &+ 2\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{DEM}}(\mathcal{A}_{\mathrm{dem}}) + 2\mathbf{Coll}^{H}_{\mathrm{FrodoPKE}} + \frac{4q_{H'}}{2^{128}} + 8q_G\sqrt{\delta}.\end{aligned}$$

□

At the same time, from Theorems 4 and 11, we note that if the DEM component is also FROB secure, then the corresponding hybrid PKE scheme will be strongly robust (i.e., SROB-CCA secure). Hence, our results in this section give a complete picture of anonymity and robustness properties of FrodoKEM as well as the hybrid PKE schemes derived from it.

5.3 KYBER

In this section, we establish anonymity and robustness of Kyber – and the hybrid PKE schemes derived from it – in the QROM. Focusing on anonymity, as mentioned in Subsection 4.5.1 above, we will work with Xagawa's *strong pseudorandomness* framework [60] which was used to establish the corresponding property for NTRU [23] – a third-round NIST PQC finalist that employs the $\mathsf{FO}_m^{\not\perp}$ transform.

To be more specific, strong pseudorandomness (i.e., SPR-CCA security) of "$\mathsf{FO}_m^{\not\perp}$-derived" KEMs in the QROM was already shown in [60]. So at a high level, we will apply our above "wrapper-based" approach – that was used to extend IND-CCA security properties of the $\mathsf{FO}_m^{\not\perp}$ transform to $\mathsf{FO}^{\mathsf{kyber}}$ (see Fig. 3.6) in Section 3.2 – to prove SPR-CCA security of Kyber, a.k.a. $\mathsf{Kyber.KEM}$, in the QROM. And as shown by Xagawa [60], SPR-CCA security of a KEM/PKE scheme also implies its ANO-CCA security.

One of the advantages of using this "$\mathsf{FO}_m^{\not\perp}$-based" approach to establish anonymity of Kyber when compared to an "$\mathsf{FO}^{\not\perp}$-based" approach used for FrodoKEM in the previous section is that the strong pseudorandomness framework does not rely on γ-spreadness of the base PKE scheme. This fact is helpful because, to the best of our knowledge, γ-spreadness of Kyber's base PKE scheme, a.k.a. $\mathsf{Kyber.PKE}$, has not been rigorously analyzed in the literature – in contrast to FrodoKEM [8].

When it comes to establishing robustness – or more technically, collision-freeness – of Kyber, we will employ similar techniques that were used to establish the corresponding property for FrodoKEM in Section 5.2 above; namely, we will rely on hardness of the claw-finding problem w.r.t. quantum random oracles (see Lemma 11).

5.3.1 *Some More Preliminaries*

Here we describe some additional security notions and relevant theorems – related to Xagawa's *strong pseudorandomness* framework discussed above – that will be primarily used to establish ANO-CCA security of Kyber, and corresponding hybrid PKE schemes, in the QROM.

5.3.1.1 *Public-Key Encryption, Revisited (Again)*

Definition 23 (Strong Pseudorandomness of PKE). *Given a PKE* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*, we define the game for its* SPR-CCA security, *w.r.t. a simulator*

```
$\mathsf{SPR\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE},\mathcal{S}}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$ \{0,1\}$
$(m,\mathsf{st}) \leftarrow \mathcal{A}^{\textsc{Dec}_{\perp}}(\mathsf{pk})$
$c_0^* \leftarrow \mathsf{Enc}(\mathsf{pk},m)$
$c_1^* \leftarrow \mathcal{S}()$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}}(c_b^*,\mathsf{st})$
return $[b' = b]$
```

```
$\mathsf{SDS\text{-}IND}^{\mathcal{A}}_{\mathsf{PKE},\mathcal{S}}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$ \{0,1\}$
$m \leftarrow\!\!\$ \mathcal{M}$
$c_0^* \leftarrow \mathsf{Enc}(\mathsf{pk},m)$
$c_1^* \leftarrow \mathcal{S}()$
$b' \leftarrow \mathcal{A}(\mathsf{pk},c_b^*)$
return $[b' = b]$
```

```
$\textsc{Dec}_a(c)$
if $c = a$ then return $\perp$
$m := \mathsf{Dec}(\mathsf{sk},c)$
return $m$
```

FIGURE 5.7: Security games for strong pseudorandomness and strong disjoint simulatability of PKE schemes. Note that these games are defined with respect to a (efficient) simulator $\mathcal{S}$. Also st is some state information maintained by the adversary $\mathcal{A}$.

$\mathcal{S}$, in Figure 5.7 and the SPR-CCA advantage measure *for adversary $\mathcal{A}$ against* PKE *(and $\mathcal{S}$) as*

$$\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{A}) = \left|\Pr[\mathsf{SPR\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE},\mathcal{S}} = 1] - \frac{1}{2}\right|.$$

Asymptotically speaking, PKE *is said to be* strongly pseudorandom *if there exists an efficient simulator $\mathcal{S}$ such that $\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{A})$ is negligible for any efficient adversary $\mathcal{A}$ w.r.t. a security parameter.*

Definition 24 (Strong Disjoint Simulatability of PKE [6, 85]). *Given a PKE* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, *with message space $\mathcal{M}$ and encryption randomness space $\mathcal{R}$, we define the game for its* SDS-IND security *– w.r.t. a simulator $\mathcal{S}$ – in Figure 5.7 and the* SDS-IND advantage measure *for adversary $\mathcal{A}$ against* PKE *(and $\mathcal{S}$) as*

$$\mathbf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{A}) = \left|\Pr[\mathsf{SDS\text{-}IND}^{\mathcal{A}}_{\mathsf{PKE},\mathcal{S}} = 1] - \frac{1}{2}\right|.$$

In addition, we define the (statistical) disjointness measure *as*

$$\begin{aligned}\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}} = \Pr[\exists (m,r) \in \mathcal{M} \times \mathcal{R} \text{ s.t. } c := \mathsf{Enc}(\mathsf{pk},m;r) \\ \mid (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}, c \leftarrow \mathcal{S}()].\end{aligned}$$

```
$\mathsf{SPR\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM},\mathcal{S}}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$\ \{0,1\}$
$(c_0^*, k_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$
$c_1^* \leftarrow \mathcal{S}()$
$k_1^* \leftarrow\!\!\$\ \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c_b^*}}(\mathsf{pk}, c_b^*, k_b^*)$
$\textbf{return } [b' = b]$
```

```
$\mathsf{SSMT\text{-}CCA}^{\mathcal{A}}_{\mathsf{KEM},\mathcal{S}}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$
$b \leftarrow\!\!\$\ \{0,1\}$
$c^* \leftarrow \mathcal{S}()$
$k_0^* \leftarrow\!\!\$\ \mathcal{K}$
$k_1^* := \mathsf{Decap}(\mathsf{sk}, c^*)$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k_b^*)$
$\textbf{return } [b' = b]$
```

```
$\textsc{Decaps}_a(c)$
$\textbf{if } c = a \textbf{ then return } \perp$
$k := \mathsf{Decap}(\mathsf{sk}, c)$
$\textbf{return } k$
```

FIGURE 5.8: Security games for strong pseudorandomness and strong smoothness of KEMs. Note that these games are defined with respect to a (efficient) simulator $\mathcal{S}$.

Asymptotically speaking, PKE *is said to be* strongly disjoint simulatable *if there exists an efficient simulator $\mathcal{S}$ such that $\mathbf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{A})$ is negligible for any efficient adversary $\mathcal{A}$ and $\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}$ is negligible w.r.t. a security parameter.*[11]

5.3.1.2 *Key Encapsulation Mechanism, Revisited*

Definition 25 (Strong Pseudorandomness and Smoothness of KEM). *Given a KEM* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *with encapsulated key space $\mathcal{K}$, we define the game for its* SPR-CCA, *resp.* SSMT-CCA, security *w.r.t. a simulator $\mathcal{S}$ in Figure 5.8 and the* SPR-CCA, *resp.* SSMT-CCA, advantage measure *for adversary $\mathcal{A}$ against* KEM *(and $\mathcal{S}$) as*

$$\mathbf{Adv}^{(\mathsf{SPR/SSMT})\text{-}\mathsf{CCA}}_{\mathsf{KEM},\mathcal{S}}(\mathcal{A}) = \left| \Pr[(\mathsf{SPR/SSMT})\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{KEM},\mathcal{S}} = 1] - \frac{1}{2} \right| .$$

Asymptotically speaking, KEM *is said to be* strongly pseudorandom *(respectively,* strongly smooth*) if there exists an efficient simulator $\mathcal{S}$ such that $\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{KEM},\mathcal{S}}(\mathcal{A})$ (respectively, $\mathbf{Adv}^{\mathsf{SSMT\text{-}CCA}}_{\mathsf{KEM},\mathcal{S}}(\mathcal{A})$) is negligible for any efficient adversary $\mathcal{A}$ w.r.t. a security parameter.*

```
$\mathsf{otSPR\text{-}CCA}^{\mathcal{A}}_{\mathsf{DEM}}$
$k \leftarrow \mathsf{KGen}$
$b \leftarrow\$ \{0,1\}$
$(m, \mathsf{st}) \leftarrow \mathcal{A}^{\textsc{Dec}_{\perp}}$
$c_0^* \leftarrow \mathsf{Enc}(k, m)$
$c_1^* \leftarrow\$ \mathcal{C}_{|m|}$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}}(c_b^*, \mathsf{st})$
return $[b' = b]$

$\textsc{Dec}_a(c)$
if $c = a$ then return $\perp$
$m := \mathsf{Dec}(\mathsf{sk}, c)$
return $m$
```

FIGURE 5.9: Security game for one-time strong pseudorandomness of DEMs. Here st is some state information maintained by the adversary $\mathcal{A}$.

5.3.1.3 *Data Encapsulation Mechanism, Revisited (Again)*

Definition 26 (One-time Strong Pseudorandomness of DEM). *Given a DEM* $\mathsf{DEM} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *with ciphertext "subspace"* $\mathcal{C}_\ell$ *associated with encryptions of messages with length* ℓ*, we define the game for its* one-time SPR-CCA (otSPR-CCA) security *in Figure 5.9 and the* otSPR-CCA advantage measure *for adversary* $\mathcal{A}$ *against* DEM *as*

$$\mathbf{Adv}^{\mathsf{otSPR\text{-}CCA}}_{\mathsf{DEM}}(\mathcal{A}) = \left| \Pr[\mathsf{otSPR\text{-}CCA}^{\mathcal{A}}_{\mathsf{DEM}} = 1] - \frac{1}{2} \right|.$$

5.3.1.4 *Useful Theorems*

As briefly mentioned in Subsection 4.5.1 above, Xagawa showed in [60] that SPR-CCA security of KEMs and PKE schemes implies their ANO-CCA security as well. We now present a more formal statement.

Theorem 13 ([60, Theorem 2.5]). *Let* KEM *be a KEM and* PKE *be a PKE scheme. Then for any* ANO-CCA *adversary* $\mathcal{A}_{\text{kem}}$ *against* KEM *(resp.,* $\mathcal{A}_{\text{pke}}$ *against* PKE*), and any simulator* $\mathcal{S}_{\text{kem}}$ *w.r.t.* KEM *(resp.,* $\mathcal{S}_{\text{pke}}$ *w.r.t.* PKE*), there*

[11] As described in [6], SDS-IND security is a measure of "*ciphertext indistinguishability*" of PKE. Hence informally, *strong disjoint simulatability* can be seen as *ciphertext indistinguishability* "plus" *statistical disjointness*.

exist SPR-CCA *adversaries* $\mathcal{B}_{\text{kem}}$ *and* $\mathcal{B}'_{\text{kem}}$ *against* KEM *(resp.,* $\mathcal{B}_{\text{pke}}$ *and* $\mathcal{B}'_{\text{pke}}$ *against* PKE*) running in about the same time as* $\mathcal{A}_{\text{kem}}$ *(resp.,* $\mathcal{A}_{\text{pke}}$*) such that*

$$\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}_{\text{kem}}) \leq \mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{KEM},\mathcal{S}_{\text{kem}}}(\mathcal{B}_{\text{kem}}) + \mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{KEM},\mathcal{S}_{\text{kem}}}(\mathcal{B}'_{\text{kem}}),$$
$$\mathbf{Adv}^{\mathsf{ANO\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}_{\text{pke}}) \leq \mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{PKE},\mathcal{S}_{\text{pke}}}(\mathcal{B}_{\text{pke}}) + \mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{PKE},\mathcal{S}_{\text{pke}}}(\mathcal{B}'_{\text{pke}}).$$

The following theorem establishes SPR-CCA security of hybrid PKE schemes from the corresponding pseudorandom properties of the underlying implicit rejection KEM and DEM wherein the KEM is additionally SSMT-CCA secure.

Theorem 14 ([60, Theorem 3.2]). *Let* $\mathsf{PKE}^{\text{hy}} = (\mathsf{KGen}^{\text{hy}}, \mathsf{Enc}^{\text{hy}}, \mathsf{Dec}^{\text{hy}})$ *be a hybrid PKE scheme obtained by composing a* δ*-correct KEM* $\mathsf{KEM} = (\mathsf{KGen}^{\text{kem}}, \mathsf{Encap}, \mathsf{Decap})$ *with a DEM* $\mathsf{DEM} = (\mathsf{KGen}^{\text{dem}}, \mathsf{Enc}, \mathsf{Dec})$*. Then for any simulator* $\mathcal{S}_{\text{kem}}$ *w.r.t.* KEM*, there exists a simulator* $\mathcal{S}_{\text{hy}}$ *w.r.t.* PKE^{hy} *where for any* SPR-CCA *adversary* $\mathcal{A}_{\text{hy}}$ *against* PKE^{hy}*, there exist* SPR-CCA *adversary* $\mathcal{A}_{\text{kem}}$ *and* SSMT-CCA *adversary* $\overline{\mathcal{A}}_{\text{kem}}$ *against* KEM*, and* otSPR-CCA *adversary* $\mathcal{A}_{\text{dem}}$ *against* DEM *such that*

$$\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{PKE}^{\text{hy}},\mathcal{S}_{\text{hy}}}(\mathcal{A}_{\text{hy}}) \leq \mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{KEM},\mathcal{S}_{\text{kem}}}(\mathcal{A}_{\text{kem}}) + \mathbf{Adv}^{\mathsf{SSMT\text{-}CCA}}_{\mathsf{KEM},\mathcal{S}_{\text{kem}}}(\overline{\mathcal{A}}_{\text{kem}})$$
$$+ \mathbf{Adv}^{\mathsf{otSPR\text{-}CCA}}_{\mathsf{DEM}}(\mathcal{A}_{\text{dem}}) + \delta.$$

The running times of $\mathcal{A}_{\text{kem}}$*,* $\overline{\mathcal{A}}_{\text{kem}}$ *and* $\mathcal{A}_{\text{dem}}$ *are the same as that of* $\mathcal{A}_{\text{hy}}$*.*

Remark 3. Note that in the above theorem, we consider SPR-CCA security and SSMT-CCA security of KEM w.r.t. the *same* simulator $\mathcal{S}_{\text{kem}}$. If KEM is shown to be SPR-CCA and SSMT-CCA secure w.r.t. different simulators, then Theorem 14 does not provide any guarantee on the SPR-CCA security of PKE^{hy}. Fortunately, in our analysis of Kyber that follows, we prove SPR-CCA and SSMT-CCA security of $\mathsf{Kyber.KEM}$ w.r.t. the same simulator – thereby establishing SPR-CCA security of hybrid PKE schemes derived from the NIST PQC standard.

The theorem below establishes SPR-CCA security of "$\mathsf{FO}_m^{\not\perp}$-derived" KEMs (see Fig. 3.1) in the QROM based on strong disjoint-simulatability of the underlying base PKE scheme.

Theorem 15 (Follows from [60, Theorems 4.1 and D.1][12]). *Given* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, has message space* $\mathcal{M}$ *and ciphertext space* $\mathcal{C}$ *which*

[12] $\mathsf{FO}_m^{\not\perp}$ is composed of two *modular* FO transforms: namely, the "T" and "$\mathsf{U}_m^{\not\perp}$" transforms defined in [4]; [60, Theorem D.1] considers the T transform and [60, Theorem 4.1] considers the $\mathsf{U}_m^{\not\perp}$ transform respectively.

only depends on the public parameters.[13] *Then for any* SPR-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\not\perp} = \mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, G_r, G_k] = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_{G_r} *and* q_{G_k} *queries to the quantum random oracles* G_r *and* G_k *respectively and at most* q_D *queries to the (classical) decapsulation oracle, and for any simulator* $\mathcal{S}$ *w.r.t.* PKE*, there exist* SDS-IND *adversary* $\mathcal{B}$ *and* OW-CPA *adversary* $\mathcal{B}'$ *against* PKE *such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{KEM}^{\not\perp},\mathcal{S}}(\mathcal{A}) \le \mathbf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{B}) + \frac{1}{2}\mathsf{Disj}_{\mathsf{PKE}_1,\mathcal{S}} + q_{G_r}\sqrt{\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}')} \\ + \frac{2(q_{G_k}+q_D)}{\sqrt{|\mathcal{M}|}} + (2 + 8(q_{G_r}+q_D+2)^2 + 8(q_{G_r}+q_{G_k}+2)^2)\delta,\end{aligned}$$

where we have the derandomized PKE scheme $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G_r]$ *(see Fig. 4.9). Moreover, the running times of* $\mathcal{B}$ *and* $\mathcal{B}'$ *are the same as that of* $\mathcal{A}$*.*

5.3.2 *Strong Disjoint Simulatability of* Kyber.PKE

In our following ANO-CCA security analysis of Kyber.KEM and the hybrid PKE schemes derived from it, we rely on the *strong disjoint simulatability* (i.e., *SDS-IND security* plus *statistical disjointness*, see Definition 24 above) of the base Kyber.PKE scheme. Fortunately, we have:

Lemma 13 (informal)**.** $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is tightly* strong disjoint simulatable *under the* MLWE *hardness assumption, in the QROM.*

Proof Sketch. Let $\mathcal{S}$ be a simulator w.r.t. Kyber.PKE which outputs a uniformly random value from the ciphertext space $\mathcal{C}$ of Kyber.PKE. (Note that $\mathcal{C}$ is a set of bit strings with a *fixed pre-specified* length [11, Section 1.2], and hence, is *efficiently samplable*.) The above observation of Kyber.PKE's public-keys and ciphertexts being pseudorandom under the MLWE assumption can be used in a straightforward manner to show that Kyber.PKE is tightly SDS-IND secure w.r.t. $\mathcal{S}$ (cf. Definition 24) under the MLWE hardness assumption – as also noted in [11, Section 4.3.2].

Coming to the statistical disjointness of Kyber.PKE w.r.t. $\mathcal{S}$ (cf. Definition 24), let $\mathcal{M}$ and $\mathcal{R}$ additionally denote the message space and encryption randomness space of Kyber.PKE respectively; also let $\mathsf{Enc}(\mathsf{pk}, \mathcal{M})$ denote the set of valid ciphertexts c of Kyber.PKE where there exists $m \in \mathcal{M}$ and $r \in \mathcal{R}$ such that $c = \mathsf{Enc}(\mathsf{pk}, m; r)$. Now it is not hard to see that we have $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}} \le \frac{|\mathsf{Enc}(\mathsf{pk},\mathcal{M})|}{|\mathcal{C}|} \le \frac{|\mathcal{M}||\mathcal{R}|}{|\mathcal{C}|}$. Note that across all parameter sets of

[13] Fortunately, this is the case for the base Kyber.PKE scheme, as can be seen in [11].

Kyber [11, Section 1], we have $|\mathcal{C}| \geq 2^{6144}$ and $|\mathcal{M} \times \mathcal{R}| = 2^{512}$. Hence, for all intents and purposes, $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}$ can be considered to be negligible. □

5.3.3 *Anonymity and Collision-Freeness of* Kyber.KEM

Towards establishing ANO-CCA security of Kyber.KEM, we first prove its concrete SPR-CCA security in the QROM while relying on the *strong disjoint simulatability* (i.e., *SDS-IND security* and *statistical disjointness*; cf. Lemma 13) of the base Kyber.PKE scheme.

Theorem 16. *Given the base PKE scheme* Kyber.PKE $= (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, let* $\mathcal{S}$ *be a simulator w.r.t.* Kyber.PKE *which outputs a uniformly random value from the ciphertext space of* Kyber.PKE*. Then for any* SPR-CCA *adversary* $\mathcal{A}$ *against* Kyber.KEM $= (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_D *classical queries to the decapsulation oracles, and at most* q_G*,* q_H *and* $q_{H'}$ *queries to the quantum random oracles* G*,* H *and* H' *respectively, there exists an* IND-CPA *adversary* $\mathcal{B}$ *and a* SDS-IND *adversary* $\mathcal{B}'$ *against* Kyber.PKE *such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\mathcal{A}) \leq\ & q_G\sqrt{\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{1}{2}\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}} \\ & + \mathbf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathsf{Kyber.PKE},\mathcal{S}}(\mathcal{B}') + (2 + 8(q_G + q_D + 2)^2 + 8(2q_G + 2)^2)\delta \\ & + \frac{2(q_{H'} + q_D)}{2^{128}} + \frac{C(q_H + 1)^3}{2^{256}} + \frac{q_H + 7q_{H'}}{2^{128}},\end{aligned}$$

where C (< 648) *is the constant from Lemma 6, and the running times of* $\mathcal{B}$ *and* $\mathcal{B}'$ *is about the same as that of* $\mathcal{A}$*.*

The proof follows quite closely to that of IND-CCA security of Kyber.KEM in the QROM above (Theorem 2). We will focus on the main differences in our SPR-CCA security analysis below.

Proof. Same as in our proof of IND-CCA security for Kyber.KEM (Theorem 2), we first consider SPR-CCA security of the "intermediate" scheme $\overline{\mathsf{Kyber}}.\mathsf{KEM} = \mathsf{FO}^{\mathsf{Kyber}}_{\mathsf{pre}}[\mathsf{Kyber.PKE}, G, H, H'] = (\overline{\mathsf{KGen}}', \overline{\mathsf{Encap}}, \overline{\mathsf{Decap}})$ (see Fig. 3.7) in the QROM.

Denote $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$, $\Omega_{\mathbf{H}'}$, $\Omega_{\mathbf{H}''}$ and $\Omega_{\overline{\mathbf{H}}}$ to be the set of all functions $\mathbf{G} : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{H} : \{0,1\}^{256} \cup \mathcal{PK} \cup \mathcal{C} \to \{0,1\}^{256}$, $\mathbf{H}' : \{0,1\}^{256} \times (\{0,1\}^{256} \cup \mathcal{C}) \to \{0,1\}^{256}$, $\mathbf{H}'' : \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ and $\overline{\mathbf{H}} : \{0,1\}^{256} \to \{0,1\}^{256}$ respectively, where $\mathcal{PK}$ is the space of all Kyber.PKE public keys and $\mathcal{C}$ is the ciphertext space of Kyber.PKE.

```
Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$
$G \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H' \leftarrow\!\!\$\ \Omega_{\mathbf{H'}}$
$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H''}}; \overline{H} \leftarrow\!\!\$\ \Omega_{\overline{\mathbf{H}}}$
$(\mathsf{pk}, \mathsf{sk}') \leftarrow \overline{\mathsf{KGen}}'$
$(c_0^*, \overline{k}_0^*) \leftarrow \overline{\mathsf{Encap}}(\mathsf{pk})$
$c_1^* \leftarrow \mathcal{S}()$
$\overline{k}_1^* \leftarrow\!\!\$\ \{0,1\}^{256}$
$b \leftarrow\!\!\$\ \{0,1\}$
$b' \leftarrow \overline{\mathcal{A}}^{G,H,H',\overline{\text{DECAPS}}_{c^*}}(\mathsf{pk}, c_b^*, \overline{k}_b^*)$
return $[b' = b]$
```

```
$\overline{\text{DECAPS}}_a(c)$   // $c \neq a$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
    return $\overline{k}'$
else return $H'(s, c)$   // $\overline{\mathsf{G}}_0$
else return $H''(c)$   // $\overline{\mathsf{G}}_1$
else return $\overline{H}(H(c))$   // $\overline{\mathsf{G}}_2$
```

FIGURE 5.10: Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$ for the proof of Theorem 16. Here $H''\colon \{0,1\}^{256} \cup \mathcal{C} \to \{0,1\}^{256}$ and $\overline{H}\colon \{0,1\}^{256} \to \{0,1\}^{256}$ are fresh *internal* random oracles, i.e., not directly accessible to $\overline{\mathcal{A}}$.

Let $\overline{\mathcal{A}}$ be an SPR-CCA adversary against $\overline{\mathsf{Kyber.KEM}}$ w.r.t. simulator $\mathcal{S}$ (described above) issuing at most q'_D classical queries to the decapsulation oracles, and q'_H and $q'_{H'}$ quantum queries to the random oracles H and H' respectively. Consider the sequence of games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$ described in Figure 5.10. It is straightforward to obtain the following based on our IND-CCA security analysis of $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ (Inequality (3.1)) in the proof of Theorem 2 above.

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right| \leq \mathbf{Adv}^{\text{SPR-CCA}}_{\overline{\mathsf{Kyber.KEM}},\mathcal{S}}(\overline{\mathcal{A}}) + \frac{2q'_{H'}}{2^{128}} + \frac{C(q'_H + q'_D + 1)^3}{2^{256}}. \quad (5.1)$$

Now we return to proving SPR-CCA security of the *actual* Kyber.KEM. Let $\mathcal{A}$ be an SPR-CCA adversary against Kyber.KEM w.r.t. $\mathcal{S}$ issuing at most q_D classical queries to the decapsulation oracles, and at most q_G, q_H and $q_{H'}$ quantum queries to the random oracles G, H and H' respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_7$ described in Fig. 5.11. These games are quite similar to the ones described in Fig. 3.9 in our IND-CCA security proof.

Game G_0**:** This game is the SPR-CCA game for Kyber.KEM with the "real" ciphertext c^* and "real" encapsulated key k^* where $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$.

Now note that the games $\mathsf{G}_0 - \mathsf{G}_3$ in Figure 5.11 are essentially *identical* to the games "$\mathsf{G}_0 - \mathsf{G}_3$" defined in Figure 3.9. Hence, from our analysis of

```
Games $\mathsf{G}_0$ – $\mathsf{G}_7$
$G \leftarrow\!\!\$\; \Omega_{\mathbf{G}}; H \leftarrow\!\!\$\; \Omega_{\mathbf{H}}; H' \leftarrow\!\!\$\; \Omega_{\mathbf{H}'}$
$H'' \leftarrow\!\!\$\; \Omega_{\mathbf{H}''}; \overline{H} \leftarrow\!\!\$\; \Omega_{\overline{\mathbf{H}}};$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'; h \leftarrow H(\mathsf{pk})$
$m^* \leftarrow\!\!\$\; \{0,1\}^{256}$
$m^* \leftarrow H(m^*)$   // $\mathsf{G}_0$
$(\overline{k}_0^*, r^*) \leftarrow G(m^*, h); \overline{k}_1^* \leftarrow\!\!\$\; \{0,1\}^{256}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$   // $\mathsf{G}_0$ – $\mathsf{G}_3$
$c^* \leftarrow \mathcal{S}()$   // $\mathsf{G}_4$ – $\mathsf{G}_7$
$k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$   // $\mathsf{G}_0$ – $\mathsf{G}_3$
$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$   // $\mathsf{G}_4$
$k^* \leftarrow\!\!\$\; \{0,1\}^{256}$   // $\mathsf{G}_5$ – $\mathsf{G}_7$
$b' \leftarrow \mathcal{A}^{G,H,H',\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k^*)$
return $b'$
```

```
$\textsc{Decaps}_a(c)$   // $c \neq a$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
  return $H'(\overline{k}', H(c))$
else
  return $H'(s, H(c))$   // $\mathsf{G}_0$ – $\mathsf{G}_1$, $\mathsf{G}_7$
  return $H''(H(c))$   // $\mathsf{G}_2, \mathsf{G}_6$
  return $H'(\overline{H}(H(c)), H(c))$   // $\mathsf{G}_3$–$\mathsf{G}_5$
```

FIGURE 5.11: Games G_0 – G_7 for the proof of Theorem 16.

these game hops in the above IND-CCA security proof, it is not hard to obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \frac{2q_H}{2^{128}} + \frac{4q_{H'}}{2^{128}}.$$

Game G_4**:** Relative to G_3 (and G_0), we modify how the challenge ciphertext c^* and corresponding encapsulated key k^* are generated. In this game, we generate (c^*, k^*) as $c^* \leftarrow \mathcal{S}()$ and $k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$ instead, where $\mathcal{S}$ is the simulator described above and $\overline{k}_1^* \leftarrow\!\!\$\; \{0,1\}^{256}$. Here we use our SPR-CCA security analysis of the intermediate $\overline{\mathsf{Kyber}}.\mathsf{KEM}$.

To be specific, recall that in the corresponding "$\mathsf{G}_3 \to \mathsf{G}_4$" hop in our above IND-CCA security proof of Kyber.KEM, we showed a reduction to IND-CCA security of the underlying $\overline{\mathsf{Kyber}}.\mathsf{KEM}$. In a similar way, it is straightforward to construct an SPR-CCA adversary $\overline{\mathcal{A}}$ against $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ w.r.t. the same $\mathcal{S}$ above such that

$$\begin{aligned}|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_4 = 1]| &= 2 \cdot \left| \Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2} \right| \\ &\leq 2\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}}) + \frac{4q_{H'}}{2^{128}} + \frac{2C(q_H + 1)^3}{2^{256}},\end{aligned}$$

where we used Inequality (5.1) w.r.t. our analysis of $\overline{\mathsf{Kyber}}.\mathsf{KEM}$.

Game G_5**:** We further modify how k^* is generated. In this game, k^* is chosen from $\{0,1\}^{256}$ uniformly at random. Similar to our analysis of the "$\mathsf{G}_4 \to \mathsf{G}_5$" hop in the proof of Theorem 2, we obtain the following by applying Lemma 2,

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_6**:** We modify the decapsulation oracle such that the oracle rejects an invalid ciphertext c by returning $H''(H(c))$. In a sense, we are reverting the changes introduced in the "$\mathsf{G}_2 \to \mathsf{G}_3$" hop above in the proof of Theorem 2. Hence, it is not hard to obtain

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Game G_7**:** We again modify the decapsulation oracle such that the oracle returns $H'(s, H(c))$ for an invalid ciphertext c. From our analysis of the "$\mathsf{G}_1 \to \mathsf{G}_2$" hop above in the proof of Theorem 2, we have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

Note that G_7 is the SPR-CCA game for $\mathsf{Kyber.KEM}$ where $\mathcal{A}$ gets a "random" ciphertext $c^* \leftarrow \mathcal{S}()$ and "random" encapsulated key $k^* \leftarrow\$ \{0,1\}^{256}$. Hence, by summing up the above bounds, we obtain

$$\begin{aligned}2\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\mathcal{A}) &= |\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_7 = 1]| \\ &\leq 2\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}}) + \frac{2C(q_H+1)^3}{2^{256}} + \frac{2q_H + 14q_{H'}}{2^{128}}.\end{aligned}$$

Finally, we replace the term "$\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}})$" with the existing SPR-CCA security bounds on the $\mathsf{FO}^{\not\perp}_m$ transform in the QROM derived in [60]. Because as previously noted in our proof of Theorem 2 above, the intermediate $\mathsf{FO}^{\not\perp'}_{\mathsf{pre}}$ transform is essentially identical to $\mathsf{FO}^{\not\perp}_m$ in the context of "single key-pair notions" such as IND-CCA security *and* SPR-CCA security. Hence, by applying Theorem 15 w.r.t. SPR-CCA security of "$\mathsf{FO}^{\not\perp}_m$-derived" KEMs in the QROM to $\overline{\mathsf{Kyber}}.\mathsf{KEM}$, we have that there exists an IND-CPA adversary

$\mathcal{B}$ and a SDS-IND adversary $\mathcal{B}'$ against Kyber.PKE w.r.t. $\mathcal{S}$, running in about the same time as that of $\overline{\mathcal{A}}$ (and $\mathcal{A}$), such that[14]

$$\mathbf{Adv}^{\mathsf{SPR\text{-}CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\overline{\mathcal{A}}) \le q_G\sqrt{\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{1}{2}\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}$$

$$+ \mathbf{Adv}^{\mathsf{SDS\text{-}IND}}_{\mathsf{Kyber.PKE},\mathcal{S}}(\mathcal{B}') + \frac{2(q_{H'} + q_D)}{2^{128}} + (2 + 8(q_G + q_D + 2)^2 + 8(2q_G + 2)^2)\delta.$$

Combining the last two inequalities finishes the proof.

□

Note that our above SPR-CCA security analysis of Kyber.KEM relies on δ-correctness of the base Kyber.PKE scheme. However as mentioned in Subsection 3.2.3 related to our IND-CCA security analysis of Kyber, this particular δ-correctness property of Kyber.PKE has been rigorously analyzed in [11, 61].

Now following Theorem 13 which states that the SPR-CCA security of a KEM implies its ANO-CCA security, and following the IND-CPA security ([11, Theorem 1]) and strong disjoint simulatability (Lemma 13) of Kyber.PKE under the MLWE hardness assumption, we have:

Corollary 1 (informal)**.** *Kyber.KEM is ANO-CCA secure in the QROM, under the MLWE hardness assumption.*

Remark 4. Note that our above SPR-CCA/ANO-CCA security analysis of Kyber.KEM in the QROM is non-tight. This is due to the existing non-tight SPR-CCA security proof for $\mathsf{FO}^{\not\perp}_m$-derived KEMs in [60] (i.e., Theorem 15). The author of [60] also discussed difficulties in using alternative proof techniques (i.e., tighter OW2H lemmas in [51, 52]) to obtain a tighter proof of SPR-CCA security w.r.t. the $\mathsf{FO}^{\not\perp}_m$ transform in the QROM. They also left the task of overcoming these difficulties as an open problem; we do the same as well, in the context of obtaining tighter proofs of post-quantum ANO-CCA security for Kyber.

Coming to SCFR-CCA security of Kyber.KEM, we can apply similar proof strategies that were used to establish strong collision-freeness of $\mathsf{FO}^{\not\perp}$-based KEMs (Theorem 8) and FrodoKEM (Theorem 11) in the QROM. The

[14] Technically, Theorem 15 includes statistical disjointness (cf. Definition 24) of a *derandomized* version of the base PKE scheme in its SPR-CCA security bounds on the final KEM. Roughly speaking, in such a derandomized PKE, the random coins used to encrypt a message m is obtained by first hashing m. But from our proof sketch of Lemma 13, it is not hard to see that statistical disjointness of the derandomized Kyber.PKE is trivially upper-bounded by disjointness of the *original* Kyber.PKE, i.e., $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}$. This is because our simulator $\mathcal{S}$ just outputs a uniformly random Kyber.PKE ciphertext.

corresponding proof for Kyber, on a high level, uses the fact that the hash of public-keys are included in the KEM's key-derivation step (in contrast to Classic McEliece). This allows us to establish SCFR-CCA security of Kyber by mainly relying on properties of quantum random oracles G and H': namely, claw-freeness and collision-resistance.

Theorem 17. *For any* SCFR-CCA *adversary $\mathcal{A}$ against the scheme* $\mathsf{Kyber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_G and $q_{H'}$ queries to the quantum random oracle G and H' respectively, we have*

$$\mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \leq \frac{C(q_G+1)^3}{2^{256}} + \frac{4C(q_{H'}+1)^3}{2^{256}} + \frac{1}{2^{256}} + \frac{4q_{H'}}{2^{128}}.$$

where C (< 648) is the constant from Lemma 11.

Proof. Denote $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$, $\Omega_{\mathbf{H}'}$ and $\Omega_{\overline{\mathbf{H}}}$ to be the set of all functions $\mathbf{G} : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{H} : \mathcal{PK} \cup \mathcal{C} \to \{0,1\}^{256}$, $\mathbf{H}' : \{0,1\}^{512} \to \{0,1\}^{256}$ and $\overline{\mathbf{H}} : \{0,1\}^{256} \to \{0,1\}^{256}$ respectively, where $\mathcal{PK}$ is the space of all $\mathsf{Kyber.PKE}$ public keys and $\mathcal{C}$ is the ciphertext space of $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$.

Let $\mathcal{A}$ be an adversary in the SCFR-CCA game for $\mathsf{Kyber.KEM}$ issuing at most q_G and $q_{H'}$ quantum queries to the random oracle G and H' respectively.

The structure of the proof is very similar to that of Theorem 10. Namely, we do the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_3$ as described in Figure 5.12. Since this sequence is similar to the game-hops "$\mathsf{G}_0 \to \mathsf{G}_3$" in the proof of Theorem 10, by a similar analysis we obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Coll}^{H}_{\mathsf{Kyber.PKE}} + \frac{4q_{H'}}{2^{128}},$$

where $\mathbf{Coll}^{H}_{\mathsf{Kyber.PKE}}$ is probability of the event "$H(\mathsf{pk}_0) = H(\mathsf{pk}_1)$" with pk_0 and pk_1 being two honestly-generated public keys of $\mathsf{Kyber.PKE}$. Now since H is modelled as a random oracle (in contrast to our analysis of FrodoKEM, see Footnote 7 of this chapter), it is easy to see that $\mathbf{Coll}^{H}_{\mathsf{Kyber.PKE}} \leq \frac{1}{2^{256}}$.

Note that the game G_0 is exactly the SCFR-CCA game for $\mathsf{Kyber.KEM}$. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}).$$

Coming to the game G_3, note that the adversary $\mathcal{A}$ wins the game if it outputs a ciphertext c such that $\textsc{Decaps}_\perp(0,c) = \textsc{Decaps}_\perp(1,c)$. Let $m'_0 = \mathsf{Dec}(\mathsf{sk}_0, c)$, $m'_1 = \mathsf{Dec}(\mathsf{sk}_1, c)$, $\overline{k_0}' \leftarrow G_{0k}(m'_0)$ and $\overline{k_1}' \leftarrow G_{1k}(m'_1)$. There are four disjoint cases that need to be considered w.r.t. this winning condition:

```
Games $\mathsf{G}_0 - \mathsf{G}_3$
$H \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H' \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
$G_2 \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; G_{0r}, G_{1r} \leftarrow\!\!\$\ \Omega_{\overline{\mathbf{H}}}$
$G_{0k}, G_{1k} \leftarrow\!\!\$\ \Omega_{\overline{\mathbf{H}}}$
$H_0^{\mathsf{rej}}, H_1^{\mathsf{rej}} \leftarrow\!\!\$\ \Omega_{\overline{\mathbf{H}}}$
$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'$
$\mathsf{inp} \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1)$
$c \leftarrow \mathcal{A}^{G, H', \textsc{Decaps}_\perp}(\mathsf{inp})$
$k_0 = \textsc{Decaps}_\perp(0, c)$
$k_1 = \textsc{Decaps}_\perp(1, c)$
return $[k_0 = k_1 \neq \perp]$
```

```
$\textsc{Decaps}_a(0, c)$    // $c \neq a$
Parse $\mathsf{sk}'_0 = (\mathsf{sk}_0, \mathsf{pk}_0, h_0, s_0)$
$m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
$(\overline{k}', r') \leftarrow G(m', h_0)$    // $\mathsf{G}_0 - \mathsf{G}_2$
$r' \leftarrow G_{0r}(m')$    // $\mathsf{G}_3$
$\overline{k}' \leftarrow G_{0k}(m')$    // $\mathsf{G}_3$
if $\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c$ then
    return $H'(\overline{k}', H(c))$
else return $H'(s_0, H(c))$    // $\mathsf{G}_0$
else return $H_0^{\mathsf{rej}}(H(c))$    // $\mathsf{G}_1$ - $\mathsf{G}_3$
```

```
$G(m, h)$
if $h = H(\mathsf{pk}_0)$ then    // $\mathsf{G}_2$-$\mathsf{G}_3$
    $r \leftarrow G_{0r}(m)$    // $\mathsf{G}_2$ - $\mathsf{G}_3$
    $\overline{k} \leftarrow G_{0k}(m)$    // $\mathsf{G}_2$ - $\mathsf{G}_3$
elseif $h = H(\mathsf{pk}_1)$ then    // $\mathsf{G}_2$-$\mathsf{G}_3$
    $r \leftarrow G_{1r}(m)$    // $\mathsf{G}_2$ - $\mathsf{G}_3$
    $\overline{k} \leftarrow G_{1k}(m)$    // $\mathsf{G}_2$ - $\mathsf{G}_3$
else $(\overline{k}, r) \leftarrow G_2(m, h)$
return $(\overline{k}, r)$
```

```
$\textsc{Decaps}_a(1, c)$    // $c \neq a$
Parse $\mathsf{sk}'_1 = (\mathsf{sk}_1, \mathsf{pk}_1, h_1, s_1)$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
$(\overline{k}', r') \leftarrow G(m', h_1)$    // $\mathsf{G}_0 - \mathsf{G}_2$
$r' \leftarrow G_{1r}(m')$    // $\mathsf{G}_3$
$\overline{k}' \leftarrow G_{1k}(m')$    // $\mathsf{G}_3$
if $\mathsf{Enc}(\mathsf{pk}_1, m'; r') = c$ then
    return $H'(\overline{k}', H(c))$
else return $H'(s_1, H(c))$    // $\mathsf{G}_0$
else return $H_1^{\mathsf{rej}}(H(c))$    // $\mathsf{G}_1$ - $\mathsf{G}_3$
```

FIGURE 5.12: Games G_0 – G_3 for the proof of Theorem 17. Here Enc and Dec are the encryption and decryption algorithms of Kyber.PKE.

- $\underline{\textsc{Decaps}_\perp(0,c) = H'(\overline{k_0}', H(c)) \wedge \textsc{Decaps}_\perp(1,c) = H'(\overline{k_1}', H(c))}$:
  - $\underline{\overline{k_0}' \neq \overline{k_1}'}$: Winning condition in this case translates to $H'(\overline{k_0}', H(c)) = H'(\overline{k_1}', H(c))$, where $\overline{k_0}' \neq \overline{k_1}'$. This implies a collision in the QRO H'. Hence using Lemma 6, we can bound the probability of this sub-event by $\frac{C(q_{H'}+1)^3}{2^{256}}$ via a straightforward reduction to the collision-resistance of H'.
  - $\underline{\overline{k_0}' = \overline{k_1}'}$: In this sub-case, note that (m_0', m_1') is a *claw* w.r.t. the pair of quantum random oracles G_{0k} and G_{1k}. Using Lemma 11, we can bound the probability of this event by $\frac{C(q_G+1)^3}{2^{256}}$ via a straightforward reduction to the claw-finding problem w.r.t. the instance (G_{0k}, G_{1k}).

- $\underline{\textsc{Decaps}_\perp(0,c) = H'(\overline{k_0}', H(c)) \wedge \textsc{Decaps}_\perp(1,c) = H_1^{\mathrm{rej}}(H(c))}$: In this case, the winning condition translates to $H'(\overline{k_0}', H(c)) = H_1^{\mathrm{rej}}(H(c))$. Note that then $((\overline{k_0}', H(c)), H(c))$ is a claw w.r.t. the pair of QROs H' and H_1^{rej}. Using Lemma 11, we can bound the probability of this event by $\frac{C(q_{H'}+1)^3}{2^{256}}$ via a straightforward reduction to the claw-finding problem w.r.t. the instance (H', H_1^{rej}).

- $\underline{\textsc{Decaps}_\perp(0,c) = H_0^{\mathrm{rej}}(H(c)) \wedge \textsc{Decaps}_\perp(1,c) = H'(\overline{k_1}', H(c))}$: The analysis here will be the same as the previous case.

- $\underline{\textsc{Decaps}_\perp(0,c) = H_0^{\mathrm{rej}}(H(c)) \wedge \textsc{Decaps}_\perp(1,c) = H_1^{\mathrm{rej}}(H(c))}$: In this case, the winning condition translates to $H_0^{\mathrm{rej}}(H(c)) = H_1^{\mathrm{rej}}(H(c))$. Note that $(H(c), H(c))$ is then a claw w.r.t. the pair of random oracles H_0^{rej} and H_1^{rej}. Using Lemma 11, we can bound the probability of this event by $\frac{C(q_{H'}+1)^3}{2^{256}}$ via a straightforward reduction to the claw-finding problem w.r.t. the instance $(H_0^{\mathrm{rej}}, H_1^{\mathrm{rej}})$.

From the above analysis, we have

$$\Pr[\mathsf{G}_3 = 1] \leq \frac{C(q_G+1)^3}{2^{256}} + \frac{4C(q_{H'}+1)^3}{2^{256}}.$$

Hence, we finally get

$$\mathbf{Adv}^{\mathsf{SCFR\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \leq \frac{C(q_G+1)^3}{2^{256}} + \frac{4C(q_{H'}+1)^3}{2^{256}} + \frac{1}{2^{256}} + \frac{4q_{H'}}{2^{128}}.$$

□

5.3.4 *Anonymity and Robustness of Hybrid PKE Derived from* Kyber.KEM

We first focus on anonymity, or more specifically, SPR-CCA security of hybrid PKE schemes obtained from Kyber.KEM via the KEM-DEM paradigm. From Theorem 14, we have that composing a *one-time strongly pseudorandom* (or, *SPR-otCCA secure*; see Definition 26) DEM with an implicitly-rejecting KEM which is both SPR-CCA secure *and strongly smooth* (or, *SSMT-CCA secure*; see Definition 25) results in an SPR-CCA secure hybrid PKE scheme. Since we already established SPR-CCA security of Kyber.KEM in Subsection 5.3.3 above, we now prove its concrete SSMT-CCA security in the QROM while relying on statistical disjointness of the base Kyber.PKE.

Theorem 18. *Let $\mathcal{S}$ be a simulator w.r.t.* $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *which outputs a uniformly random value from the ciphertext space of* $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$. *For any* SSMT-CCA *adversary $\mathcal{A}$ against the scheme* $\mathsf{Kyber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *w.r.t. $\mathcal{S}$ issuing at most q_H and $q_{H'}$ queries to the quantum random oracles H and H' respectively, we have*

$$\mathbf{Adv}^{\mathsf{SSMT\text{-}CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\mathcal{A}) \leq \mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}} + \frac{2q_{H'}+1}{2^{128}} + \frac{C(q_H+1)^3}{2 \cdot 2^{256}},$$

where C (< 648) is the constant from Lemma 6.

Proof. Denote $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$, $\Omega_{\mathbf{H}'}$ and $\Omega_{\mathbf{H}''}$ to be the set of all functions $\mathbf{G} : \{0,1\}^{512} \to \{0,1\}^{512}$, $\mathbf{H} : \mathcal{PK} \cup \mathcal{C} \to \{0,1\}^{256}$, $\mathbf{H}' : \{0,1\}^{512} \to \{0,1\}^{256}$ and $\mathbf{H}'' : \{0,1\}^{256} \to \{0,1\}^{256}$ respectively, where $\mathcal{PK}$ is the space of all Kyber.PKE public keys and $\mathcal{C}$ is the ciphertext space of Kyber.PKE.

Let $\mathcal{A}$ be an adversary in the SSMT-CCA game for Kyber.KEM issuing at most q_H and $q_{H'}$ quantum queries to the random oracles H and H' respectively. Consider the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_6$ described in Figure 5.13.

Game G_0**:** This game is equivalent to the SSMT-CCA game for Kyber.KEM with the random encapsulated key $k^* \leftarrow\$ \{0,1\}^{256}$ and simulated ciphertext $c^* \leftarrow \mathcal{S}()$.

```
Games $\mathsf{G}_0 - \mathsf{G}_6$
$G \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H \leftarrow\!\!\$\ \Omega_{\mathbf{H}}$
$H' \leftarrow\!\!\$\ \Omega_{\mathbf{H'}}; H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H''}}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
$c^* \leftarrow \mathcal{S}()$    // $\mathsf{G}_0, \mathsf{G}_6$
$c^* \leftarrow \mathcal{S}() \setminus \mathsf{Enc}(\mathsf{pk}, \mathcal{M})$    // $\mathsf{G}_1 - \mathsf{G}_5$
$k^* \leftarrow\!\!\$\ \{0,1\}^{256}$    // $\mathsf{G}_0 - \mathsf{G}_2$
$k^* \leftarrow H''(H(c^*))$    // $\mathsf{G}_3$
$k^* \leftarrow H'(s, H(c^*))$    // $\mathsf{G}_4$
$k^* := \mathsf{Decap}(\mathsf{sk}', c^*)$    // $\mathsf{G}_5 - \mathsf{G}_6$
$b' \leftarrow \mathcal{A}^{G,H,H',\textsc{Decaps}_{c^*}}(\mathsf{pk}, c^*, k^*)$
return $b'$
```

```
$\textsc{Decaps}_a(c)$    // $c \neq a$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
if $c = c^*$ then return $\perp$
$m' := \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
    return $H'(\overline{k}', H(c))$
else
    return $H'(s, H(c))$    // $\mathsf{G}_0$–$\mathsf{G}_1$, $\mathsf{G}_4$–$\mathsf{G}_6$
    return $H''(H(c))$    // $\mathsf{G}_2 - \mathsf{G}_3$
```

FIGURE 5.13: Games $\mathsf{G}_0 - \mathsf{G}_6$ for the proof of Theorem 18.

Game G_1: We then modify how c^* is generated. In this game, c^* is generated by $\mathcal{S}()$ conditioned on that c^* is outside of the set $\mathsf{Enc}(\mathsf{pk}, \mathcal{M})$[15]. More specifically, the game does a (potentially inefficient) check on whether $c^* \in \mathsf{Enc}(\mathsf{pk}, \mathcal{M})$ and aborts if it is the case. Note that this potential inefficiency does not really matter in our analysis since we will bound the difference between subsequent games using *statistical* bounds anyway.

Coming to the difference between games G_0 and G_1, it is bounded by the value $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}$, and we have

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}.$$

Game G_2: We next modify the "implicit rejection" of the decapsulation oracle. In this game, the oracle rejects by outputting $H''(H(c))$ instead of $H'(s, H(c))$, where H'' is an internal and independent random oracle not directly accessible to $\mathcal{A}$. From the "$\mathsf{G}_1 \rightarrow \mathsf{G}_2$" hop in the proof of Theorem 2 above, we obtain the following via Lemma 2:

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_{H'}}{2^{128}}.$$

[15] Recall that $\mathsf{Enc}(\mathsf{pk}, \mathcal{M})$ denotes the set of valid ciphertexts c of $\mathsf{Kyber.PKE}$ where there exists $m \in \mathcal{M}$ and $r \in \mathcal{R}$ such that $c = \mathsf{Enc}(\mathsf{pk}, m; r)$; here $\mathcal{M}$ and $\mathcal{R}$ denote the message space and encryption randomness space of $\mathsf{Kyber.PKE}$ respectively.

Game G_3**:** We next modify how k^* is generated. In this game, k^* is computed as $H''(H(c^*))$ instead of being chosen uniformly at random.

Notice that the adversary can only access H'' via the decapsulation oracle. Thus, if the adversary cannot query $c \neq c^*$ such that $H(c) = H(c^*)$, then the adversary cannot obtain any information on $H''(H(c^*))$ and this value looks completely random. Similar to the "$\overline{\mathsf{G}}_1 \to \overline{\mathsf{G}}_2$" hop above in our IND-CCA security proof of Kyber.KEM, we can bound the difference between G_2 and G_3 via a straightforward reduction to the collision resistance of H. Hence, we have from Lemma 6

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \frac{C(q_H + 1)^3}{2^{256}}.$$

Game G_4**:** We next replace all invocations of $H''(H(\cdot))$ in this game – particularly, during generation of k^* and decapsulation of ciphertexts – with $H'(s, H(\cdot))$. Again from the "$\mathsf{G}_1 \to \mathsf{G}_2$" hop above, we can use the pseudorandomness of H' (Lemma 2) to obtain

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2(q_{H'} + 1)}{2^{128}}.$$

Game G_5**:** In this game, we compute k^* as $k^* := \mathsf{Decap}(\mathsf{sk}', c^*)$ instead of $k^* \leftarrow H'(s, H(c^*))$. Anyways the result of $\mathsf{Decap}(\mathsf{sk}', c^*)$ in G_5 will be equal to $H'(s, H(c^*))$ as in G_4. Because note that c^* is an invalid ciphertext since it is outside of $\mathsf{Enc}(\mathsf{pk}, \mathcal{M})$. Thus, even if the decryption of c^* yields some plaintext m', the re-encrypted ciphertext $c' = \mathsf{Enc}(\mathsf{pk}, m'; r')$ cannot be equivalent to c^*. Hence, we have

$$\Pr[\mathsf{G}_4 = 1] = \Pr[\mathsf{G}_5 = 1].$$

Game G_6**:** We finally modify how c^* is generated. In this game, c^* is generated by $\mathcal{S}()$ (and there is no check by the game on whether $c^* \in \mathsf{Enc}(\mathsf{pk}, \mathcal{M})$). We note that this game is the SSMT-CCA game for Kyber.KEM with simulated ciphertext $c^* \leftarrow \mathcal{S}()$ and decapsulated key $k^* := \mathsf{Decap}(\mathsf{sk}, c^*)$.

The difference is again bounded by $\mathsf{Disj}_{\mathsf{Kyber.PKE}, \mathcal{S}}$, and we have

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \mathsf{Disj}_{\mathsf{Kyber.PKE}, \mathcal{S}}.$$

Summing up the above differences, we have

$$\begin{aligned} 2\mathbf{Adv}^{\mathsf{SSMT\text{-}CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) &= |\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_6 = 1]| \\ &\leq 2\mathsf{Disj}_{\mathsf{Kyber.PKE}, \mathcal{S}} + \frac{4q_{H'} + 2}{2^{128}} + \frac{C(q_H + 1)^3}{2^{256}}. \end{aligned}$$

□

Coming to robustness, it follows from Theorems 4 and 17 that composing $\mathsf{Kyber.KEM}$ with an FROB secure DEM (see Definition 19) will result in an SROB-CCA secure hybrid PKE scheme. In other words, composing Kyber with a one-time strongly pseudorandom and robust DEM will result in a post-quantum strongly anonymous *and* strongly robust PKE scheme.

Remark 5. As mentioned in Subsection 3.2.3 (Remark 2), *Saber* [62] – a NIST PQC third-round finalist – uses the $\mathsf{FO}^{\mathsf{kyber}}$ transform in its KEM construction as well. Therefore, our above results on anonymity and robustness of Kyber in the post-quantum setting also apply to Saber in a similar manner.

5.4 SUMMARY

In this chapter, we applied our generic results of Chapter 4 on anonymity and robustness of implicit rejection KEMs, and corresponding hybrid PKE schemes, to three NIST PQC KEMs: Classic McEliece, FrodoKEM and Kyber. For Classic McEliece, we highlighted a surprising property which shows that the KEM does not lead to robust PKE schemes via the standard KEM-DEM paradigm. However, since Xagawa [60] was able to establish anonymity of hybrid PKE schemes derived from Classic McEliece, it would be interesting to find applications for such schemes which require anonymity but where a lack of robustness is not an issue.

On the positive side, we showed that FrodoKEM and Kyber can be used to build anonymous *and* robust hybrid PKE schemes in the post-quantum setting. For FrodoKEM, we had to adapt our generic analysis of $\mathsf{FO}^{\not\perp}$ in Chapter 4 to the specific FO-variant used by the KEM. Whereas for Kyber, we had to use newer techniques – i.e., Xagawa's strong pseudorandomness framework [60] – in our analysis. We hope that these results provide further confidence to practitioners in using the new NIST standard Kyber and the BSI-recommended FrodoKEM not only in general-purpose applications that need IND-CCA security but also in emerging modern applications that require anonymity and robustness.

6

FUNCTIONALITY ENHANCEMENTS: EFFICIENT THRESHOLD DECRYPTION

The early days of post-quantum cryptography – especially in the context of NIST's PQC standardization process – looked at how to build basic primitives such as simple public-key encryption or digital signatures. However, our existing (pre-quantum) public-key algorithms often provide more advanced functionalities than what is offered by basic public-key primitives. For example, one may have so-called *group signatures*, *identity-based encryption* or *threshold public-key cryptosystems*, just to name a few. Focusing on the last class of algorithms, in a threshold public-key cryptosystem, roughly speaking, the underlying secret key is split into two or more *key-shares* across different users. The sharing is done in such a way so that even if some number of shares (below a certain *threshold*) are compromised, no information is leaked about the original key. Hence, this threshold approach significantly improves the confidentiality of secret keys in practical cryptographic implementations. At the same time, an important advantage of this approach is that the secret-key shares can be separately used, where the "distributed" operation across multiple users with respective shares results in the correct output, as if the original secret key was used by an equivalent single-user cryptographic algorithm. In fact, NIST has initiated plans to standardize threshold schemes for (potentially quantum-resistant) cryptographic primitives [24]. In view of this, we consider threshold – or, distributed – decryption for IND-CCA secure *hybrid* public-key encryption in this chapter. Such threshold PKE schemes have various applications: for example, they are used in *e-voting systems* [86, 87], *blockchain systems* [88], and *side-channel resistant* implementations of cryptosystems [89].

Even in the context of pre-quantum cryptography, distributed decryption for hybrid systems is problematic for many schemes, since to maintain IND-CCA security one would need to apply a distributed decryption procedure to the symmetric DEM component, which is rather expensive for long messages. There is an additional problem when we consider the post-quantum setting. Namely, as mentioned in Chapter 3, most NIST PQC candidates for public-key encryption construct a hybrid scheme by first building a OW-/IND-CPA secure PKE scheme and then creating an IND-CCA hybrid scheme using the Fujisaki-Okamoto (FO) transform [1, 4] in conjunction with the standard KEM-DEM paradigm [15]. However, the problem with the FO design pattern is that the decryption procedure needs to perform a hash to obtain the random coins used for encryption for the "re-encryption check". And this can be a complicated process to perform in a threshold manner for the lattice-based schemes – e.g., *Kyber* [11], *FrodoKEM* [16], *Saber* [62] – especially if this involves sampling discrete Gaussians or other distributions which are not compatible with whichever underlying methodology one is using to perform the threshold decryption.

In prior work [90], a *non-hybrid* lattice-based encryption scheme is given with a corresponding secure distributed decryption protocol. But the encryption scheme is only IND-CPA secure. In [91], a generic procedure for obtaining an arbitrary threshold variant of any functionality is provided; however the construction makes use of *fully homomorphic encryption*, and hence, is not very practical. Coming to work that relates to NIST's PQC standardization process, a distributed decryption operation was given for the first-round candidate *LIMA* [92] in [93]; more specifically, the operation was provided for the base (non-hybrid) PKE scheme underlying LIMA. An outline to "thresholdize" the hybrid PKE schemes obtained from LIMA was also given in [93]. However, the instantiation would not preserve the CCA security guarantees of the hybrid construction. Also from a performance perspective, the problem with the distributed decryption of LIMA was that it is a scheme based on the FO transform. And as mentioned above, the secure evaluation of the hash function and re-encryption operation during decryption is costly in the distributed setting.

The main contribution of this chapter is a generic transform which supports distributed decryption for hybrid PKE schemes and provides IND-CCA security in a post-quantum setting (i.e., in the QROM). More specifically we use a generic *multi-party computation (MPC)* framework to perform the distributed decryption, and the decryption algorithm provided by our transform is efficient within this framework. Our transform deviates from the above "FO + KEM-DEM" paradigm, and instead can be seen as closely related to the so-called *Tag-KEM* framework [22]. The key take-away from our generic hybrid construction is that the DEM component can be any one-time IND-CPA secure symmetric encryption scheme, and the PKE scheme underlying the KEM component can be any *rigid*[1] deterministic OW-CPA secure scheme which is perfectly correct. We also discuss how to extend the security analysis of our transform, in a non-generic manner,

[1] Recall that a deterministic PKE scheme is said to be rigid if decryption of a ciphertext which is not the output of an encryption operation always returns $\perp$; see also Definition 4.

to PKE schemes that are not perfectly correct. Regarding the potential applicability of our transform to NIST PQC schemes, it is worth pointing out that the fourth-round candidate *Classic McEliece* [21] and the third-round finalist *NTRU* [23] use deterministic, rigid and perfectly correct base PKE schemes in their corresponding KEM constructions.

CHAPTER ORGANIZATION. Section 6.1 contains some MPC-related preliminaries that are relevant to this chapter. In Section 6.2, we discuss problems with prior frameworks in the literature (i.e., the KEM-DEM framework [15] and Tag-KEM framework [22]) in the context of obtaining post-quantum secure hybrid PKE schemes with efficient distributed decryption; we then provide a high-level overview of our new framework, simply called "Hybrid", which solves these problems. A detailed description of our Hybrid transform, and the accompanying formal security analysis in the QROM, is presented in Section 6.3.

6.1 MPC PRELIMINARIES

In the following, we describe some MPC-related concepts – with a particular focus on threshold cryptography – that are relevant to understand the results of this chapter. However, we consider a detailed discussion on MPC "as a whole" to be beyond the scope of this thesis, and instead refer the reader to standard textbooks on MPC (e.g., [46, 94, 95]).

6.1.1 *PKE With Distributed Decryption*

The goal of our work is to produce threshold public-key encryption for long messages; namely, we would want to share the decryption key among a set of entities so that a given subset needs to come together to decrypt. Given a set of n parties $\mathcal{P} = \{P_1, \ldots, P_n\}$, we consider so-called *access structures* A consisting of a monotonically increasing set of subsets of $2^{\mathcal{P}}$. A set S is said to be qualified if $S \in A$, and unqualified otherwise.

Now given a PKE scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, we say that the scheme admits a distributed decryption functionality for an access structure A, if there are two n-party protocols Π_{KGen} and Π_{Dec}: the protocol Π_{KGen} produces for each party some data sk_i which are *shares* of a valid secret key sk with respect to KGen, and the protocol Π_{Dec} on input an agreed ciphertext c from all parties in $S \in A$, and the value sk_i from all parties in S, will output the value $m = \mathsf{Dec}(\mathsf{sk}, c)$.

The distributed decryption protocols are said to be secure in the IND-ATK sense (ATK $\in \{\text{CPA}, \text{CCA}\}$) if an unqualified set of adversarial parties cannot, while interacting with a qualified set of parties, break the IND-ATK security of the underlying encryption scheme. This security definition can be made more formal by saying that the distributed decryption protocol should act like an ideal decryption functionality; see e.g., [96, 97] for a specific instantiation.

We shall assume an actively secure MPC protocol for the access structure A, and will then construct an algorithm which implements the algorithm Dec within the MPC protocol. Thus it automatically becomes a distributed protocol Π_{Dec} for the decryption functionality, and its security is inherited from the underlying MPC protocol. More concretely, our methodology uses a generic actively-secure-with-abort[2] MPC functionality defined via the so-called *Linear Secret Sharing (LSS)* over a finite field.

Now the challenging part is to develop an encryption scheme PKE – and the corresponding instantiation of Dec – so as to enable the underlying MPC system to provide an *efficient* distributed implementation. We will do exactly this in the subsequent sections. But it is also worth pointing out that because of our reliance on generic MPC techniques, our approach does not minimize the level of interaction needed between parties in the threshold decryption procedure. Hence, an open problem would be to develop a methodology, or a concrete scheme, which can utilize minimal amount of communication possible by potentially using MPC in a non-generic way.

6.1.2 *MPC Friendly Hash Function:* Rescue

Our generic construction of hybrid PKE schemes with efficient distributed decryption will make use of *MPC-friendly hash functions*, such as those in [98, 99]. These hash function constructions are primarily *sponge-based* (see [46, Section 8.8] for a detailed description of the sponge construction). In this chapter, we will consider one such construction called "Rescue" as introduced in [98].

At a very high level, Rescue maintains a state of $t = r + c$ finite field elements in $\mathbb{F}_q$, for a prime q. The initial state of the "sponge" is defined to be the vector of t zero elements. A message is first mapped into $n = d \cdot r$

[2] This means that inputs of the parties remain private throughout the execution of the protocol, and when a set of adversaries deviate from the protocol, honest parties will catch this with overwhelming probability and then abort the protocol. This should be compared to passively secure protocols which offer a much weaker guarantee that security is only preserved if all parties follow the precise protocol steps correctly.

elements in $\mathbb{F}_q$, i.e., $m_0, m_1, \ldots, m_{n-1}$. The elements are absorbed into the sponge in d absorption phases, where r elements are absorbed in each phase. At each phase, a permutation $f : \mathbb{F}_q^t \to \mathbb{F}_q^t$ is applied resulting in a state $s_0, \ldots, s_{t-1}$. At the end of absorption, the r values $s_c, \ldots, s_{t-1}$ are output from the state. This process can then be repeated, with more data absorbed and then squeezed out. Thus overall, we are defining a map $H : \mathbb{F}_q^n \to \mathbb{F}_q^r$.

A nice feature about Rescue is that the primitive calls f involved with absorbing and squeezing the sponge can be efficiently implemented in a secure distributed setting. See for example [100] for a discussion on implementing Rescue in an MPC system.

6.2 TECHNICAL OVERVIEW

As mentioned in Section 2.3 above, a standard method to construct efficient PKE schemes for large messages is the KEM-DEM paradigm [15]. In such a paradigm, the actual message is encrypted via the relatively efficient DEM component, and the one-time symmetric DEM key is transferred to the recipient via the KEM component. Let $\mathsf{KEM} = (\mathsf{KGen}^{\mathrm{kem}}, \mathsf{Encap}, \mathsf{Decap})$ be an IND-CCA secure KEM and $\mathsf{DEM} = (\mathsf{KGen}^{\mathrm{dem}}, \mathsf{Enc}^{\mathrm{dem}}, \mathsf{Dec}^{\mathrm{dem}})$ be a one-time IND-CCA secure DEM. Then the IND-CCA secure hybrid PKE scheme $\mathsf{PKE}^{\mathrm{hy}} = (\mathsf{KGen}^{\mathrm{hy}}, \mathsf{Enc}^{\mathrm{hy}}, \mathsf{Dec}^{\mathrm{hy}})$ obtained by composing the two schemes using the KEM-DEM paradigm is described in Figure 2.4. In particular, the encryption algorithm outputting (c_0, c_1) for $\mathsf{Enc}^{\mathrm{hy}}$ is along the lines of:

$$(c_0, k) \leftarrow \mathsf{Encap}(\mathsf{pk}), \;\; c_1 \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k, m).$$

However, there is a problem with this hybrid construction when one looks for a distributed variant of the decryption algorithm $\mathsf{Dec}^{\mathrm{hy}}$. Even if the decapsulation algorithm of KEM has an efficient distributed operation, one cannot derive an efficient distributed $\mathsf{Dec}^{\mathrm{hy}}$ this way since the decryption of DEM also needs to be executed in a distributed manner. And executing $\mathsf{Dec}^{\mathrm{dem}}$ in a distributed manner for standard symmetric encryption schemes is possible, but very inefficient for long messages.

One obvious way to get around this problem is for the distributed decryption operation for $\mathsf{PKE}^{\mathrm{hy}}$ to output k in the clear after the KEM distributed decapsulation has been executed – thereby enabling the decryption using DEM to be done in the clear. We call such a hybrid PKE scheme "leaky", as the decryption algorithm leaks the underlying symmetric DEM key even if the DEM ciphertext does not decrypt correctly. Now this approach might seem intuitively attractive; however, it breaks the IND-CCA security of

```
$\mathsf{Dec}^{\mathrm{hy}}_{\mathrm{leak}}(\mathsf{sk}, (c_0, c_1))$
$k \leftarrow \mathsf{Decap}(\mathsf{sk}, c_0)$
if $k = \perp$ then return $(\perp, \perp)$
$m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
return $(k, m)$
```

FIGURE 6.1: Leaky decryption functionality with respect to $\mathsf{PKE}^{\mathrm{hy}}$.

$\mathsf{PKE}^{\mathrm{hy}}$ via a trivial attack. Namely, since the decrypting parties now obtain the DEM key *before* it is known whether the key is valid for DEM, the new "leaky" decryption functionality w.r.t. $\mathsf{PKE}^{\mathrm{hy}}$ (which we will call $\mathsf{Dec}^{\mathrm{hy}}_{\mathrm{leak}}$), and the functionality of any decryption oracle given to an adversary in the IND-CCA sense, will be of the form described in Figure 6.1. This provides an immediate IND-CCA attack on the hybrid construction. An adversary can just take the target ciphertext (c_0^*, c_1^*) and submit (c_0^*, c_1) to the decryption oracle for a random value c_1. With high probability, it will receive $(k, \perp)$. Then it can use k to decrypt c_1^*, and win the security game. It is to avoid this attack that we modify the KEM-DEM framework.

Before discussing our new framework in more detail, it helps to first consider the Tag-KEM framework of [22]. At a high level, the Tag-KEM framework gives the following hybrid construction:

$$(c_0, \overline{k}) \leftarrow \mathsf{Encap}(\mathsf{pk}), \;\; (k, \mu) \leftarrow H(\overline{k}), \;\; c_1 \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k, m), \;\; c_2 \leftarrow G(c_1, \mu)$$

where G and H are hash functions.[3] This hybrid construction is secure if KEM is IND-CCA secure and DEM is one-time IND-CPA secure. And one of the applications of the Tag-KEM framework mentioned in [22] is that of threshold hybrid public-key encryption. Their argument is as follows. Since DEM in the above construction only needs to be one-time IND-CPA secure, we can instantiate it with the one-time pad. In such a case, outputting m already leaks k. Therefore, revealing the value k before applying the decryption of c_1 cannot break security, as that would contradict the main security theorem in [22]. Thus one can apply threshold decryption to obtain the decapsulation of c_0, securely evaluate the value μ and perform the

[3] In the original description of Tag-KEM framework in [22], H is replaced by a key-derivation function and G is replaced by a MAC (with μ being the MAC key). But it is not hard to see that by modelling G and H as random oracles, we get the desired properties of these primitives.

"MAC check: $c_2 = G(c_1, \mu)$" in a distributed fashion; if the check verifies, the DEM key k can be leaked and c_1 can be decrypted in the clear.

However, if DEM is the one-time pad encryption scheme, then this implies that hash function H needs to be securely evaluated to produce a key k as long as the message m. This results in an expensive distributed decryption algorithm, which defeats the whole purpose of leaking k for efficiency in the first place. To be more specific, we want both efficient hybrid distributed decryption and an efficient DEM operation. If we take an AES-based DEM, then the output of hash function H will be a bit vector in $\{0,1\}^{|k|}$. But the hash input $\overline{k}$ will be "native" to the underlying KEM, and thus in general, an element of a set such as $\mathbb{F}_p^n$ for some modulus p – which is the case for most post-quantum KEMs. This means H needs to map from one arithmetic domain to another securely in a distributed manner, which can be quite expensive in practice. In addition to this efficiency issue, it is also not clear how to formally prove security of the above Tag-KEM construction in the post-quantum QROM setting.

This brings us to our hybrid construction which overcomes the above issues. Our solution is inspired by FO-style transformations in that we start with a weakly secure, i.e., OW-CPA secure, base PKE scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$. On a high level, our construction – which we simply call "Hybrid" – outputs a ciphertext of the form (c_0, c_1, c_2, c_3) with

$$\overline{k} \leftarrow \mathcal{M},\ k \leftarrow H(\overline{k}),\ \mu \leftarrow H'(\overline{k}),$$
$$c_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, \overline{k}),\ c_1 \leftarrow \mathsf{Enc}^{\mathsf{dem}}(k, m),\ c_2 \leftarrow G(c_1, \mu),\ c_3 \leftarrow H''(\overline{k}),$$

where the hash functions G, H, H' and H'' are modelled as (quantum) random oracles, and $\mathcal{M}$ is the message space of PKE. The distributed decryption algorithm checks the c_2 and c_3 components in a distributed manner, similar to the above Tag-KEM framework, and then leaks the key $\overline{k}$ in the clear – enabling k to be produced and hence m to be decrypted from the c_1 component. We formally prove that this scheme is IND-CCA secure in the QROM – even with this form of leaky decryption – if the base scheme PKE, in addition to being OW-CPA secure, is *rigid* and *perfectly correct*, and the scheme DEM is one-time IND-CPA secure; in Subsection 6.3.1, we also discuss how our security proof can be extended to the case when PKE is not perfectly correct.

In contrast to the Tag-KEM framework, DEM can be instantiated with *any* one-time IND-CPA secure scheme – i.e., DEM need not be the one-time pad – in our Hybrid transform to achieve an efficient distributed decryption algorithm. One of the reasons for this efficiency is that, unlike the Tag-

KEM construction, we leak the key $\bar{k}$ and not k, thereby precluding the hash function H from mapping between different arithmetic domains in a secure distributed manner. This same problem does not occur with G, H' and H'' as we are free to select these hash functions so that they can be efficiently evaluated in a secure distributed setting; in other words, we can have G, H' and H'' to be MPC-friendly hash functions such as Rescue. On the downside however, we have an additional ciphertext component "$c_3 = H''(\bar{k})$" where H'' is a *length-preserving* hash – i.e., H'' has domain and co-domain equal to $\mathcal{M}$. We rely on this extra component to obtain a security proof in the QROM using the techniques in [50].[4] In light of more powerful proof techniques introduced in recent QROM literature (e.g., in [57]), it is an interesting open problem to come up with an alternative transform to Hybrid which has smaller ciphertexts, while at the same time, admits a security proof in the QROM.

6.3 THE Hybrid CONSTRUCTION

In this section, we formally describe our generic "Hybrid" construction, and prove the IND-CCA security of resulting hybrid PKE schemes – with respect to the leaky decryption functionality discussed in Section 6.2 above – in the QROM.

Let $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic and rigid PKE scheme with message space $\mathcal{M}$. Also let $\mathsf{DEM} = (\mathsf{KGen}^{\mathrm{dem}}, \mathsf{Enc}^{\mathrm{dem}}, \mathsf{Dec}^{\mathrm{dem}})$ be a randomized DEM scheme with key space $\mathcal{K}$ and ciphertext space $\mathcal{C}$; we additionally assume that $\mathsf{KGen}^{\mathrm{dem}}$ generates a symmetric key $k \leftarrow\$ \mathcal{K}$ uniformly at random. For our hybrid construction, we define the following four hash functions:

$$\begin{aligned} H &: \mathcal{M} \longrightarrow \mathcal{K}, \\ H', H'' &: \mathcal{M} \longrightarrow \mathcal{M}, \\ G &: \mathcal{C} \times \mathcal{M} \longrightarrow \mathcal{F}. \end{aligned}$$

Here the co-domain $\mathcal{F}$ of function G is to be interpreted as a finite field $\mathbb{F}_q^t$, for some prime q, which is typically the case for MPC-friendly hash

[4] The authors of [50] modified the original FO transformation in [1] by adding a similar ciphertext component so as to obtain a security proof in the QROM. It was later discovered that the proof had gaps in it [37]. On a high level, the gaps were related to the fact that base PKE schemes used in the FO transform can be randomized. A similar issue does not arise in our QROM security proof for the Hybrid transform because we assume deterministic base PKE scheme.

functions such as Rescue (see Subsection 6.1.2); because after all, one needs to instantiate G, H' and H'' with such functions in order to achieve an efficient hybrid threshold decryption operation using our Hybrid transform.

Now our hybrid PKE construction[5] $\mathsf{PKE}^{\text{hy}} = (\mathsf{KGen}^{\text{hy}}, \mathsf{Enc}^{\text{hy}}, \mathsf{Dec}^{\text{hy}})$ obtained via the Hybrid transform is described in Figure 6.2. Notice how the decryption function leaks the key $\bar{k}$ even when the decryption function $\mathsf{Dec}^{\text{dem}}$ might fail. This will allow us, in our threshold decryption operation, to also leak this key before the algorithm $\mathsf{Dec}^{\text{dem}}$ is called – thereby enabling $\mathsf{Dec}^{\text{dem}}$ to be applied in the clear; a more detailed description of the threshold operation is presented in Subsection 6.3.2 below. The only question now is whether leaking this key is secure. The attack described in the previous section w.r.t. "thresholdizing" the standard KEM-DEM framework does not apply to our construction, as an invalid ciphertext will get rejected (with high probability) by the "$c_3 = H''(\bar{k})$" and "$c_2 = G(c_1, H'(\bar{k}))$" checks; in such a case, the key $\bar{k}$ will not be leaked to the adversary.

In fact, the following theorem shows that – in the QROM – our hybrid construction PKE^{hy} is IND-CCA secure in a model where the key $\bar{k}$ leaks during decryption as above, when the underlying PKE is OW-CPA secure and perfectly correct (in Subsection 6.3.1, we discuss how to handle PKE with correctness errors), and DEM is *any* one-time IND-CPA secure scheme.

Theorem 19. *Let* $\mathsf{PKE}^{\text{hy}} = (\mathsf{KGen}^{\text{hy}}, \mathsf{Enc}^{\text{hy}}, \mathsf{Dec}^{\text{hy}})$ *be the hybrid PKE construction obtained by composing a deterministic* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *and a (randomized)* $\mathsf{DEM} = (\mathsf{KGen}^{\text{dem}}, \mathsf{Enc}^{\text{dem}}, \mathsf{Dec}^{\text{dem}})$ *via the* Hybrid *transform (see Fig. 6.2). Suppose* PKE *is perfectly correct and rigid (with message space $\mathcal{M}$). Then for any* IND-CCA *adversary $\mathcal{A}_{\text{hy}}$ against* PKE^{hy} *issuing at-most q_G, q_H, $q_{H'}$ and $q_{H''}$ queries to the quantum random oracles G, H, H' and H'' respectively, and at-most q_D queries to the (classical) decryption oracle, there exist a* one-time IND-CPA *adversary $\mathcal{A}_{\text{dem}}$ against* DEM, *and* OW-CPA *adversaries $\mathcal{A}_{\text{pke}}$ and $\mathcal{A}'_{\text{pke}}$ against* PKE *such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}^{\text{hy}}}(\mathcal{A}_{\text{hy}}) \leq \mathbf{Adv}^{\mathsf{otIND\text{-}CPA}}_{\mathsf{DEM}}(\mathcal{A}_{\text{dem}})$$

$$+ 2(q_H + q_{H'})\sqrt{\frac{2q_G}{\sqrt{|\mathcal{M}|}} + \frac{q_D}{|\mathcal{F}|} + \mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\text{pke}}) + 2q_{H''}\sqrt{\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}'_{\text{pke}})}}.$$

[5] Note that we use the term "construction" – and not "scheme" – when referring to PKE^{hy} since, technically speaking, it is not a PKE scheme in the sense of Definition 1; this is because the decryption algorithm Dec^{hy} outputs/leaks the value $\bar{k}$ in addition to message m. In other words, PKE^{hy} should not be seen as a PKE scheme "in isolation", but instead should be viewed in context of the threshold application we are considering in this chapter.

```
$\mathsf{KGen}^{\mathrm{hy}}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
return $(\mathsf{pk}, \mathsf{sk})$
```

```
$\mathsf{Enc}^{\mathrm{hy}}(\mathsf{pk}, m)$
$\overline{k} \leftarrow \mathcal{M}$
$k \leftarrow H(\overline{k})$
$\mu \leftarrow H'(\overline{k})$
$c_0 := \mathsf{Enc}(\mathsf{pk}, \overline{k})$
$c_1 \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k, m)$
$c_2 \leftarrow G(c_1, \mu)$
$c_3 \leftarrow H''(\overline{k})$
$c := (c_0, c_1, c_2, c_3)$
return $c$
```

```
$\mathsf{Dec}^{\mathrm{hy}}(\mathsf{sk}, c)$
Parse $c = (c_0, c_1, c_2, c_3)$
$\overline{k} := \mathsf{Dec}(\mathsf{sk}, c_0)$
if $\overline{k} = \perp$ then return $(\perp, \perp)$
$t \leftarrow H''(\overline{k})$
if $t \neq c_3$ then return $(\perp, \perp)$
$\mu \leftarrow H'(\overline{k})$
$t' \leftarrow G(c_1, \mu)$
if $t' \neq c_2$ then return $(\perp, \perp)$
$k \leftarrow H(\overline{k})$
$m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
return $(\overline{k}, m)$
```

FIGURE 6.2: Hybrid construction.

Here the running times of $\mathcal{A}_{\mathrm{dem}}$, $\mathcal{A}_{\mathrm{pke}}$ and $\mathcal{A}'_{\mathrm{pke}}$ are about the same as that of $\mathcal{A}_{\mathrm{hy}}$. More importantly, note that the responses to decryption oracle queries made by $\mathcal{A}_{\mathrm{hy}}$ leak the key $\overline{k}$ as noted above in Fig. 6.2.

Proof. Denote $\Omega_{\mathbf{G}}$, $\Omega_{\mathbf{H}}$ and $\Omega_{\mathbf{H}'}$ to be the set of all functions $\mathbf{G} : \mathcal{C} \times \mathcal{M} \to \mathcal{F}$, $\mathbf{H} : \mathcal{M} \to \mathcal{K}$ and $\mathbf{H}' : \mathcal{M} \to \mathcal{M}$ respectively, where $\mathcal{M}$ is the message space of PKE, and $\mathcal{K}$ and $\mathcal{C}$ are the key space and ciphertext space of DEM respectively.

Let $\mathcal{A}_{\mathrm{hy}}$ be an adversary against the IND-CCA security of $\mathsf{PKE}^{\mathrm{hy}}$ issuing at most q_G, q_H, $q_{H'}$ and $q_{H''}$ quantum queries to the random oracles G, H, H' and H'' respectively and at most q_D classical decryption queries. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_{12}$ described in Figures 6.3, 6.5 and 6.7.

Game G_0: The game G_0 is exactly the IND-CCA security game associated with $\mathsf{PKE}^{\mathrm{hy}}$. Hence, we have

$$\left| \Pr[\mathsf{G}_0 = 1] - \frac{1}{2} \right| = \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}^{\mathrm{hy}}}(\mathcal{A}_{\mathrm{hy}}).$$

Game G_1: In game G_1, we introduce some cosmetic changes to the setup. Namely, we generate the values $\overline{k}^*$, k^*, μ^* and c_0^* before $\mathcal{A}_{\mathrm{hy}}$ outputs the pair of challenge messages (m_0, m_1). This does not affect $\mathcal{A}_{\mathrm{hy}}$'s view in

```
Games $G_0$ - $G_3$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$G \leftarrow\$ \Omega_{\mathbf{G}}$; $H \leftarrow\$ \Omega_{\mathbf{H}}$
$H' \leftarrow\$ \Omega_{\mathbf{H'}}$; $H'' \leftarrow\$ \Omega_{\mathbf{H'}}$
$\overline{k}^* \leftarrow\$ \mathcal{M}$    // $G_1$–$G_3$
$k^* \leftarrow H(\overline{k}^*)$; $\mu^* \leftarrow H'(\overline{k}^*)$    // $G_1$–$G_2$
$k^* \leftarrow\$ \mathcal{K}$; $\mu^* \leftarrow\$ \mathcal{M}$    // $G_3$
$c_0^* := \mathsf{Enc}(\mathsf{pk}, \overline{k}^*)$    // $G_1$–$G_3$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\mathrm{DEC}_{\perp}^{\mathrm{hy}}}(\mathsf{pk})$
$b \leftarrow\$ \{0,1\}$
$\overline{k}^* \leftarrow\$ \mathcal{M}$    // $G_0$
$k^* \leftarrow H(\overline{k}^*)$; $\mu^* \leftarrow H'(\overline{k}^*)$    // $G_0$
$c_0^* := \mathsf{Enc}(\mathsf{pk}, \overline{k}^*)$    // $G_0$
$c_1^* \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$
$c_2^* \leftarrow G(c_1^*, \mu^*)$
$c_3^* \leftarrow H''(\overline{k}^*)$
$c^* := (c_0^*, c_1^*, c_2^*, c_3^*)$
$b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\mathrm{DEC}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
return $[b' = b]$
```

```
$\mathrm{DEC}_a^{\mathrm{hy}}(c)$
if $c = a$ then return $\perp$
Parse $c = (c_0, c_1, c_2, c_3)$
$\overline{k} := \mathsf{Dec}(\mathsf{sk}, c_0)$
if $\overline{k} = \perp$ then
    $\overline{k}' \leftarrow\$ \mathcal{M}$; query $H''(\overline{k}')$    // $G_2$–$G_3$
    return $(\perp, \perp)$
$t \leftarrow H''(\overline{k})$
if $t \neq c_3$ then return $(\perp, \perp)$
if $c_0 = c_0^*$ then    // $G_2$–$G_3$
    $\mu \leftarrow \mu^*$; $t' \leftarrow G(c_1, \mu^*)$    // $G_2$–$G_3$
else $\mu \leftarrow H'(\overline{k})$; $t' \leftarrow G(c_1, \mu)$
if $t' \neq c_2$ then return $(\perp, \perp)$
$k \leftarrow H(\overline{k})$; $m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
return $(\overline{k}, m)$
```

FIGURE 6.3: Games G_0 – G_{12} for the proof of Theorem 19.

any way when it queries the oracles G, H, H', H'' and $\textsc{Dec}^{\mathrm{hy}}_{\perp}$ in the "pre-challenge phase" (i.e., before $\mathcal{A}_{\mathrm{hy}}$ outputs (m_0, m_1)). Hence,

$$\Pr[\mathsf{G}_1 = 1] = \Pr[\mathsf{G}_0 = 1].$$

Game G_2**:** In game G_2, we modify the decryption oracle $\textsc{Dec}^{\mathrm{hy}}_{a}$ (with $a \in \{\perp, c^*\}$) as follows: if $c_0 = c_0^*$, then we replace the hash evaluation "$\mu \leftarrow H'(\bar{k})$" with "$\mu \leftarrow \mu^*$". (We also make another cosmetic change to $\textsc{Dec}^{\mathrm{hy}}_{a}$ where, if $\bar{k} = \perp$, we make a *classical* H''-query on a uniformly random $\bar{k}' \leftarrow\!\!\$\ \mathcal{M}$. This change will become apparent later on when we make further modifications to $\textsc{Dec}^{\mathrm{hy}}_{a}$ that allows us to decrypt any ciphertext without using sk.) Note that the games G_1 and G_2 are equivalent unless there is a decryption failure w.r.t. the ciphertext c_0^*. But since we assumed that PKE is perfectly correct, we have

$$\Pr[\mathsf{G}_2 = 1] = \Pr[\mathsf{G}_1 = 1].$$

Game G_3**:** In the setup of game G_3, we replace the hash evaluations "$k^* \leftarrow H(\bar{k}^*)$" and $\mu^* \leftarrow H'(\bar{k}^*)$" with "$k^* \leftarrow\!\!\$\ \mathcal{K}$" and "$\mu^* \leftarrow\!\!\$\ \mathcal{M}$" respectively. That is, k^* and μ^* are now uniformly random values that are generated independently of the QROs H and H' respectively. We first bound the success probability of $\mathcal{A}_{\mathrm{hy}}$ in G_3 via a reduction to the one-time IND-CPA security (i.e., otIND-CPA security; see Definition 11) of DEM. Let $\mathcal{A}_{\mathrm{dem}}$ be a one-time IND-CPA adversary against DEM that works as follows:

- Runs $\mathsf{KGen}^{\mathrm{hy}}$ to obtain $(\mathsf{pk}, \mathsf{sk})$.
- Generates $\bar{k}^* \leftarrow\!\!\$\ \mathcal{M}$, $\mu^* \leftarrow\!\!\$\ \mathcal{M}$ and computes $c_0^* := \mathsf{Enc}(\mathsf{pk}, \bar{k}^*)$.
- Uses a $2q_G$-wise independent function, $2q_H$-wise independent function, $2q_{H'}$-wise independent function and $2q_{H''}$-wise independent function to simulate the QROs G, H, H' and H'' respectively, as noted in Lemma 1.
- Runs $\mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}^{\mathrm{hy}}_{\perp}}(\mathsf{pk})$ by answering the quantum random oracle queries and classical decryption queries as in G_3, and finally obtains (m_0, m_1).
- Forwards (m_0, m_1) to its one-time IND-CPA challenger and gets the ciphertext c_1^* in return. Note that the uniform secret key k^* is generated

implicitly by the challenger (i.e., $k^* \leftarrow\!\!\$\ \mathcal{K}$[6]) as well as the bit b ($\leftarrow\!\!\$\ \{0,1\}$). Thus, we have $c_1^* \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$.

- Computes $c_2^* \leftarrow G(c_1^*, \mu^*)$ and $c_3^* \leftarrow H''(\overline{k}^*)$.
- Runs $\mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}_{c^*}^{\mathrm{hy}}}(c^*)$, where $c^* = (c_0^*, c_1^*, c_2^*, c_3^*)$ by answering the random oracle queries and decryption queries as in G_3, and finally obtains a bit b'.
- Forwards bit b' to its one-time IND-CPA challenger as the final output.

$A^{H\times H'}(\overline{k}^*, (k^*, \mu^*))$

```
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$G \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H'}}$
$c_0^* := \mathsf{Enc}(\mathsf{pk}, \overline{k}^*)$
$(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}_{\perp}^{\mathrm{hy}}}(\mathsf{pk})$
$b \leftarrow\!\!\$\ \{0,1\}$
$c_1^* \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$
$c_2^* \leftarrow G(c_1^*, \mu^*)$
$c_3^* \leftarrow H''(\overline{k}^*)$
$c^* := (c_0^*, c_1^*, c_2^*, c_3^*)$
$b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
**return** $[b' = b]$
```

$\textsc{Dec}_a^{\mathrm{hy}}(c)$

```
**if** $c = a$ **then return** $\perp$
Parse $c = (c_0, c_1, c_2, c_3)$
$\overline{k} := \mathsf{Dec}(\mathsf{sk}, c_0)$
**if** $\overline{k} = \perp$ **then**
    $\overline{k}' \leftarrow\!\!\$\ \mathcal{M}$; query $H''(\overline{k}')$
    **return** $(\perp, \perp)$
$t \leftarrow H''(\overline{k})$
**if** $t \neq c_3$ **then return** $(\perp, \perp)$
**if** $c_0 = c_0^*$ **then**
    $\mu \leftarrow \mu^*$; $t' \leftarrow G(c_1, \mu^*)$
**else** $\mu \leftarrow H'(\overline{k})$; $t' \leftarrow G(c_1, \mu)$
**if** $t' \neq c_2$ **then return** $(\perp, \perp)$
$k \leftarrow H(\overline{k})$; $m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
**return** $(\overline{k}, m)$
```

FIGURE 6.4: Algorithm $A^{H\times H'}$ for the proof of Theorem 19.

It is easy to see that

$$\left|\Pr[\mathsf{G}_3 = 1] - \frac{1}{2}\right| = \mathbf{Adv}_{\mathsf{DEM}}^{\mathsf{otIND\text{-}CPA}}(\mathcal{A}_{\mathrm{dem}}).$$

[6] Since we assumed that $\mathsf{KGen}^{\mathrm{dem}}$ generates keys uniformly at random from the key space $\mathcal{K}$ (see Fig. 6.2).

Game G_4**:** Now using the OW2H lemma (Lemma 3), we bound the difference between the success probabilities of $\mathcal{A}_{\text{hy}}$ in G_2 and G_3.[7] Let A be an oracle algorithm that has quantum access to the random oracle $H \times H'$, where $(H \times H')(\bar{k}) = (H(\bar{k}), H'(\bar{k}))$. Figure 6.4 describes $A^{H\times H'}$'s operation on input $(\bar{k}^*, (k^*, \mu^*))$. Note that the algorithm $A^{H\times H'}$ makes at most $q_H + q_{H'}$ number of queries to the random oracle $H \times H'$ to respond to $\mathcal{A}_{\text{hy}}$'s oracle queries[8].

With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_2 = 1]$ and $P_A^2 = \Pr[\mathsf{G}_3 = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 3 w.r.t. the algorithm $A^{H\times H'}$. To analyze the corresponding probability P_B in Lemma 3, we hence define game G_4 (see Fig. 6.5) such that $P_B = \Pr[\mathsf{G}_4 = 1]$. From Lemma 3, we thus have

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq 2(q_H + q_{H'})\sqrt{\Pr[\mathsf{G}_4 = 1]}.$$

Game G_5**:** In game G_5, we replace the evaluations of oracle $G(.,\mu^*)$ with that of a truly random quantum oracle $R(.)$. Specifically, let $\Omega_{\mathbf{R}}$ be the set of all functions $\mathbf{R} : \mathcal{C} \to \mathcal{F}$. Then $R(\leftarrow\$ \, \Omega_{\mathbf{R}})$ is an internal oracle that is not directly accessible by $\mathcal{A}_{\text{hy}}$. We can justify this replacement using Lemma 2 w.r.t. the pseudorandomness of $G(.,\mu^*)$, with PRF key $\mu^* \leftarrow\$ \, \mathcal{M}$, to obtain the following via a straightforward reduction:

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_5 = 1]| \; \leq \; \frac{2q_G}{\sqrt{|\mathcal{M}|}}.$$

Game G_6**:** In game G_6, we modify the decryption oracle as follows: if $c_0 = c_0^*$, return $(\bot, \bot)$. (We also make a cosmetic change where we replace "$c_2^* \leftarrow R(c_1^*)$" with "$c_2^* \leftarrow\$ \, \mathcal{F}$", since the random function R would have *only* been used on c_1^* throughout G_6 and no other c_1-values.) Note that the only way the execution of games G_5 and G_6 would differ is if $\mathcal{A}_{\text{hy}}$ made decryption queries of the form $c = (c_0^*, c_1, c_2, c_3^*)$ where $c_1 \neq c_1^*$ and $R(c_1) = c_2$; also in such an event, the number of H-queries with argument $\bar{k}^*$ in G_5 and G_6 will go "out of sync" resulting in a difference in $\mathcal{A}_{\text{hy}}$'s respective success probabilities. Since R is an internal random oracle not directly accessible by $\mathcal{A}_{\text{hy}}$, we can bound the probability of the event

[7] Here we note that one could use more recent variants of the OW2H lemma – e.g., as proposed in [37, 51, 52] – to obtain a tighter security proof for PKE^{hy} in the QROM (also see Footnote 12 of this chapter).

[8] If $A^{H\times H'}$ wants to respond to $\mathbf{A}$'s H-query, then $A^{H\times H'}$ prepares a uniform superposition of all states in the output register corresponding to H' (see Footnote 3 of Chapter 3, and also [50]).

```
Games $\mathsf{G}_4$ - $\mathsf{G}_7$, $\mathsf{G}_9$, $\mathsf{G}_{11}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$G \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H' \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$
$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H}'}$   // $\mathsf{G}_4 - \mathsf{G}_7$
$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{poly}}$   // $\mathsf{G}_9$, $\mathsf{G}_{11}$
$R \leftarrow\!\!\$\ \Omega_{\mathbf{R}}$   // $\mathsf{G}_5$
$\overline{k}^* \leftarrow\!\!\$\ \mathcal{M}; k^* \leftarrow\!\!\$\ \mathcal{K}; \mu^* \leftarrow\!\!\$\ \mathcal{M}$
$c_0^* := \mathsf{Enc}(\mathsf{pk}, \overline{k}^*)$
$i \leftarrow\!\!\$\ \{1, \ldots, q_H + q_{H'}\}$
run until $i$-th query to oracle $H \times H'$
    $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{G, H, H', H'', \textsc{Dec}_{\perp}^{\mathrm{hy}}}(\mathsf{pk})$
    $b \leftarrow\!\!\$\ \{0, 1\}$
    $c_1^* \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$
    $c_2^* \leftarrow G(c_1^*, \mu^*)$   // $\mathsf{G}_4$
    $c_2^* \leftarrow R(c_1^*)$   // $\mathsf{G}_5$
    $c_2^* \leftarrow\!\!\$\ \mathcal{F}$   // $\mathsf{G}_6$–$\mathsf{G}_7$, $\mathsf{G}_9$, $\mathsf{G}_{11}$
    $c_3^* \leftarrow H''(\overline{k}^*)$   // $\mathsf{G}_4$–$\mathsf{G}_6$
    $c_3^* \leftarrow\!\!\$\ \mathcal{M}$   // $\mathsf{G}_7$, $\mathsf{G}_9$, $\mathsf{G}_{11}$
    $c^* := (c_0^*, c_1^*, c_2^*, c_3^*)$
    $b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{G, H, H', H'', \textsc{Dec}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
measure the argument $\overline{k}'$ of the
 $i$-th query to oracle $H \times H'$
return $[\overline{k}' = \overline{k}^*]$
```

```
$\textsc{Dec}_a^{\mathrm{hy}}(c_0, c_1, c_2, c_3)$   // $\mathsf{G}_4$ - $\mathsf{G}_7$, $\mathsf{G}_9$
if $c_0 = c_0^*$ then
  return $(\perp, \perp)$   // $\mathsf{G}_6$–$\mathsf{G}_7$, $\mathsf{G}_9$
$\overline{k} := \mathsf{Dec}(\mathsf{sk}, c_0)$
if $\overline{k} = \perp$ then
  $\overline{k}' \leftarrow\!\!\$\ \mathcal{M}$; query $H''(\overline{k}')$
  return $(\perp, \perp)$
$t \leftarrow H''(\overline{k})$
if $t \neq c_3$ then return $(\perp, \perp)$
if $c_0 = c_0^*$ then   // $\mathsf{G}_4$–$\mathsf{G}_5$
  $\mu \leftarrow \mu^*; t' \leftarrow G(c_1, \mu^*)$   // $\mathsf{G}_4$
  $t' \leftarrow R(c_1)$   // $\mathsf{G}_5$
else $\mu \leftarrow H'(\overline{k}); t' \leftarrow G(c_1, \mu)$
if $t' \neq c_2$ then return $(\perp, \perp)$
$k \leftarrow H(\overline{k}); m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
return $(\overline{k}, m)$
```

```
$\textsc{Dec}_a^{\mathrm{hy}}(c_0, c_1, c_2, c_3)$   // $\mathsf{G}_{11}$
if $c_0 = c_0^*$ then
  return $(\perp, \perp)$
Compute set of roots $S$
  of polynomial $H''(x) - c_3$
if $\exists \overline{k} \in S$ s.t. $\mathsf{Enc}(\mathsf{pk}, \overline{k}) = c_0$
 then
  query $H''(\overline{k})$
  $\mu \leftarrow H'(\overline{k}); t' \leftarrow G(c_1, \mu)$
  if $t' \neq c_2$ then
    return $(\perp, \perp)$
  else $k \leftarrow H(\overline{k}); m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
    return $(\overline{k}, m)$
else $\overline{k}' \leftarrow\!\!\$\ \mathcal{M}$; query $H''(\overline{k}')$
  return $(\perp, \perp)$
```

FIGURE 6.5: Games $\mathsf{G}_4 - \mathsf{G}_7$, G_9, G_{11} for the proof of Theorem 19. Also $\mathcal{A}_{\mathrm{hy}}$ does not make (classical) queries to oracle $\textsc{Dec}_a^{\mathrm{hy}}$ of the form $(c_0, c_1, c_2, c_3) = a$.

"$R(c_1) = c_2$" w.r.t. a *single* decryption query $c = (c_0^*, c_1, c_2, c_3^*)$ by $1/|\mathcal{F}|$. Using a union bound, we conclude that

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \frac{q_D}{|\mathcal{F}|}.$$

```
$\hat{A}^{H''}(\overline{k}^*, c_3^*)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
    $G \leftarrow\$ \Omega_{\mathbf{G}}; H \leftarrow\$ \Omega_{\mathbf{H}}; H' \leftarrow\$ \Omega_{\mathbf{H}'}$
    $k^* \leftarrow\$ \mathcal{M}$
    $c_0^* := \mathsf{Enc}(\mathsf{pk}, \overline{k}^*)$
    $i \leftarrow\$ \{1, \ldots, q_H + q_{H'}\}$
    run until $i$-th query to oracle $H \times H'$
        $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}_{\perp}^{\mathrm{hy}}}(\mathsf{pk})$
        $b \leftarrow\$ \{0,1\}$
        $c_1^* \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$
        $c_2^* \leftarrow\$ \mathcal{F}$
        $c^* := (c_0^*, c_1^*, c_2^*, c_3^*)$
        $b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
    measure the argument $\overline{k}'$ of the
     $i$-th query to oracle $H \times H'$
    return $[\overline{k}' = \overline{k}^*]$
```

```
$\textsc{Dec}_a^{\mathrm{hy}}(c)$
    if $c = a$ then return $\perp$
    Parse $c = (c_0, c_1, c_2, c_3)$
    if $c_0 = c_0^*$ then return $(\perp, \perp)$
    $\overline{k} := \mathsf{Dec}(\mathsf{sk}, c_0)$
    if $\overline{k} = \perp$ then
        $\overline{k}' \leftarrow\$ \mathcal{M}$; query $H''(\overline{k}')$
        return $(\perp, \perp)$
    $t \leftarrow H''(\overline{k})$
    if $t \neq c_3$ then return $(\perp, \perp)$
    $\mu \leftarrow H'(\overline{k})$; $t' \leftarrow G(c_1, \mu)$
    if $t' \neq c_2$ then return $(\perp, \perp)$
    $k \leftarrow H(\overline{k})$; $m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
    return $(\overline{k}, m)$
```

FIGURE 6.6: Algorithm $\hat{A}^{H''}$ for the proof of Theorem 19.

Games G_7 **and** G_8**:** In the setup of game G_7, we replace the computation "$c_3^* \leftarrow H''(\overline{k}^*)$" with "$c_3^* \leftarrow\$ \mathcal{M}$". That is, c_3^* is now a uniformly random value that is generated independently of $\overline{k}^*$ and the QRO H''. Using Lemma 3, we bound the difference between the success probabilities of $\mathcal{A}_{\mathrm{hy}}$ in G_6 and G_7. Let $\hat{A}$ be an algorithm that has quantum access to the random oracle H''. Figure 6.6 describes $\hat{A}^{H''}$'s operation on input $(\overline{k}^*, c_3^*)$. Note that the algorithm $\hat{A}^{H''}$ makes at most $q_{H''}$ queries to the random oracle H'' to respond to $\mathcal{A}_{\mathrm{hy}}$'s oracle queries.

Games G_8 and G_{10}

```
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$G \leftarrow\!\!\$\ \Omega_{\mathbf{G}}; H \leftarrow\!\!\$\ \Omega_{\mathbf{H}}; H' \leftarrow\!\!\$\ \Omega_{\mathbf{H'}}$
$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H'}}$   // $\mathsf{G}_8$
$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{poly}}$   // $\mathsf{G}_{10}, \mathsf{G}_{12}$
$\overline{k}^* \leftarrow\!\!\$\ \mathcal{M}; k^* \leftarrow\!\!\$\ \mathcal{K}; \mu^* \leftarrow\!\!\$\ \mathcal{M}$
$c_0^* := \mathsf{Enc}(\mathsf{pk}, \overline{k}^*)$
$j \leftarrow\!\!\$\ \{1, \dots, q_{H''}\}$
run until $j$-th query to oracle $H''$
   $i \leftarrow\!\!\$\ \{1, \dots, q_H + q_{H'}\}$
   run until $i$-th query to oracle $H \times H'$
      $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\mathrm{Dec}_{\perp}^{\mathrm{hy}}}(\mathsf{pk})$
      $b \leftarrow\!\!\$\ \{0,1\}$
      $c_1^* \leftarrow \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$
      $c_2^* \leftarrow\!\!\$\ \mathcal{F}$
      $c_3^* \leftarrow\!\!\$\ \mathcal{M}$
      $c^* := (c_0^*, c_1^*, c_2^*, c_3^*)$
      $b' \leftarrow \mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\mathrm{Dec}_{c^*}^{\mathrm{hy}}}(c^*, \mathsf{st})$
   measure the argument $\overline{k}'$ of the
    $i$-th query to oracle $H \times H'$
measure the argument $\overline{k}''$ of the
 $j$-th query to oracle $H''$
return $[\overline{k}'' = k^*]$
```

$\mathrm{Dec}_a^{\mathrm{hy}}(c_0, c_1, c_2, c_3)$ // $\mathsf{G}_8, \mathsf{G}_{10}$

```
if $c_0 = c_0^*$ then
   return $(\perp, \perp)$
$\overline{k} := \mathsf{Dec}(\mathsf{sk}, c_0)$
if $\overline{k} = \perp$ then
   $\overline{k}' \leftarrow\!\!\$\ \mathcal{M}$; query $H''(\overline{k}')$
   return $(\perp, \perp)$
$t \leftarrow H''(\overline{k})$
if $t \neq c_3$ then return $(\perp, \perp)$
else $\mu \leftarrow H'(\overline{k})$; $t' \leftarrow G(c_1, \mu)$
if $t' \neq c_2$ then return $(\perp, \perp)$
$k \leftarrow H(\overline{k})$; $m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
return $(\overline{k}, m)$
```

$\mathrm{Dec}_a^{\mathrm{hy}}(c_0, c_1, c_2, c_3)$ // G_{12}

```
if $c_0 = c_0^*$ then
   return $(\perp, \perp)$
Compute set of roots $S$
   of polynomial $H''(x) - c_3$
if $\exists \overline{k} \in S$ s.t. $\mathsf{Enc}(\mathsf{pk}, \overline{k}) = c_0$
 then
   query $H''(\overline{k})$
   $\mu \leftarrow H'(\overline{k})$; $t' \leftarrow G(c_1, \mu)$
   if $t' \neq c_2$ then
      return $(\perp, \perp)$
   else $k \leftarrow H(\overline{k})$
      $m := \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$
      return $(\overline{k}, m)$
else $\overline{k}' \leftarrow\!\!\$\ \mathcal{M}$; query $H''(\overline{k}')$
   return $(\perp, \perp)$
```

FIGURE 6.7: Games G_8, G_{10} and G_{12} for the proof of Theorem 19. Also $\mathcal{A}_{\mathrm{hy}}$ does not make (classical) queries to oracle $\mathrm{Dec}_a^{\mathrm{hy}}$ of the form $(c_0, c_1, c_2, c_3) = a$.

With the above construction of $\hat{A}$, we have $P^1_{\hat{A}} = \Pr[\mathsf{G}_6 = 1]$ and $P^2_{\hat{A}} = \Pr[\mathsf{G}_7 = 1]$, where $P^1_{\hat{A}}$ and $P^2_{\hat{A}}$ are as defined in Lemma 3 w.r.t. the algorithm $\hat{A}^{H''}$. Therefore, we now define game G_8 (see Figure 6.7) to analyze the corresponding probability $P_{\hat{B}}$ in Lemma 3; note here that $P_{\hat{B}} = \Pr[\mathsf{G}_8 = 1]$. From Lemma 3, we thus have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]| \ \leq \ 2q_{H''}\sqrt{\Pr[\mathsf{G}_8 = 1]}.$$

Games G_9 and G_{10}: In games G_9 and G_{10}, we replace the random oracle H'' with a $2q_{H''}$-wise independent function, following Lemma 1. Random polynomials of degree $2q_{H''} - 1$ over the finite field representation of $\mathcal{M}$ are $2q_{H''}$-wise independent. Let $\Omega_{\mathbf{poly}}$ be the set of all such polynomials. Then specifically, we are replacing the step "$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{H'}}$" with "$H'' \leftarrow\!\!\$\ \Omega_{\mathbf{poly}}$" in both games. From Lemma 1, as this change is indistinguishable when the oracle H'' is queried at most $q_{H''}$ times, we have G_7 and G_9 (respectively, G_8 and G_{10}) to be equivalent. Therefore,

$$\Pr[\mathsf{G}_7 = 1] = \Pr[\mathsf{G}_9 = 1], \quad \Pr[\mathsf{G}_8 = 1] = \Pr[\mathsf{G}_{10} = 1].$$

Games G_{11} and G_{12}: In G_{11} and G_{12}, we modify the decryption oracle – the same way in both games (Fig. 6.5 describes G_{11} and Fig. 6.7 describes G_{12}, respectively) – such that the secret key sk is not used to decrypt a ciphertext $c = (c_0, c_1, c_2, c_3)$. To analyze this change to $\textsc{Dec}^{\mathrm{hy}}_a$, we define two "bad" events in games G_9 – G_{12}:

- Let bad_1 denote the event that $\mathcal{A}_{\mathrm{hy}}$ asks for the decryption of $c = (c_0, c_1, c_2, c_3)$ such that c_0 is a ciphertext for which there are two *distinct* messages $\overline{k}, \overline{k}'$ that encrypt to it – i.e., $\mathsf{Enc}(\mathsf{pk}, \overline{k}) = \mathsf{Enc}(\mathsf{pk}, \overline{k}') = c_0$.
- Let bad_2 denote the event that $\mathcal{A}_{\mathrm{hy}}$ asks for the decryption of $c = (c_0, c_1, c_2, c_3)$ such that $\mathsf{Dec}(\mathsf{sk}, c_0) = \perp$ *and* there exists a root $\overline{k}'$ of the polynomial $H''(x) - c_3$ (recall that H'' in now a random *polynomial* of degree $2q_{H''} - 1$) which satisfies $\mathsf{Enc}(\mathsf{pk}, \overline{k}') = c_0$.

Setting $\mathsf{bad} = \mathsf{bad}_1 \vee \mathsf{bad}_2$, we have the following w.r.t. games G_9 and G_{10}:

$$\begin{aligned} \Pr[\mathsf{G}_9 = 1] &\leq \Pr[\mathsf{bad}] + \Pr[\neg\mathsf{bad}]\Pr[\mathsf{G}_9 = 1 \mid \neg\mathsf{bad}] \\ \Pr[\mathsf{G}_{10} = 1] &\leq \Pr[\mathsf{bad}] + \Pr[\neg\mathsf{bad}]\Pr[\mathsf{G}_{10} = 1 \mid \neg\mathsf{bad}] \end{aligned}$$

Now in order to show that $\Pr[\mathsf{G}_9 = 1|\neg\mathsf{bad}] \leq \Pr[\mathsf{G}_{11} = 1|\neg\mathsf{bad}]$ and $\Pr[\mathsf{G}_{10} = 1|\neg\mathsf{bad}] \leq \Pr[\mathsf{G}_{12} = 1|\neg\mathsf{bad}]$, it is sufficient to show, assuming the event $\neg\mathsf{bad}$ occurs: (1) the new decryption oracle returns the same output as the previous oracle when queried on any ciphertext, (2) the queries submitted to the random oracles H, H' and H'' remain "in sync" after this modification to $\textsc{Dec}_a^{\mathrm{hy}}$ (e.g., the j-th query to H'' at a particular stage in G_{10} corresponds to the j-th query to H'' in the *same* stage of G_{12}), and (3) upon measuring the argument of the i-th query to oracle $(H \times H')$ in G_9 and G_{11} (resp., the j-th query to oracle H'' in G_{10} and G_{12}), the probability of the outcome being $\overline{k}^*$ in G_{11} (resp., G_{12}) is greater than or equal to that in G_9 (resp., G_{10}).

Suppose $\mathcal{A}_{\mathrm{hy}}$ asks for the decryption of $c = (c_0, c_1, c_2, c_3)$. Let $\overline{k} = \mathsf{Dec}(\mathsf{sk}, c_0)$. Consider the following cases while assuming the event $\neg\mathsf{bad}$ occurs:

1. $\underline{c_0 = c_0^*}$: The $\textsc{Dec}_a^{\mathrm{hy}}$ oracle in G_9, G_{10}, G_{11} and G_{12} returns $(\perp, \perp)$. At the same time, no queries are made to H, H' and H'' at this stage (in particular, no query on $\overline{k}^*$).

2. $\underline{c_0 \neq c_0^* \text{ and } \overline{k} = \perp}$: The oracle $\textsc{Dec}_a^{\mathrm{hy}}$ in G_9 and G_{10} returns $(\perp, \perp)$. Since the event $\neg\mathsf{bad}$ (and hence, $\neg\mathsf{bad}_2$) happens, the oracle $\textsc{Dec}_a^{\mathrm{hy}}$ in G_{11} and G_{12} returns $(\perp, \perp)$ as well, as there does not exist a root $\overline{k}' \in S$ such that $\mathsf{Enc}(\mathsf{pk}, \overline{k}') = c_0$.

   No queries are made to H and H' in this case. On the other hand, a single *classical* query is made to H'' on a uniformly random value in $\mathcal{M}$ in games G_9 – G_{12}. Hence in particular, the probability of the query being $\overline{k}^*$ is equal in G_{10} and G_{12}.

3. $\underline{c_0 \neq c_0^*, \overline{k} \neq \perp \text{ and } H''(\overline{k}) \neq c_3}$: $\textsc{Dec}_a^{\mathrm{hy}}$ in G_9 and G_{10} returns $(\perp, \perp)$. Since the event $\neg\mathsf{bad}$ (and hence, $\neg\mathsf{bad}_1$) happens, $\textsc{Dec}_a^{\mathrm{hy}}$ in G_{11} and G_{12} returns $(\perp, \perp)$ as well, as there does not exist a root $\overline{k}' \in S$ such that $\mathsf{Enc}(\mathsf{pk}, \overline{k}') = c_0$; otherwise, from the rigidity of PKE (see Definition 4), we have $\mathsf{Enc}(\mathsf{pk}, \overline{k}) = c_0 = \mathsf{Enc}(\mathsf{pk}, \overline{k}')$ with $\overline{k} \neq \overline{k}'$ (since $H''(\overline{k}) \neq c_3 = H''(\overline{k}')$), contradicting the event $\neg\mathsf{bad}_1$ happening.

   No queries are made to H and H' in this case. In games G_9, G_{10}, a *classical* query is made to H'' on $\overline{k}$, to do the check "$(H''(\overline{k}) = c_3)$". As PKE is rigid, we have $\mathsf{Enc}(\mathsf{pk}, \overline{k}) = c_0 \neq c_0^*$. Since PKE is also deterministic, it must be the case that $\overline{k} \neq \overline{k}^*$. In G_{11} and G_{12}, as there does not exist a root $\overline{k}' \in S$ such that $\mathsf{Enc}(\mathsf{pk}, \overline{k}') = c_0$, we make a *classical* H''-query on a uniformly random value from $\mathcal{M}$. This step essentially keeps the H''-oracle calls "in sync" across both decryption oracles. Now it is not hard to see that if the j-th query to oracle H'' – where $j \leftarrow\!\!\$ \{1, \ldots, q_{H''}\}$ was sampled at the beginning of G_{10} and G_{12} – is at this stage, namely when $\mathcal{A}_{\mathrm{hy}}$ made this particular decryption query, then the probability of the measurement outcome w.r.t. this *classical* H''-query being $\bar{k}^*$ is 0 in G_{10} and $1/|\mathcal{M}|$ in G_{12}.

4. $\underline{c_0 \neq c_0^*, \bar{k} \neq \perp, H''(\bar{k}) = c_3 \text{ and } G(c_1, H'(\bar{k})) \neq c_2}$: $\textsc{Dec}_a^{\mathrm{hy}}$ in G_9 and G_{10} returns $(\perp, \perp)$. $\textsc{Dec}_a^{\mathrm{hy}}$ in G_{11} and G_{12} also returns $(\perp, \perp)$, as now we have $\bar{k} \in S$ (since $H''(\bar{k}) - c_3 = 0$) such that $\mathsf{Enc}(\mathsf{pk}, \bar{k}) = c_0$, which follows from PKE's rigidity. At the same time, as the event $\neg\mathsf{bad}$ (and hence, $\neg\mathsf{bad}_1$) happens, there must not exist a *different* root $\bar{k}' \in S$ such that $\mathsf{Enc}(\mathsf{pk}, \bar{k}') = c_0$. Since the G-check fails w.r.t. $\bar{k}$ in this new decryption oracle as well, i.e., $G(c_1, H'(\bar{k})) \neq c_2$, $(\perp, \perp)$ is returned.

   No query is made to H in this case. But a *classical* query is made to H' and H'' on the value $\bar{k}$ at this stage in G_9 – G_{12}. Thus, all oracles call are "in sync" across both versions of $\textsc{Dec}_a^{\mathrm{hy}}$, and the probability of the measurement outcome w.r.t. this *classical* $(H \times H')$-query (resp., H''-query) in G_9 and G_{11} (resp., G_{10} and G_{12}) being $\bar{k}^*$ is 0.

5. $\underline{c_0 \neq c_0^*, \bar{k} \neq \perp, H''(\bar{k}) = c_3 \text{ and } G(c_1, H'(\bar{k})) = c_2}$: $\textsc{Dec}_a^{\mathrm{hy}}$ in G_9 and G_{10} returns $(\bar{k}, \mathsf{Dec}^{\mathrm{dem}}(H(\bar{k}), c_1))$. $\textsc{Dec}_a^{\mathrm{hy}}$ in G_{11} and G_{12} also returns the pair $(\bar{k}, \mathsf{Dec}^{\mathrm{dem}}(H(\bar{k}), c_1))$ following a similar analysis above, the only difference being that now the (sole) root in S, namely $\bar{k}$, also satisfies the G-check: $G(c_1, H'(\bar{k})) = c_2$.

   In this case, a *classical* query is made to H, H' and H'' on $\bar{k}$ in G_9 – G_{12}. Again, all oracles call are "in sync" across both versions of $\textsc{Dec}_a^{\mathrm{hy}}$, and the probability of the measurement outcome w.r.t. any of the two *classical* $(H \times H')$-queries, corresponding to the $H(\bar{k})$ and $H'(\bar{k})$ calls respectively, in G_9 and G_{11} (resp., the single H''-query, corresponding to the $H''(\bar{k})$ call, in G_{10} and G_{12}) being k^* is 0.

Thus, we finally have that $\Pr[\mathsf{G}_9 = 1 | \neg\mathsf{bad}] \leq \Pr[\mathsf{G}_{11} = 1 | \neg\mathsf{bad}]$ and $\Pr[\mathsf{G}_{10} = 1 | \neg\mathsf{bad}] \leq \Pr[\mathsf{G}_{12} = 1 | \neg\mathsf{bad}]$. At the same time, it is not hard to see that the probability $\Pr[\neg\mathsf{bad}]$ (and hence, $\Pr[\mathsf{bad}]$) should be the same in games G_9 and G_{11} (resp., G_{10} and G_{12}). This is because, the event $\neg\mathsf{bad}$ depends on $\mathcal{A}_{\mathrm{hy}}$'s queries to the $\textsc{Dec}_a^{\mathrm{hy}}$ oracle. In the above analysis, since we showed that – assuming the event $\neg\mathsf{bad}$ occurs – the $\textsc{Dec}_a^{\mathrm{hy}}$ oracles have the same input-output behavior in $\mathsf{G}_9 - \mathsf{G}_{12}$, $\mathcal{A}_{\text{hy}}$'s view (and hence, execution) in G_9 and G_{11} (resp., G_{10} and G_{12}) should be *identical* until the first $\textsc{Dec}_a^{\text{hy}}$-query that violates $\neg\mathsf{bad}$; this means the probability that a given $\textsc{Dec}_a^{\text{hy}}$-query made by $\mathcal{A}_{\text{hy}}$ satisfies the event $\neg\mathsf{bad}$ remains the same in G_9 and G_{11} (resp., G_{10} and G_{12}) while conditioning on the event that all of $\mathcal{A}_{\text{hy}}$'s previous $\textsc{Dec}_a^{\text{hy}}$-queries are consistent with $\neg\mathsf{bad}$. So we have the following:

$$\Pr[\mathsf{G}_9 = 1] \leq \Pr[\mathsf{bad}] + \Pr[\neg\mathsf{bad} \wedge \mathsf{G}_{11} = 1] \leq \Pr[\mathsf{bad}] + \Pr[\mathsf{G}_{11} = 1],$$
$$\Pr[\mathsf{G}_{10} = 1] \leq \Pr[\mathsf{bad}] + \Pr[\neg\mathsf{bad} \wedge \mathsf{G}_{12} = 1] \leq \Pr[\mathsf{bad}] + \Pr[\mathsf{G}_{12} = 1].$$

Since we assumed that PKE is perfectly correct, we have $\Pr[\mathsf{bad}] = \Pr[\mathsf{bad}_1] = \Pr[\mathsf{bad}_2] = 0$.[9]

Finally, since the $\textsc{Dec}_a^{\text{hy}}$ oracle in games G_{11} and G_{12} does not use the secret key sk to decrypt any ciphertext, we can bound the success probability of $\mathcal{A}_{\text{hy}}$ in G_{11} and G_{12} via reductions to the OW-CPA security of PKE. Let $\mathcal{A}_{\text{pke}}$ (resp., $\mathcal{A}'_{\text{pke}}$) be an OW-CPA adversary against PKE that, given an input (pk, c_0^*), works as follows:

- Generates $k^* \leftarrow\$ \mathcal{K}$ and $\mu^* \leftarrow\$ \mathcal{M}$. Note that the uniform message $\overline{k}^*$ is generated implicitly by the OW-CPA challenger (along with the public key pk) such that $\mathsf{Enc}(\mathsf{pk}, \overline{k}^*) = c_0^*$.
- Uses a $2q_G$-wise independent function, $2q_H$-wise independent function, $2q_{H'}$-wise independent function and $2q_{H''}$-wise independent polynomial to simulate the quantum random oracles G, H, H' and H'' respectively, as noted in Lemma 1.
- Selects $i \leftarrow\$ \{1, \ldots, q_H + q_{H'}\}$ (resp., $j \leftarrow\$ \{1, \ldots, q_{H''}\}$).
- Until the i-th (resp., j-th) query to the oracle $H \times H'$ (resp., H'') is made, does the following:
    - Runs $\mathcal{A}_{\text{hy}}^{G,H,H',H'',\textsc{Dec}_{\perp}^{\text{hy}}}(\mathsf{pk})$ by answering the quantum random oracle queries and classical decryption queries as in G_{11} (resp., G_{12}), and finally obtains (m_0, m_1).

      (Note that the OW-CPA adversaries $\mathcal{A}_{\text{pke}}$ and $\mathcal{A}'_{\text{pke}}$ can simulate $\textsc{Dec}_a^{\text{hy}}$ without possessing the secret key sk.)

[9] The reason we went into the trouble of defining bad_1 and bad_2 events in the first place – despite PKE being perfectly correct – will be made clear in Subsection 6.3.1 where we discuss extending our security proof to the case where PKE exhibits decryption errors.

  - Samples a bit $b \leftarrow\$ \{0,1\}$ to compute $c_1^* = \mathsf{Enc}^{\mathrm{dem}}(k^*, m_b)$. Generates the rest of the ciphertext components as $c_2^* \leftarrow\$ \mathcal{F}$ and $c_3^* \leftarrow\$ \mathcal{M}$.
  - Runs $\mathcal{A}_{\mathrm{hy}}^{G,H,H',H'',\textsc{Dec}_{c^*}^{\mathrm{hy}}}(c^*)$, where $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$, by answering the quantum random oracle queries and classical decryption queries as in G_{11} (resp., G_{12}), and obtains a bit b'.
- Measures the argument $\overline{k}'$ of the i-th (resp., j-th) query to the oracle $H \times H'$ (resp., H'') and outputs $\overline{k}'$; if $\mathcal{A}_{\mathrm{hy}}$ makes less than i (resp., j) queries, output $\perp$.

From the above construction of adversaries $\mathcal{A}_{\mathrm{pke}}$ and $\mathcal{A}'_{\mathrm{pke}}$, it is easy to see that

$$\Pr[\mathsf{G}_{11} = 1] \leq \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}_{\mathrm{pke}}), \quad \Pr[\mathsf{G}_{12} = 1] \leq \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}'_{\mathrm{pke}}).$$

By combining all the above bounds w.r.t. the success probabilities of $\mathcal{A}_{\mathrm{hy}}$ in each of the games $\mathsf{G}_0 - \mathsf{G}_{12}$, we get

$$\mathbf{Adv}_{\mathsf{PKE}^{\mathrm{hy}}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}_{\mathrm{hy}}) \leq \mathbf{Adv}_{\mathsf{DEM}}^{\mathsf{otIND\text{-}CPA}}(\mathcal{A}_{\mathrm{dem}})$$
$$+ 2(q_H + q_{H'})\sqrt{\frac{2q_G}{\sqrt{|\mathcal{M}|}} + \frac{q_D}{|\mathcal{F}|} + \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}_{\mathrm{pke}}) + 2q_{H''}\sqrt{\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}'_{\mathrm{pke}})}}.$$

□

6.3.1 *Extension to "Non Perfectly-Correct"* PKE

In Theorem 19, we relied on the deterministic PKE scheme being perfectly correct. And as mentioned earlier, in the context of NIST's PQC standardization process, there are KEM candidates which use a base deterministic PKE scheme that is perfectly correct (and rigid): for example, Classic McEliece [21] and NTRU [23]. However, other important lattice-based NIST PQC candidates – such as Kyber [11], FrodoKEM [16] and Saber [62] – use a base (randomized) PKE scheme that exhibits decryption errors.[10]

[10] One way to *de-randomize* such PKE schemes is to apply the RO-based "T" transform of [4] (also see Figure 4.9). In fact, it was shown in [4, 41] that the T transform also confers rigidity. However, the transform includes a re-encryption check during decryption. And as argued at the start of this chapter, performing such re-encryption in a threshold manner could be a complicated process for lattice-based schemes.

Fortunately, by carefully going over the sequence of game-hops in our proof of Theorem 19, it is not hard to see how one can extend the analysis to PKE that may not be perfectly correct. To be more specific, in our analysis of games G_{11} and G_{12} above, we had the "bad" events bad_1 and bad_2 happening with zero probability because of PKE's perfect correctness.[11] In the case of PKE with decryption errors, one can bound the bad_1 and bad_2 probabilities with appropriate correctness properties which are specific to the concrete instantiation of PKE.

For example, in [27], we bounded the bad_1 probability using so-called *"δ-collision freeness"* of PKE; roughly speaking, a PKE scheme is said to be δ-collision free if the probability of the scheme having collisions – i.e., two messages encrypting to the same ciphertext – is bounded by δ where the probability is taken over generation of public/private key pairs.[12] Similarly, we also bounded the probability of bad_2 happening by the so-called "$\perp$-Aware *security*" of PKE; at a high level, the property captures the difficulty of an adversary, given the public key pk, to come up with a plaintext/ciphertext pair (m, c) such that $c = \mathsf{Enc}(\mathsf{pk}, m)$ but $\mathsf{Dec}(\mathsf{sk}, c) = \perp$. Then in the same paper [27], we gave an explicit construction of a deterministic rigid PKE which has an efficient distributed decryption procedure and whose one-wayness security is based on hardness of the *learning-with-rounding (LWR)* problem [101] (similar to Saber). Later, we rigorously analyzed the δ-collision freeness of our LWR-based PKE for concrete values of δ. Similarly, we established concrete $\perp$-Aware security of PKE by relying on the hardness of a novel lattice-based problem which we called the *large-vector-problem (LVP)* (see [27] for a formal definition).

6.3.2 *Threshold Variant*

Assuming there are protocols Π_{KGen} and Π_{Dec} which implement the base PKE in a threshold manner, a threshold variant of our above Hybrid construction is immediate. We simply apply the base threshold decryption operation to c_0, keeping the result in a shared form. The parties then securely evaluate G, H' and H'' to perform the respective checks in $\mathsf{Dec}^{\mathsf{hy}}$

[11] We also relied on perfect correctness in the $\mathsf{G}_1 \rightarrow \mathsf{G}_2$ game-hop. At the same time, we could have easily used the notion of "δ-correctness" (see Definition 2) for the same.

[12] In fact, this is equivalent to the *"ϵ-injectivity"* property introduced in [51, 52] for deterministic (base) PKE schemes in the context of obtaining tighter security proofs w.r.t. FO transforms in the QROM. At a high level, a deterministic PKE scheme is said to be ϵ-injective if the probability of the corresponding encryption function *not* being injective is bounded by ϵ where the probability is again taken over generation of public/private key pairs.

(see Figure 6.2). More concretely, by instantiating G, H' and H'' with MPC-friendly sponge-like hash functions such as Rescue (see Subsection 6.1.2 above), our distributed decryption operation for Hybrid-based $\mathsf{PKE}^{\mathrm{hy}}$ construction would consist of the following steps w.r.t. an input ciphertext $c = (c_0, c_1, c_2, c_3)$:

1. Absorb c_1 into "sponge" G in the clear.[13]
2. Apply Π_{Dec} for a distributed decryption of c_0, keeping the result $\bar{k}$ in shared form.
3. Securely absorb these shares of $\bar{k}$ into sponge H''.
4. Securely evaluate the squeezing of H'' to obtain c'_3 in the clear.
5. Reject the ciphertext if $c_3 \neq c'_3$.
6. Securely absorb the shares of $\bar{k}$ into sponge H'.
7. Securely evaluate squeezing of H', keeping the output in shared form.
8. Securely absorb the shares of $H'(\bar{k})$ into G.
9. Securely evaluate the squeezing of G to obtain c'_2 in the clear.
10. Reject the ciphertext if $c_2 \neq c'_2$.
11. Open $\bar{k}$ to all players.
12. Compute $k = H(\bar{k})$ in the clear
13. Compute $m = \mathsf{Dec}^{\mathrm{dem}}(k, c_1)$ in the clear and output it.

6.4 SUMMARY

In this chapter, we presented a new variant of the KEM-DEM framework, closely related to Tag-KEMs, which can be used to construct hybrid PKE schemes with an efficient distributed decryption procedure. Moreover, our generic framework – called the "Hybrid" transform – provably maintains IND-CCA security of the overall threshold implementation in a post-quantum setting (i.e., in the QROM).

[13] Since we are using a sponge-like function for G such as Rescue [98], or even SHA-3, we can insert the first c_1 argument for G during a distributed decryption in the clear as the ciphertext (c_0, c_1, c_2, c_3) is supposed to be public.

We also briefly discussed the potential applicability of our Hybrid framework to certain schemes in the NIST PQC standardization process, such as Classic McEliece and NTRU. Furthermore, given NIST's recent interest to standardize threshold cryptographic primitives, we hope our results inspire further research into constructing more efficient – and post-quantum secure – hybrid threshold public-key encryption.

7 CONCLUSIONS

In this thesis, we explored methods to enhance post-quantum secure public-key encryption schemes in ways which are agnostic to the underlying hardness assumptions. We focused particularly on PKE schemes that are a part of NIST's PQC standardization process. Concretely, this thesis considered the following enhancements:

IND-CCA Security Enhancements (Chapter 3). Most NIST PQC candidates for PKE employ variants of the Fujisaki-Okamoto (FO) transformation [1–4] to enhance their security to achieve the traditional notion of IND-CCA security. However, as we argued in this thesis, certain important NIST PQC schemes diverge from the standard FO transforms in ways which invalidate their concrete IND-CCA security claims in a post-quantum setting (i.e., in the QROM).

More specifically, we made a case study of two such schemes: namely, the current NIST PQC standard Kyber [11], and the third-round alternate candidate FrodoKEM [16] which is currently recommended by the German federal agency BSI. We re-examined the FO-variants used in these two schemes, and by focusing on the differences between these variants and the standard FO transforms, we identified gaps in their initial IND-CCA security claims. Following our observations, we re-established the concrete IND-CCA security of Kyber and FrodoKEM in the QROM by tailoring our analysis to handle the above differences in a rigorous manner.

Anonymity and Robust Enhancements (Chapters 4, 5). Given that the NIST PQC standards are intended to be widely used for decades to come, we argued for a *broader* analysis of these schemes with respect to security properties that go beyond the target notions set by NIST – especially in view of modern cryptographic applications that require such properties. Focusing on the PKE primitive, we shortlisted two such "beyond IND-CCA" properties in this thesis: namely, *anonymity* [12] and *robustness* [13]. We first presented a generic analysis of anonymity and robustness for PKE schemes built using the KEM-DEM paradigm, since this paradigm is used by most NIST candidates. Also as noted above, most PKE candidates in the NIST PQC process use underlying KEMs that are constructed from variants of the standard FO transforms. In this thesis, we analyzed one such standard transform called $\mathsf{FO}^{\not\perp}$ [4] with respect to its anonymity and robustness enhancing properties in the QROM.

We then studied the applicability of our above generic analysis on $\mathsf{FO}^{\not\perp}$-based KEMs to three specific NIST PQC KEMs: namely, the current standard Kyber [11], the third-round alternate candidate FrodoKEM [16], and the fourth-round candidate – and BSI-recommended – Classic McEliece [21]. For Classic McEliece, we showed that the scheme does not lead to robust KEM-DEM hybrid PKE schemes using a concrete "attack"; this also meant that our generic analysis cannot be extended to Classic McEliece to prove its post-quantum anonymity. Fortunately, we were able to show that FrodoKEM and Kyber do result in anonymous and robust hybrid PKE schemes in the post-quantum setting. For FrodoKEM, we adapted our above QROM analysis of $\mathsf{FO}^{\not\perp}$ to the specific FO-variant used by the NIST scheme. For the NIST standard Kyber, we adapted our techniques that were used to establish its concrete IND-CCA security in the QROM above, in conjunction with other recent techniques from the literature – namely, the so-called "*strong pseudorandomness*" framework [60].

Threshold Enhancements (Chapter 6). NIST's PQC standardization process currently only considers the primitives of PKE and digital signatures with basic functionalities. In view of NIST's recent plans to also standardize more advanced *threshold* schemes for (potentially post-quantum) cryptographic primitives, we explored ways to enhance the decryption functionality of quantum-resistant PKE schemes to a *distributed* setting. First, we identified issues with the generic design paradigm used by most NIST PQC candidates for PKE – i.e., the "FO + KEM-DEM" paradigm – in the context of obtaining IND-CCA secure *and* efficient threshold schemes. We then presented an alternative to the above paradigm called the "Hybrid" framework which overcomes the above issues; namely, our framework can be used to generically construct PKE schemes that have an efficient distributed decryption functionality, and at the same time are provably IND-CCA secure in the QROM. We also briefly discussed the potential applicability of our Hybrid framework to certain *perfectly correct* NIST PQC schemes such as the fourth-round candidate Classic McEliece [21] and the third-round finalist NTRU [23]. Regarding PKE schemes that may not be perfectly correct, we discussed ways to extend our analysis – albeit in a non-generic manner.

Impact. A tangible real-world impact of this thesis can be seen in NIST's recent plans [18, 19] to essentially replace the FO-variant currently used by the new PQC standard Kyber [11] with one of the standard FO transforms.

As addressed by a representative of the Kyber team [20], this is in part because of our arguments above on how the differences between Kyber's variant of the FO transform and the standard FO transforms invalidate the initial QROM IND-CCA security claims made in Kyber's NIST specification document [11].

Regarding the impact of our results on applications that require properties "beyond IND-CCA" security such as anonymity, a recent line of works [102–104] have used Kyber in post-quantum instantiations of *password-less authentication* methods defined in the Fast IDentity Online (FIDO) standards – specifically related to users' account recovery. This is enabled by our work that established post-quantum anonymity of Kyber, which in-turn provides privacy guarantees via *unlinkability* of users' public keys in the view of FIDO-based servers.

Finally, coming to NIST's recent plans to standardize threshold cryptographic schemes [24], our work on the aforementioned "Hybrid" framework – specifically, its application in [27] to "thresholdize" the NIST PQC third-round finalist *Saber* [62] – has found some interest in the community [105, 106], especially in the context of standardizing quantum-resistant (threshold) *Fully Homomorphic Encryption (FHE)* schemes that are actively secure. To be more specific, it was argued in [105, 106] that a potentially simpler path to standardize such threshold FHE schemes is to standardize their corresponding building blocks, which include threshold PKE schemes. And it was suggested in [105, 106] that one could build upon our work on the "Hybrid" framework to standardize such a class of FHE building blocks in the post-quantum setting.

7.1 FUTURE WORK

In this thesis, we made significant progress towards generically enhancing quantum-resistant PKE schemes – especially in the context of NIST's PQC standardization process. But our work also gives rise to some interesting open problems, which we divide into the following three categories.

7.1.1 *Tighter Analyses in the QROM*

In our concrete QROM IND-CCA security analyses of Kyber and FrodoKEM in Chapter 3, we mainly relied on the so-called "One-Way to Hiding (OW2H) lemma" [36, 37]. However, tighter variants of the OW2H lemma have since been introduced in the literature, e.g., in [51, 52]. At the same time, applying these variants to the aforementioned NIST PQC KEMs is not so straightforward; it would require an analysis of the corresponding base PKE schemes in a non-generic manner (i.e., dependent on the underlying hardness assumptions) to check if the schemes satisfy the prerequisite properties for applying the above tighter OW2H variants. Nonetheless, doing such an analysis would lead to tighter IND-CCA security proofs for Kyber and FrodoKEM in the QROM. It would also be interesting to study these OW2H variants using a *formal verification* framework – similar in spirit to Unruh's work [56] – given the real-world importance of the above NIST schemes, and as an extension, their corresponding security analyses. Finally, our above observations also apply in the context of obtaining tighter proofs of "beyond IND-CCA" security, such as anonymity (or, ANO-CCA security), for Kyber, FrodoKEM, and other NIST candidates in the QROM.

7.1.2 *"Beyond Anonymity, Robustness and Threshold" Enhancements*

In this thesis, we looked at the NIST candidates for PKE through the lens of modern "beyond IND-CCA" security properties such as anonymity and robustness, and "beyond basic PKE" functionalities such as threshold decryption. It would be interesting to expand this study to other important properties and functionalities that are relevant for emerging cryptographic applications. An example of one such property/functionality that was quite recently considered in the literature is that of *multi-receiver PKE* [107]. Roughly speaking, such a scheme encrypts a message in a *single-shot* to multiple receivers' public keys, and can be more efficient when compared to encrypting the message separately to each receiver using a basic PKE scheme. Multi-receiver PKE schemes have applications in group-oriented end-to-end secure messaging. These schemes were recently analysed in a post-quantum setting in [108].

7.1.3 *Generic v/s Non-Generic Analyses*

Finally, note that most of our analyses in this thesis were generic and modular – i.e., they were not dependent on any particular post-quantum hardness assumption, and only made *black-box* use of underlying basic primitives (such as "weakly" secure base PKE schemes). The main reason for this, as highlighted in Chapter 1, is that such analyses enable designers of enhanced PKE schemes to focus on instantiating the basic primitives with appropriate hardness assumptions; this is, in general, a much easier task when compared to *directly* constructing advanced schemes from such assumptions. However at the same time, it is important not to lose sight of the advantages offered by potentially non-generic analyses/constructions. For example, in the context of our "Hybrid" framework in Chapter 6 for constructing IND-CCA secure PKE schemes with efficient distributed decryption, note that we make explicit use of a generic base *multi-party computation (MPC)* functionality. This in-turn leads to many rounds of communication among users during the threshold decryption procedure. Hence, it would be interesting to develop an alternative framework which can potentially use MPC in a non-generic way to reduce the communication complexity. To conclude, it is important to analyse the trade-offs between generic and non-generic solutions towards the goal of achieving enhanced public-key encryption schemes in a post-quantum setting.

BIBLIOGRAPHY

1. Fujisaki, E. & Okamoto, T. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *CRYPTO'99*. *LNCS* **1666** (ed Wiener, M. J.) 537 (1999).
2. Fujisaki, E. & Okamoto, T. Secure Integration of Asymmetric and Symmetric Encryption Schemes. *Journal of Cryptology* **26**, 80 (2013).
3. Dent, A. W. A Designer's Guide to KEMs. *9th IMA International Conference on Cryptography and Coding*. *LNCS* **2898** (ed Paterson, K. G.) 133 (2003).
4. Hofheinz, D., Hövelmanns, K. & Kiltz, E. A Modular Analysis of the Fujisaki-Okamoto Transformation. *TCC 2017, Part I*. *LNCS* **10677** (eds Kalai, Y. & Reyzin, L.) 341 (2017).
5. Jiang, H., Zhang, Z., Chen, L., Wang, H. & Ma, Z. IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited. *CRYPTO 2018, Part III*. *LNCS* **10993** (eds Shacham, H. & Boldyreva, A.) 96 (2018).
6. Saito, T., Xagawa, K. & Yamakawa, T. Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model. *EUROCRYPT 2018, Part III*. *LNCS* **10822** (eds Nielsen, J. B. & Rijmen, V.) 520 (2018).
7. Don, J., Fehr, S., Majenz, C. & Schaffner, C. Online-Extractability in the Quantum Random-Oracle Model. *EUROCRYPT 2022, Part III*. *LNCS* **13277** (eds Dunkelman, O. & Dziembowski, S.) 677 (2022).
8. Hövelmanns, K., Hülsing, A. & Majenz, C. Failing Gracefully: Decryption Failures and the Fujisaki-Okamoto Transform. *ASIACRYPT 2022, Part IV*. *LNCS* **13794** (eds Agrawal, S. & Lin, D.) 414 (2022).
9. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C. & Zhandry, M. Random Oracles in a Quantum World. *ASIACRYPT 2011*. *LNCS* **7073** (eds Lee, D. H. & Wang, X.) 41 (2011).
10. Bellare, M. & Rogaway, P. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. *ACM CCS 93* (eds Denning, D. E., Pyle, R., Ganesan, R., Sandhu, R. S. & Ashby, V.) 62 (1993).

11. Avanzi, R., Bos, J., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G. & Stehlé, D. CRYSTALS-Kyber: NIST Round 3 Submission, Algorithm Specifications And Supporting Documentation (v3.02) (2021).

12. Bellare, M., Boldyreva, A., Desai, A. & Pointcheval, D. Key-Privacy in Public-Key Encryption. *ASIACRYPT 2001. LNCS* **2248** (ed Boyd, C.) 566 (2001).

13. Abdalla, M., Bellare, M. & Neven, G. Robust Encryption. *TCC 2010. LNCS* **5978** (ed Micciancio, D.) 480 (2010).

14. Camenisch, J. & Lysyanskaya, A. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. *EUROCRYPT 2001. LNCS* **2045** (ed Pfitzmann, B.) 93 (2001).

15. Cramer, R. & Shoup, V. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM Journal on Computing* **33**, 167 (2003).

16. Alkim, E., Bos, J. W., Ducas, L., Longa, P., Mironov, I., Naehrig, M., Nikolaenko, V., Peikert, C., Raghunathan, A. & Stebila, D. FrodoKEM: NIST round 3 submission (2021).

17. BSI Technical Guideline - Cryptographic Mechanisms: Recommendations and Key Lengths [BSI TR-02102-1]. *German Federal Office for Information Security, BSI* (2023).

18. Moody, D. [NIST PQC Forum] Subject: Discussion about Kyber's tweaked FO transform. `https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/WFRDl8DqYQ4/m/UXSO-ElSAwAJ`.

19. NIST. Module-Lattice-based Key-Encapsulation Mechanism Standard. *US Department of Commerce, NIST*. Federal Information Processing Standards Publications (FIPS PUBS) 203, Initial Public Draft (2023).

20. Schwabe, P. [NIST PQC Forum] Subject: Kyber decisions, part 2: FO transform. `https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/C0D3W1KoINY/m/99kIvydoAwAJ`.

21. Albrecht, M. R., Bernstein, D. J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R., Paterson, K. G., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Tjhai, C. J., Tomlinson, M. & Wang, W. Classic McEliece: NIST round 4 submission (2022).

22. Abe, M., Gennaro, R. & Kurosawa, K. Tag-KEM/DEM: A New Framework for Hybrid Encryption. *Journal of Cryptology* **21**, 97 (2008).

23. Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Saito, T., Schanck, J. M., Schwabe, P., Whyte, W., Xagawa, K., Yamakawa, T. & Zhang, Z. NTRU: NIST round 3 submission (2021).

24. Brandão, L. T. A. N. & Peralta, R. NIST First Call for Multi-Party Threshold Schemes (Initial Public Draft). *US Department of Commerce, NIST* (2023).

25. Grubbs, P., Maram, V. & Paterson, K. G. Anonymous, Robust Post-quantum Public Key Encryption. *EUROCRYPT 2022, Part III. LNCS* **13277** (eds Dunkelman, O. & Dziembowski, S.) 402 (2022).

26. Maram, V. & Xagawa, K. Post-quantum Anonymity of Kyber. *PKC 2023, Part I. LNCS* **13940** (eds Boldyreva, A. & Kolesnikov, V.) 3 (2023).

27. Cong, K., Cozzo, D., Maram, V. & Smart, N. P. Gladius: LWR Based Efficient Hybrid Public Key Encryption with Distributed Decryption. *ASIACRYPT 2021, Part IV. LNCS* **13093** (eds Tibouchi, M. & Wang, H.) 125 (2021).

28. Alamati, N. & Maram, V. Quantum CCA-Secure PKE, Revisited. *PKC 2024 (To Appear)* (2024).

29. Alamati, N., Maram, V. & Masny, D. Non-Observable Quantum Random Oracle Model. *Post-Quantum Cryptography - 14th International Workshop, PQCrypto 2023, Proceedings. Lecture Notes in Computer Science* **14154** (eds Johansson, T. & Smith-Tone, D.) 417 (2023).

30. Jauch, M. & Maram, V. Quantum Cryptanalysis of OTR and OPP: Attacks on Confidentiality, and Key-Recovery. *30th International Conference on Selected Areas in Cryptography, SAC 2023 (To Appear).* https://eprint.iacr.org/2023/1157 (2023).

31. Maram, V., Masny, D., Patranabis, S. & Raghuraman, S. On the Quantum Security of OCB. *IACR Trans. Symm. Cryptol.* **2022**, 379 (2022).

32. Maram, V. On the Security of NTS-KEM in the Quantum Random Oracle Model. *Code-Based Cryptography - 8th International Workshop, CBCrypto 2020, Revised Selected Papers. Lecture Notes in Computer Science* **12087** (eds Baldi, M., Persichetti, E. & Santini, P.) 1 (2020).

33. Liu-Zhang, C., Maram, V. & Maurer, U. On Broadcast in Generalized Network and Adversarial Models. *24th International Conference on Principles of Distributed Systems, OPODIS 2020. LIPIcs* **184** (eds Bramas, Q., Oshman, R. & Romano, P.) 25:1 (2020).

34. Nielsen, M. & Chuang, I. Quantum Computation and Quantum Information. *Cambridge University Press* (2000).

35. Zhandry, M. Secure Identity-Based Encryption in the Quantum Random Oracle Model. *CRYPTO 2012. LNCS* **7417** (eds Safavi-Naini, R. & Canetti, R.) 758 (2012).

36. Unruh, D. Revocable Quantum Timed-Release Encryption. *EUROCRYPT 2014. LNCS* **8441** (eds Nguyen, P. Q. & Oswald, E.) 129 (2014).

37. Ambainis, A., Hamburg, M. & Unruh, D. Quantum Security Proofs Using Semi-classical Oracles. *CRYPTO 2019, Part II. LNCS* **11693** (eds Boldyreva, A. & Micciancio, D.) 269 (2019).

38. Ambainis, A., Rosmanis, A. & Unruh, D. Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding. *55th FOCS*, 474 (2014).

39. Hülsing, A., Rijneveld, J. & Song, F. Mitigating Multi-target Attacks in Hash-Based Signatures. *PKC 2016, Part I. LNCS* **9614** (eds Cheng, C.-M., Chung, K.-M., Persiano, G. & Yang, B.-Y.) 387 (2016).

40. Zhandry, M. A Note on the Quantum Collision and Set Equality Problems. *Quantum Information and Computation* **15** (2015).

41. Bernstein, D. J. & Persichetti, E. Towards KEM Unification. `https://eprint.iacr.org/2018/526` (2018).

42. Bellare, M. & Rogaway, P. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. *EUROCRYPT 2006. LNCS* **4004** (ed Vaudenay, S.) 409 (2006).

43. Shoup, V. Sequences of games: a tool for taming complexity in security proofs. `https://eprint.iacr.org/2004/332` (2004).

44. Katz, J. & Lindell, Y. Introduction to Modern Cryptography, Second Edition. *Chapman and Hall/CRC* (2014).

45. Boneh, D. & Zhandry, M. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. *CRYPTO 2013, Part II. LNCS* **8043** (eds Canetti, R. & Garay, J. A.) 361 (2013).

46. Boneh, D. & Shoup, V. A Graduate Course in Applied Cryptography (version 0.6). `https://toc.cryptobook.us/book.pdf` (2023).

47. Jiang, H., Zhang, Z. & Ma, Z. Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model. *PKC 2019, Part II. LNCS* **11443** (eds Lin, D. & Sako, K.) 618 (2019).

48. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A. & Smith-Tone, D. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. *US Department of Commerce, NIST* (2022).

49. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *J. ACM* **56** (2009).

50. Targhi, E. E. & Unruh, D. Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms. *TCC 2016-B, Part II. LNCS* **9986** (eds Hirt, M. & Smith, A. D.) 192 (2016).

51. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A. & Persichetti, E. Tighter Proofs of CCA Security in the Quantum Random Oracle Model. *TCC 2019, Part II. LNCS* **11892** (eds Hofheinz, D. & Rosen, A.) 61 (2019).

52. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R. & Sun, S. Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security. *EUROCRYPT 2020, Part III. LNCS* **12107** (eds Canteaut, A. & Ishai, Y.) 703 (2020).

53. Ding, X., Esgin, M. F., Sakzad, A. & Steinfeld, R. An Injectivity Analysis of Crystals-Kyber and Implications on Quantum Security. *ACISP 22. LNCS* **13494** (eds Nguyen, K., Yang, G., Guo, F. & Susilo, W.) 332 (2022).

54. Langlois, A. & Stehlé, D. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.* **75**, 565 (2015).

55. Hövelmanns, K., Kiltz, E., Schäge, S. & Unruh, D. Generic Authenticated Key Exchange in the Quantum Random Oracle Model. *PKC 2020, Part II. LNCS* **12111** (eds Kiayias, A., Kohlweiss, M., Wallden, P. & Zikas, V.) 389 (2020).

56. Unruh, D. Post-Quantum Verification of Fujisaki-Okamoto. *ASIACRYPT 2020, Part I. LNCS* **12491** (eds Moriai, S. & Wang, H.) 321 (2020).

57. Zhandry, M. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. *CRYPTO 2019, Part II. LNCS* **11693** (eds Boldyreva, A. & Micciancio, D.) 239 (2019).

58. Katsumata, S., Kwiatkowski, K., Pintore, F. & Prest, T. Scalable Ciphertext Compression Techniques for Post-quantum KEMs and Their Applications. *ASIACRYPT 2020, Part I. LNCS* **12491** (eds Moriai, S. & Wang, H.) 289 (2020).
59. Ge, J., Shan, T. & Xue, R. Tighter QCCA-Secure Key Encapsulation Mechanism with Explicit Rejection in the Quantum Random Oracle Model. *CRYPTO 2023, Part V* **14085** (eds Handschuh, H. & Lysyanskaya, A.) 292 (2023).
60. Xagawa, K. Anonymity of NIST PQC Round 3 KEMs. *EUROCRYPT 2022, Part III. LNCS* **13277** (eds Dunkelman, O. & Dziembowski, S.) 551 (2022).
61. Kreuzer, K. Verification of the (1–δ)-Correctness Proof of CRYSTALS-KYBER with Number Theoretic Transform. `https://eprint.iacr.org/2023/027` (2023).
62. Basso, A., Mera, J. M. B., D'Anvers, J., Karmakar, A., Roy, S. S., Beirendonck, M. V. & Vercauteren, F. Saber: NIST round 3 submission (2021).
63. D'Anvers, J.-P., Karmakar, A., Roy, S. S. & Vercauteren, F. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM. *AFRICACRYPT 18. LNCS* **10831** (eds Joux, A., Nitaj, A. & Rachidi, T.) 282 (2018).
64. Bernstein, D. J. [NIST PQC Forum] Subject: Anonymity of KEMs in the QROM. `https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/8k3MhD_5stk/m/TWGKtuL4BgAJ`.
65. Chen, Z., Lu, X., Jia, D. & Li, B. IND-CCA Security of Kyber in the Quantum Random Oracle Model, Revisited. *Information Security and Cryptology - 18th International Conference, Inscrypt 2022, Beijing, December 11-13, 2022, Revised Selected Papers* (2022).
66. Barbosa, M. & Hülsing, A. The security of Kyber's FO-transform. `https://eprint.iacr.org/2023/755` (2023).
67. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E. & Virza, M. Zerocash: Decentralized Anonymous Payments from Bitcoin. *2014 IEEE Symposium on Security and Privacy*, 459 (2014).
68. Barth, A., Boneh, D. & Waters, B. Privacy in Encrypted Content Distribution Using Private Broadcast Encryption. *FC 2006. LNCS* **4107** (eds Di Crescenzo, G. & Rubin, A.) 52 (2006).

69. Libert, B., Paterson, K. G. & Quaglia, E. A. Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model. *PKC 2012*. *LNCS* **7293** (eds Fischlin, M., Buchmann, J. & Manulis, M.) 206 (2012).

70. Sako, K. An Auction Protocol Which Hides Bids of Losers. *PKC 2000*. *LNCS* **1751** (eds Imai, H. & Zheng, Y.) 422 (2000).

71. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P. & Shi, H. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. *CRYPTO 2005*. *LNCS* **3621** (ed Shoup, V.) 205 (2005).

72. Farshim, P., Libert, B., Paterson, K. G. & Quaglia, E. A. Robust Encryption, Revisited. *PKC 2013*. *LNCS* **7778** (eds Kurosawa, K. & Hanaoka, G.) 352 (2013).

73. Farshim, P., Orlandi, C. & Roşie, R. Security of Symmetric Primitives under Incorrect Usage of Keys. *IACR Trans. Symm. Cryptol.* **2017**, 449 (2017).

74. Grubbs, P., Lu, J. & Ristenpart, T. Message Franking via Committing Authenticated Encryption. *CRYPTO 2017, Part III*. *LNCS* **10403** (eds Katz, J. & Shacham, H.) 66 (2017).

75. Dodis, Y., Grubbs, P., Ristenpart, T. & Woodage, J. Fast Message Franking: From Invisible Salamanders to Encryptment. *CRYPTO 2018, Part I*. *LNCS* **10991** (eds Shacham, H. & Boldyreva, A.) 155 (2018).

76. Len, J., Grubbs, P. & Ristenpart, T. Partitioning Oracle Attacks. *USENIX Security 2021* (eds Bailey, M. & Greenstadt, R.) 195 (2021).

77. Mohassel, P. A Closer Look at Anonymity and Robustness in Encryption Schemes. *ASIACRYPT 2010*. *LNCS* **6477** (ed Abe, M.) 501 (2010).

78. Melchor, C. A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bos, J., Deneuville, J., Dion, A., Gaborit, P., Lacan, J., Persichetti, E., Robert, J., Véron, P. & Zémor, G. HQC: NIST round 3 submission (2021).

79. Fujisaki, E. & Okamoto, T. How to Enhance the Security of Public-Key Encryption at Minimum Cost. *PKC'99*. *LNCS* **1560** (eds Imai, H. & Zheng, Y.) 53 (1999).

80. Hayashi, R. & Tanaka, K. PA in the Two-Key Setting and a Generic Conversion for Encryption with Anonymity. *ACISP 06*. *LNCS* **4058** (eds Batten, L. M. & Safavi-Naini, R.) 271 (2006).

81. Goldwasser, S., Micali, S. & Rivest, R. L. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing* **17**, 281 (1988).

82. McEliece, R. J. A public-key cryptosystem based on algebraic coding theory. *The Deep Space Network Progress Report 42-44*. `https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF`, 114 (1978).

83. Niederreiter, H. Knapsack-Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory* **15**, 159 (1986).

84. Lindner, R. & Peikert, C. Better Key Sizes (and Attacks) for LWE-Based Encryption. *CT-RSA 2011. LNCS* **6558** (ed Kiayias, A.) 319 (2011).

85. Liu, X. & Wang, M. QCCA-Secure Generic Key Encapsulation Mechanism with Tighter Security in the Quantum Random Oracle Model. *PKC 2021, Part I. LNCS* **12710** (ed Garay, J.) 3 (2021).

86. Adida, B. Helios: Web-based Open-Audit Voting. *USENIX Security 2008* (ed van Oorschot, P. C.) 335 (2008).

87. Clarkson, M. R., Chong, S. & Myers, A. C. Civitas: Toward a Secure Voting System. *2008 IEEE Symposium on Security and Privacy*, 354 (2008).

88. Kokoris-Kogias, E., Alp, E. C., Gasser, L., Jovanovic, P., Syta, E. & Ford, B. CALYPSO: Private Data Management for Decentralized Ledgers. *Proc. VLDB Endow.* **14**, 586 (2020).

89. Nikova, S., Rechberger, C. & Rijmen, V. Threshold Implementations Against Side-Channel Attacks and Glitches. *Information and Communications Security* (eds Ning, P., Qing, S. & Li, N.) 529 (2006).

90. Bendlin, R. & Damgård, I. Threshold Decryption and Zero-Knowledge Proofs for Lattice-Based Cryptosystems. *TCC 2010. LNCS* **5978** (ed Micciancio, D.) 201 (2010).

91. Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P. M. R. & Sahai, A. Threshold Cryptosystems from Threshold Fully Homomorphic Encryption. *CRYPTO 2018, Part I. LNCS* **10991** (eds Shacham, H. & Boldyreva, A.) 565 (2018).

92. Smart, N. P., Albrecht, M. R., Lindell, Y., Orsini, E., Osheter, V., Paterson, K. G. & Peer, G. LIMA: NIST round 1 submission (2018).

93. Kraitsberg, M., Lindell, Y., Osheter, V., Smart, N. P. & Talibi Alaoui, Y. Adding Distributed Decryption and Key Generation to a Ring-LWE Based CCA Encryption Scheme. *ACISP 19*. *LNCS* **11547** (eds Jang-Jaccard, J. & Guo, F.) 192 (2019).

94. Cramer, R., Damgård, I. & Nielsen, J. Secure Multiparty Computation and Secret Sharing. *Cambridge University Press* (2015).

95. Evans, D., Kolesnikov, V. & Rosulek, M. A Pragmatic Introduction to Secure Multi-Party Computation (2018).

96. Shoup, V. & Gennaro, R. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. *EUROCRYPT'98*. *LNCS* **1403** (ed Nyberg, K.) 1 (1998).

97. Shoup, V. & Gennaro, R. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. *Journal of Cryptology* **15**, 75 (2002).

98. Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S. & Szepieniec, A. Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. `https://eprint.iacr.org/2019/426` (2019).

99. Grassi, L., Kales, D., Khovratovich, D., Roy, A., Rechberger, C. & Schofnegger, M. Starkad and Poseidon: New Hash Functions for Zero Knowledge Proof Systems. `https://eprint.iacr.org/2019/458` (2019).

100. Bonte, C., Smart, N. P. & Tanguy, T. Thresholdizing HashEdDSA: MPC to the Rescue. `https://eprint.iacr.org/2020/214` (2020).

101. Banerjee, A., Peikert, C. & Rosen, A. Pseudorandom Functions and Lattices. *EUROCRYPT 2012*. *LNCS* **7237** (eds Pointcheval, D. & Johansson, T.) 719 (2012).

102. Pu, S., Thyagarajan, S. A., Döttling, N. & Hanzlik, L. Post Quantum Fuzzy Stealth Signatures and Applications. *ACM CCS 2023* (eds Meng, W., Jensen, C. D., Cremers, C. & Kirda, E.) 371 (2023).

103. Brendel, J., Clermont, S. & Fischlin, M. Post-Quantum Asynchronous Remote Key Generation for FIDO2 Account Recovery. `https://eprint.iacr.org/2023/1275` (2023).

104. Wilson, S. M. Post-Quantum Account Recovery for Passwordless Authentication. *University of Waterloo* (2023).

105. Polyakov, Y. FHE-Related Comments on NIST First Call for Multi-Party Threshold Schemes. `https://csrc.nist.gov/csrc/media/presentations/2023/mpts2023-day2-talk-fhe-comments/images-media/mpts2023-2a1-slides--yuriy--FHE-comments.pdf` (2023).

106. Badawi, A. A., Alexandru, A., Genise, N., Micciancio, D., Polyakov, Y., R.V., S. & Vaikuntanathan, V. Comments on NIST First Call for Multi-Party Threshold Schemes. `https://csrc.nist.gov/csrc/media/pubs/ir/8214/c/ipd/docs/nistir-8214c-ipd-public-feedback.pdf` (2023).

107. Kurosawa, K. Multi-recipient Public-Key Encryption with Shortened Ciphertext. *PKC 2002*. *LNCS* **2274** (eds Naccache, D. & Paillier, P.) 48 (2002).

108. Alwen, J., Hartmann, D., Kiltz, E., Mularczyk, M. & Schwabe, P. Post-Quantum Multi-Recipient Public Key Encryption. *ACM CCS 2023* (eds Meng, W., Jensen, C. D., Cremers, C. & Kirda, E.) 1108 (2023).