EPFL

Thèse n° 9762

Security in the Presence of Quantum Adversaries

Présentée le 25 octobre 2023

Faculté informatique et communications
Laboratoire de sécurité et de cryptographie
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Khashayar BAROOTI

Acceptée sur proposition du jury

Prof. O. N. A. Svensson, président du jury
Prof. S. Vaudenay, directeur de thèse
Prof. S. Fehr, rapporteur
Prof. C. Majenz, rapporteur
Prof. A. Chiesa, rapporteur

École polytechnique fédérale de Lausanne

2023

To all the brave women of my country…
Women, Life, Freedom

Abstract

With the looming threat of large-scale quantum computers, a fair portion of recent cryptographic research has focused on examining cryptographic primitives from the perspective of a quantum adversary. Shor's 1994 result revealed that quantum computers can efficiently solve the discrete logarithm and factorization problems, the foundation of public-key cryptography's hardness assumptions. As a response, the field of post-quantum cryptography has emerged, aiming to redesign classical cryptographic primitives to maintain security against quantum adversaries.

Conversely, quantum computation presents new opportunities for cryptographic design. It may be possible to construct cryptographic primitives designed specifically for quantum parties, relying on weaker assumptions compared to classical cryptography or even eliminating the need for any computational assumptions altogether. This has opened up exciting possibilities for exploring quantum-enhanced cryptographic schemes.

In this thesis, we delve into both aspects: classical cryptography guaranteeing security against quantum adversaries and the potential opportunities presented by cryptographic primitives harnessing quantum computation.

Throughout the first part of the thesis, we focus on post-quantum signature schemes. We examine signature schemes within the Minicrypt realm, built on the MPC-in-the-head framework and symmetric-key primitives. The results we present demonstrate that the security level of these schemes is influenced by the multiplication complexity of the underlying symmetric-key cipher. We specifically analyse the `PICNIC` signature scheme, instantiated with the `LowMC` block cipher family, and establish the importance of maintaining a sufficient round complexity in the block cipher to ensure security.

The second part of the thesis focuses on cryptographic primitives specifically designed for parties utilizing quantum computation. We thoroughly explore the concept of public-key encryption (PKE) in the quantum domain and tackle the question of whether it is feasible to construct PKE schemes using assumptions weaker than those required in classical settings. We demonstrate that it is indeed possible to construct a quantum PKE scheme by relying solely on the existence of one-way functions or potentially weaker assumptions.

Additionally, we explore the utilization of self-testing techniques from quantum mechanics in the field of learning theory. We focus on the challenge of constructing classifiers that exhibit robustness against test examples drawn from arbitrary distributions, including adversarially chosen examples. We showcase the application of self-testing techniques to offer cryptographic guarantees for such tasks within a quantum learning model.

Abstract

Keywords: Cryptography, Post-Quantum Cryptography, Digital Signature Schemes, MPC-in-the-head, Block-Ciphers, Public-Key Encryption, Chosen-Ciphertext Security, Learning Theory, Delegation of Quantum Computation

Résumé

La sourde menace posée par les ordinateurs quantiques a conduit une partie des cryptologues à réexaminer les primitives cryptographiques du point de vue d'un adversaire doté d'une force de calcul quantique. Le résultat de Shor de 1994 a révélé que ces machines seraient capables de résoudre efficacement les problèmes du logarithme discret et de la factorisation. En réaction, la communauté cryptographique s'est employée à construire des alternatives aux primitives dites "classiques" afin de garantir la sécurité dans un monde quantique. Ce nouveau domaine de recherche a pris le nom de cryptographie post-quantique.
Cependant, la puissance du calcul quantique offre également de nouvelles opportunités en termes de conception cryptographique. Par exemple, il peut être possible de construire des primitives quantiques qui s'appuient sur des hypothèses plus faibles comparé à leur équivalent classique, ou mieux, sur aucune hypothèse calculatoire du tout. L'informatique quantique ouvre donc ainsi la voie à de nouvelles avancées conséquentes dans la construction de systèmes cryptographiques.
Dans cette thèse, nous explorons ces deux aspects ; c'est-à-dire la cryptographie post-quantique qui a pour but de construire des algorithmes classiques qui résistent aux ordinateurs quantiques, et la cryptographie quantique proprement dite, qui se concentre sur la construction de primitives opérant directement sur un ordinateur quantique.
Dans la première partie de la thèse, nous nous concentrons sur les systèmes de signature post-quantiques. Plus particulièrement, nous examinons des signatures numériques construites à partir de la technique dite "MPC-dans-la-tête" et de primitives symétriques. Les résultats obtenus démontrent que le niveau de sécurité de ces constructions est influencé par la complexité des multiplications dans le système de chiffrement symétrique sous-jacent. Plus spécifiquement, nous analysons `PICNIC` instancié avec le chiffrement par bloc `LowMC` et montrons l'importance de maintenir un nombre de répétitions élevé dans le système de chiffrement par bloc.
Dans la seconde partie de la thèse, nous nous concentrons sur des primitives quantiques, c'est-à-dire destinées à être exécutées sur des ordinateurs quantiques. Nous explorons en détail le concept de chiffrement à clé publique dans un monde quantique et nous abordons la question de savoir s'il est possible de construire un tel système en se basant sur des hypothèses plus faibles que dans un monde classique.
En outre, nous explorons l'application de techniques d'autocontrôle issues de la mécanique quantique au domaine la théorie de l'apprentissage. Nous nous concentrons sur le défi de construire des classifieurs robustes contre des échantillons test tirés de distributions arbi-

traires (y compris des échantillons choisis par un adversaire). Nous démontrons comment les techniques d'autocontrôle peuvent fournir des garanties pour cette tâche dans un modèle d'apprentissage quantique.

Mots-clés : Cryptographie, Cryptographie post-quantique, Signature numérique, Chiffrement par bloc, Chiffrement à clé publique, Sécurité contre les attaques à texte chiffré choisi, Théorie de l'apprentissage, Délégation du calcul quantique

Acknowledgements

While I was doing my bachelor's studies, one of my professors jokingly told us: *"to be successful in academia, you should either be a great researcher or find great researchers and stick to them!"* Now, five years later, I find this jest remarkably true. Although not very successful, I am a living example of the second category in that quip. Thus, I take a moment to express my gratitude to all of those who made me look smarter than I actually am in this chapter.

First, I would like to thank my advisor Professor Serge Vaudenay. I started my Ph.D. journey at the age of 21, with a love for mathematics and an abundance of unfounded confidence. A Single encounter with Serge was all it took for me to regain my humility. Over five years have passed since that initial meeting, yet each discussion with him continues to evoke the same sense. I have learned so much from him during my five years in the lab that can not be described. A rare luxury that Serge provided me with from the beginning of my Ph.D. was having the freedom to choose the problems I would like to work on. I can not express how appreciative I am of the independence I gained from this. Serge, I am deeply appreciative of the trust you placed in me with this opportunity, and thank you for all you have taught me.
I would also like to express my gratitude to the members of my examination committee. My heartfelt thanks go to Professor Ola Svenson, who chaired my defense as the examination president, and to Professor Alessandro Chiesa, who served as my internal examiner. Furthermore, I extend my appreciation to Professor Christian Majenz and Professor Serge Fehr for graciously agreeing to be my external examiners and for their dedication in traveling to Switzerland to attend the examination in person. It is often said that examiners do not delve deeply into the technical content of the thesis; however, I was pleasantly surprised by the thoroughness with which my examiners inspected my work and provided valuable insights. I cannot overemphasize my gratitude for this.

Next, my heartfelt thanks go to Martine Corval, whose invaluable assistance made every administrative aspect of this journey seamless for all of us. I can hardly imagine the multitude of errors I would have encountered without her unwavering support. Martine, your patience in handling my occasionally unconventional requests, the numerous French translations you helped us with, and the warm smile that greeted us each morning when we arrived are a small list of all I would be forever grateful for. Your retirement and absence from the lab will be keenly felt. I would also like to express my gratitude to Sylvie Buchard who has made us not feel the absence of Martine from the lab since her retirement.

Acknowledgements

For five years, my desk in INF240 has been my constant haven. During this time, I've had the privilege of sharing this space with two brilliant individuals— Handan Kilinc, albeit for a shorter duration, and Andrea Caforio, with whom I have spent four remarkable years. I extend my heartfelt gratitude to both of them for the great discussions and lasting memories we've created.
Allow me to elaborate on my appreciation for Andrea, with whom I have shared an office for the majority of my journey at EPFL. Andi epitomizes the ideal office mate. There has always been a perfect blend of focused work and lighthearted banter. Andi, thank you for bearing my presence for four years! Thank you for all the music suggestions, the hidden corners of the internet culture you introduced me to, and the great memories. Also, thanks a lot for graduating before me, even though you started later than me —a friendly reminder that I am not as smart as I might think!
To not bore the reader, I will just briefly mention all the lab members. I would like to thank; Gwangbae, for all the times he locked me in my own office by removing the door handle; Laurane, for infusing much-needed social interactions into our lab; Abdullah, for sharing an abundance of Turkish memes with me; Fatih, for all the soul-crushing de-motivational speeches; Betül for all the outdoor activities she organized; Benedikt, for consistently handling the toughest tasks when TAing the cryptography course; Hailun, for treating us to homemade Chinese delicacies; Ritam, for the passionate political discussions; Daniel, ..., well for being the honorary third resident of INF240; Loïs for introducing us to the best city in the world, [1] and fête des Vendanges; and Boris, for his unwavering enthusiasm to join me in the line for the Kashmiri food truck.
The last lab member that I would like to talk about is Subhadeep. I had the absolute pleasure to collaborate with Subhadeep and it is not far-fetched to say this thesis would not have existed without him. Thank you for building my confidence as a researcher. Thank you for teaching me all I know about symmetric cryptography, and most importantly for the valuable lesson in stubbornness when it comes to research, to never give up on problems too soon.

During these years I have had the opportunity to work with some of the most brilliant people I could have wished for. So I would like to take a moment to express my appreciation. First, I would like to thank Grzegorz Głuch and Rüdiger Urbanke for the long meetings and discussions we had and the amazing journey of going from learning a new topic to conducting research on it. I would also like to express my gratitude to Alex B. Grilo, Or Sattath, Michael Walter, and Quoc-Huy Vu. I owe my personal favorite result to working with this amazing team.
Furthermore, I wish to thank Marc-Olivier Renou for providing me with an understanding of quantum information from a physical perspective and finally, Thomas Vidick, whom I have learned all I know about quantum computing from.
I would like to extend my heartfelt thanks to Giulio Malavolta for the amazing three months he hosted me at Max-Planck Institute and for everything I have learned from him during and after my time in Bochum. Additionally, I wish to express my appreciation to Ahmadreza, Phillip,

[1] Neuchâtel

and Behzad for their warm hospitality and for ensuring that I felt entirely at home during my visit to their team.

A great thing about doing a Ph.D. at EPFL is that the school does a fantastic job of fostering a strong sense of community among the students who have just joined the program. Thanks to this supportive environment, I've had the privilege of meeting some of my dearest friends over the course of these five years. Apart from the people whom I have already mentioned, I would like to thank Mahyar and Andreas, who were the best flatmates I could have asked for, Thanasis, Etienne, Xinrui, Atri, Baran, Ognjen, Arnout, Victor, and many other amazing people I have met during my time at EPFL. To this amazing group of people, I owe some of the most unforgettable moments of my life.
Another group of people whom I have my best memories of Lausanne with is the Navid, Omid, and Kamran or I like to call them the *Tabriz vs. Tehran* crew. I genuinely cannot imagine how boring these years would have been if I did not know them.

Whenever I thought about giving up my career as a researcher I imagined quitting my Ph.D. and launching a music career, until I crossed paths with the most talented musician I have ever known, my partner Lea. Thanks to her I had a well-needed reality check that I should stay committed to the things I am at least slightly talented in and leave music to people with genuine talent, like her. I thank her for making my experience in Switzerland so much more enjoyable than it was before I knew her and for always being there when I needed her support.
Where I come from, parents often wield a substantial influence over their children's life paths, educational choices, and career decisions. I count myself among the fortunate few who enjoyed complete agency over their life choices, knowing whatever path I chose, my parents would stand firmly behind me and believe in me. I would not have been courageous enough to make the pivotal decision I have if I did not have their support and for that, I am forever grateful. Words can not describe how lucky I feel to have them.

Lausanne, October 10, 2023

Contents

Introduction

In a *must-read* survey titled *"A Personal View of Average-Case Complexity"*, Russell Impagliazzo [Imp95] introduced the five realms of average-case complexity hardness:

1. Algorithmica: a realm where P = NP.
2. Heuristica: a realm where NP problems are hard in the worst case but easy on average.
3. Pessiland: a realm where NP problems are hard on average, but no one-way functions exist. Hard NP problems can be easily created, but not hard NP problems for which the solution is known.
4. Minicrypt: a realm where one-way functions exist, but public-key cryptography might not.
5. Cryptomania: a realm where public-key cryptography is possible.

The study of the latter two realms and their relationship forms the foundation of cryptographic research. Therefore, cryptographic research is often divided into two segments. The first segment is symmetric-key cryptography, which focuses on the study of cryptographic objects residing in Minicrypt. This sub-field is dedicated to investigating schemes such as commitments, symmetric-key encryption, message authentication codes, collision-resistant hash functions, and more. Hardness assumptions in symmetric cryptography are often based on heuristics rather than being derived from well-structured mathematical problems.

The second sub-field of cryptography focuses on public-key cryptography, which involves the study of objects within the realm of Cryptomania. Public-key encryption, trapdoor permutations, and secure multi-party computation are among the key objects of interest in this sub-field. Unlike symmetric cryptography, public-key cryptography relies heavily on mathematically structured problems and well-established, often number-theoretic, hardness assumptions. The most commonly employed hardness assumptions in public-key cryptography are the discrete logarithm problem and the factoring problem. A non-exhaustive list of cryptographic objects in both worlds and their relations is visualized in Figure 1.

However, a new threat emerged in the peaceful realms of cryptography with the advent of quantum computing. In 1994, Peter Shor's seminal work [Sho94] demonstrated that both

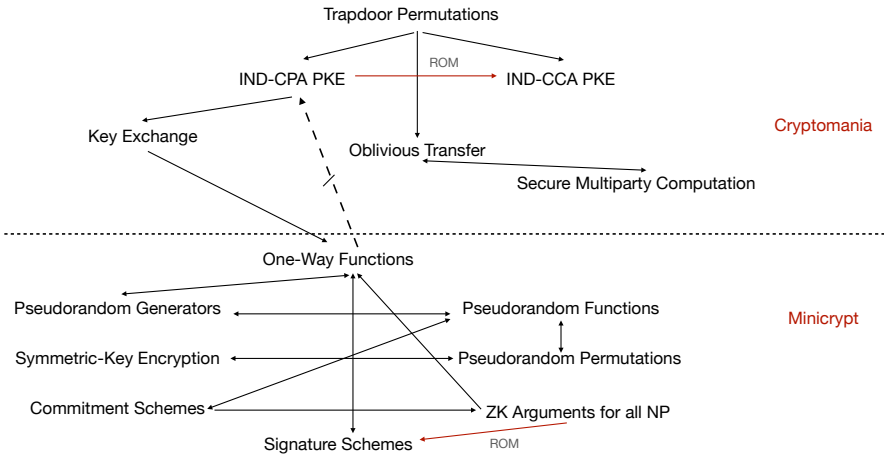


Figure 1: A Visualization of Cryptographic primitives and their relations. The arrows indicate which primitives can be built from which and the dotted lines indicate the existence of an oracle separation.

the discrete logarithm and factorization problems could be solved by quantum computers in polynomial time. As previously mentioned, the hardness of factorization and discrete logarithm served as the foundation for many cryptographic objects in Cryptomania. On the other hand, quantum computing also presented intriguing possibilities. The quantum key distribution protocol by Bennett and Brassard [BB84] and the quantum money protocol by Wiesner [Wie83] hinted at the potential for building cryptographic primitives from weaker or even no computational assumptions. This development raised a crucial question: "How will the achievement of quantum supremacy impact Impagliazzo's cryptographic realms?" In light of this question, we can approach the issue from two distinct perspectives.

Classical Cryptography, Secure Against Quantum Adversaries:

Examining classical cryptographic primitives through the lens of quantum computing has given rise to a new sub-field known as *post-quantum cryptography*. One of the primary focuses of this field has been replacing existing Cryptomania hardness assumptions with alternative problems that are believed to remain difficult for quantum computers to solve. Extensive research has been dedicated to constructing cryptographic primitives based on various new assumptions, such as the hardness of the learning with errors problem (LWE) [Reg05] and finding isogenies between elliptic curves [RS06].

It is important to stress that the pursuit of quantum-secure cryptography is not merely of theoretical interest. In fact, the National Institute of Standards and Technology (NIST) has held a standardization process for post-quantum cryptography [AAC+22]. The procedure primarily targeted two key primitives: key-encapsulation mechanisms (KEM) and digital

signature schemes. Several schemes were proposed in both categories, to name a few: key-encapsulation mechanisms such as CRYSTALS-KYBER [SAB+20], Classic McEliece [ABC+20] and SIKE [JAC+20], and digital signature schemes including CRYSTALS-DILITHIUM [LDK+20] and `PICNIC` [CDG+17].

Although defining hard problems have been the main approach in designing post-quantum KEMs, the landscape for signature schemes is slightly different. Despite often being constructed using Cryptomania assumptions, as shown by Rompel [Rom90], one-way functions are sufficient to construct signature schemes, establishing that signatures are within the realm of Minicrypt. Moreover, a seminal result by Goldreich, Micali, and Wigderson [GMW87] established that commitment schemes imply zero-knowledge arguments for all NP statements. Relative to a random oracle, combining this result with the Fiat-Shamir transformation [FS87], immediately implies that commitment schemes are sufficient for constructing digital signature schemes. However, employing this methodology results in very large signature sizes and highly inefficient signing algorithms.

Following the blueprint of transforming ZK arguments to signatures, Ishai et al. [IKOS07] demonstrated another approach to build zero-knowledge arguments for NP based on honest-majority secure multiparty computation.[2] The basic idea of this approach involves the prover executing the NP verification algorithm in a multiparty manner "in their head" and committing to the computations performed on each share. Subsequently, the verifier requests the prover to reveal views of certain players to verify the correctness of the computation. This framework offers another avenue for building digital signatures, wherein the computation cost and signature size are determined by the expense of running the verification procedure of the NP statement in a multiparty manner.

Subsequently, Giacomelli et al. [GMO16] showed how this methodology can be further optimized for statements regarding boolean circuit evaluation. Combining this result with statements regarding the evaluation of a pseudorandom function (PRF) leads to actually practical digital signature schemes where the security can be reduced to the security of the PRF and commitment scheme used.

The interesting observation is that, in contrast to Cryptomania assumptions, assumptions in Minicrypt are not believed to be significantly impacted by the emergence of quantum computing. For instance, standardized symmetric-key encryption schemes like AES [AES01] are not believed to be vulnerable to quantum adversaries. This has led to a new avenue for designing post-quantum signature schemes using the MPC-in-the-head framework from [IKOS07] where the security only relies on the underlying symmetric cryptography schemes [CDG+17, BdK+21, dDOS19].

Quantum Cryptography:

[2] It is worth noting that although secure multiparty computation (MPC) belongs to Cryptomania, achieving honest-majority MPC can be accomplished unconditionally.

We briefly explored the challenges of enhancing the security of classical cryptography against quantum adversaries. However, in a world where quantum supremacy has been achieved, a new question arises: *"What if, in addition to the adversary, the parties involved in the cryptographic protocol are also quantum?"*

Wiesner's quantum money protocol [Wie83] and Bennett and Brassard's quantum key distribution [BB84] were some of the first concrete proposals of leveraging quantum resources which can both be viewed as ways to build cryptographic tasks which require weaker assumptions than classically needed. Recently it has been shown that oblivious transfer and arbitrary multiparty computation, both primitives which are classically in Cryptomania, can be built from Minicrypt assumptions if quantum communication is allowed [BCKM21, GLSV21]. This advancement has led to a growing belief that leveraging quantum computation might bridge the gap between Minicrypt and Cryptomania[3].

On another note, Ji et al. [JLS18] proposed a new minimalistic cryptographic primitive called pseudo-random state generators (PRSG), and demonstrated that the existence of PRSG is implied by the existence of one-way functions. Subsequently, Kretschmer [Kre21] showed that with respect to an oracle, PRSG might exist in a world in which one-way functions do not. Subsequent research improved upon this result, establishing the existence of an oracle, relative to which $\mathsf{P} = \mathsf{NP}$ while PRSGs can still be present [KQST22]. Moreover, it was shown that PRSGs are sufficient to construct quantum variants of a number of cryptographic primitives such as bit-commitments and one-time signatures which require the existence of one-way functions classically [MY22b, AQY22]. In Impagliazzo's framework, these findings suggest that quantum cryptographic tasks may be achievable even in Algorithmica.

Another area in cryptographic research where quantum computing introduces a separation is the study of interactive proofs. Interestingly, certain phenomena in quantum mechanics can be viewed as accomplishing cryptographic objectives. An example of this is the Bell inequality, where the actions of two provers achieving a CHSH score of $2\sqrt{2}$ exhibit a unique strategy, up to an isometry. This allows a classical referee to verify the behavior of two quantum parties, who share a Bell state, by communicating with them exclusively through classical means. This leads to the execution of cryptographic tasks that are typically impossible using space-separated classical parties, such as certifying the authenticity of outputs generated by a process as genuinely random. Another example of this is the seminal result of Ji et al. [JNV+22] where it was shown that two space-separated parties sharing entanglement can convince a classical verifier that a touring machine halts, i.e. $\mathsf{MIP}^* = \mathsf{RE}$. Whereas, without the shared entanglement it was shown by Babai et al. [BFL90] that the set of languages that have interactive proofs in this model is equal to NEXP.

Circling back to Impagliazzo's picture, recent results have shown how space separation can be replaced by Cryptomania assumptions for such protocols. Examples of this are the classical

[3] Classically there is evidence that public-key encryption can not be built only assuming the existence of one-way functions [IR90]

verification of quantum computation protocol by Mahadev [Mah18], and the compiler to build a computationally sound test of quantumness from any non-local game by Kalai et al. [KLVY22].

Outline of this thesis. The research presented in this dissertation addresses both questions regarding the impact of quantum computation on the field of cryptography. Consequently, the content of this thesis is divided into two main parts, each focusing on one of the aforementioned questions. Part I is dedicated to analyzing signature schemes specifically designed to ensure post-quantum security. On the other hand, Part II investigates cryptographic primitives that leverage quantum computation, which classically either rely on stronger cryptographic assumptions or are downright unattainable.

Part I: This part of the thesis consists of two chapters. In these chapters, we delve into post-quantum signature schemes based on the MPC-in-the-head paradigm from [IKOS07] and [GMO16]. As previously mentioned, the primary overhead of these schemes is closely tied to the computational cost of the MPC computation involved in the verification procedure of the NP statement. In the case of proving statements in the form of $C(x) = y$, where C represents a boolean circuit, this overhead is directly linked to the number of multiplication gates present in C. Consequently, various instantiations of this paradigm have emerged, utilizing symmetric-key primitives with low multiplication gate count or low multiplication depth. These choices aim to minimize the computational burden of the signing algorithm and reduce the size of the resulting signature.

In Chapters 2 and 3, we re-examine this selection and establish the following guiding principle:

"The security level of signatures derived from the MPC-in-the-head paradigm is directly influenced by the multiplication depth of its underlying block-cipher."

The specific signature scheme examined in these chapters is `PICNIC` [CDG+17], where the boolean circuit is instantiated with `LowMC` [ARS+15]. `LowMC` is a block cipher explicitly designed for MPC applications. We demonstrate that diminishing the number of rounds in `LowMC` to achieve improved efficiency parameters for `PICNIC` would result in significant security vulnerabilities, even when only classical security is of concern.

Part II: This part of the thesis focuses on the exploration of cryptographic primitives tailored for parties utilizing quantum computation. In Chapter 4, we revisit the concept of public-key encryption (PKE), the main pillar of Cryptomania, when keys are allowed to be quantum states. The objective of this chapter is to address the following question: *"Can public-key encryption be constructed using assumptions that are weaker than those required in classical settings within a quantum realm?"*

We affirmatively address the aforementioned question by demonstrating that it is feasible to construct a quantum public-key encryption (qPKE) scheme that offers the robust notion of

adaptive indistinguishability against chosen-ciphertext attacks (IND-CCA2), relying solely on the existence of one-way functions, placing IND-CCA2 secure qPKE within the framework of Minicrypt. Furthermore, we establish that achieving non-adaptive indistinguishability against chosen-ciphertext security (IND-CCA1) can be accomplished from assumptions that potentially fall within the realm of Algorithmica. To give a tight description, we also establish that the notion of public-key encryption still cannot provide information-theoretic security, even with the incorporation of quantum keys and ciphertexts.

In Chapter 5, we delve into the field of learning theory. A prominent area of research in modern learning theory focuses on classification tasks where the test-time examples are not drawn from the intended distribution. An example of this scenario arises when the examples are subjected to adversarial perturbations. Previous studies have explored this problem under various regimes, where the perturbations applied to the test-time examples are restricted to a fixed set, such as small perturbations in the ℓ_2 norm [SZS+14, NYC15]. However, these regimes have their limitations since ensuring classifier robustness against such perturbations necessitates prior knowledge of the set of possible perturbations, which is an unrealistic assumption in most real-world scenarios.

Golwasser et al. [GKKM20] took a different approach by proposing a regime that removes restrictions on the set of perturbations an adversary can apply, i.e. the set of test-time examples is arbitrary. However, the learning model comes with two important caveats,

1. Transductivity: The learning algorithm is provided with both the (unlabeled) test examples and the (labelled) training samples, allowing it to access the entire dataset.
2. Selectivity: The learning algorithm has the ability to abstain from answering certain classification queries, providing an additional level of flexibility.

It should be noted that without further requirements on the learning algorithm, it can be demonstrated that the task becomes intractable. Furthermore, the authors of [GKKM20] establish a lower bound on the rejection rate for any classifier aiming to be robust against an arbitrary set of test examples. This highlights the inherent challenges and limitations associated with developing classifiers that can handle unrestricted perturbations in a transductive and selective learning setting.

The crux of the problem seems to be the following question:

"Can a classifier ascertain that the queries presented to it adhere to the correct distribution?"

In classical settings, the problem is intractable, even for the specific case of a uniform distribution. There is no possible method to verify if the provided samples were uniformly sampled. However, self-testing techniques from quantum mechanics offer potential solutions to this verification problem. One such example is the use of the Bell inequality, which guarantees that

if a pair of provers achieves a CHSH score of $2\sqrt{2}$, their outputs are in fact uniformly sampled. This observation hints that answering this question might be easier in a quantum learning model.

The work presented in chapter 5 demonstrates how this idea can be generalized for any distribution which can be efficiently estimated. Building on this idea we show how in a quantum learning model, cryptographic assumptions would lead to classifiers with lower rejection rates than the lower-bound proven in [GKKM20] as long as the distribution of samples can be efficiently estimated.

Personal Bibliography. Lastly, we give a comprehensive list (in order of appearance) of all the co-authored publications accomplished in the course of this Ph.D. research. The content of the entries marked in bold font is the material included in this document. The paper "Cryptanalysis of `LowMC` instances using single plaintext/ciphertext pair" was decorated with the Best Paper Awards at its respective conference.

1. Subhadeep Banik, Khashayar Barooti, and Takanori Isobe. Cryptanalysis of Plantlet. IACR Trans. Symm. Cryptol., 2019(3):103–120, 2019. [BBI19]

2. **Subhadeep Banik, Khashayar Barooti, F. Betül Durak, and Serge Vaudenay. Cryptanalysis of LowMC instances using single plaintext/ciphertext pair. IACR Trans. Symmetric Cryptol., 2020(4):130–146, 2020 [BBDV20]**

3. **Subhadeep Banik, Khashayar Barooti, Serge Vaudenay, and Hailun Yan. New attacks on LowMC instances with a single plaintext/ciphertext pair. Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I, volume 13090 of Lecture Notes in Computer Science, pages 303–331. Springer, 2021 [BBVY21]**

4. **Grzegorz Gluch, Khashayar Barooti, and Rüdiger Urbanke. Breaking a classical barrier for classifying arbitrary test examples in the quantum model. In International Conference on Artificial Intelligence and Statistics, pages 11457–11488. PMLR, 2023 [GBU23]**

5. Khashayar Barooti, Grzegorz Głuch, and Marc-Olivier Renou. How hard is it to fake entanglement? a complexity theoretic view of nonlocality and its applications to delegating quantum computation, 2023 arxiv:2303.02080 [BGR23]

6. **Khashayar Barooti, Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-key encryption with quantum keys. Cryptology ePrint Archive, Paper 2023/877, 2023 [BGHD+23]**

7. Khashayar Barooti, Daniel Collins, Simone Colombo, Loïs Huguenin-Dumittan, and Serge Vaudenay. On Active Attack Detection in Messaging with Immediate Decryption. Advances in Cryptology - CRYPTO 2023 - 43rd International Cryptology Conference [BCC+23]

1 Preliminaries

Throughout this document, the security parameter is often denoted by λ. We write $n(\lambda) = \mathsf{poly}(\lambda)$, if there exists a polynomial f, such that $n < f(\lambda)$. We also write $n(\lambda) = \mathsf{negl}(\lambda)$, if for any polynomial f, there exists N such that for $\lambda > N$, $n(\lambda) < \frac{1}{f(\lambda)}$.

Quantum algorithms/devices are represented by circuits consisting of gates from a fixed set of universal gates. A common choice of universal gates can be found in section 1.3. We say an algorithm is in Quantum Polynomial Time (QPT) if it can be represented by a circuit with $\mathsf{poly}(n)$ gates, where n is the size of the input.

1.1 PICNIC Signature Scheme

In this section, we briefly introduce the PICNIC signature scheme which will be the main focus of chapters 2 and 3. PICNIC [CDG+17] is a highly tweakable signature scheme based on an MPC-in-head paradigm, which advanced to the third round of NIST post-quantum cryptography competition [AAC+22]. The authors propose several different parameters for various security levels and applications.

PICNIC signature is built using Fiat-Shamir transformation of a proof of knowledge protocol based on the MPC-in-head paradigm by Ishai et al. [IKOS07]. The high-level idea is as follows, imagine we have a multi-party computation of a function f. Each player has a share of the input x, and the output $y = f(x)$ is publicly known. The goal of the prover is to prove the knowledge of x. To do so, the prover simulates all players and commits to all the states and transcripts. Later the verifier is allowed to corrupt a random subset of players, having access to their full state. Having this information in hand, the verifier can check whether the computation was done correctly from the corrupted players' perspective.

In the case of PICNIC this paradigm is instantiated using a block cipher. Let $\mathsf{Enc}(K, \mathsf{pt})$ be the encryption of the plaintext pt using the key K. The function f in the previous paradigm is instantiated as $\mathsf{Enc}(*, \mathsf{pt})$ for a public plaintext pt. The plaintext/ciphertext pair $(\mathsf{pt}, \mathsf{ct})$ is used as the public key of the signature scheme (verification key), and encryption key K is used as

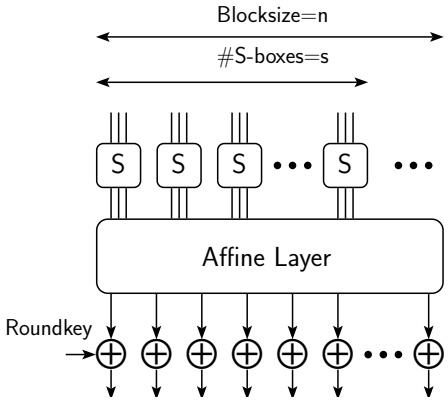


Figure 1.1: LowMC Round Function

the secret key (signing key). If an adversary can recover the encryption key given only a single ciphertext, plaintext pair (ct, pt) i.e. the public key of the signature scheme, then in effect he computes the secret signing key. This allows him to forge a signature by following exactly the honest prover protocol with the recovered signing key.

As multiplications are quite costly when computing a function in a multi-party manner, signatures such as PICNIC need to be instantiated with ciphers specifically designed for MPC use cases, for instance, ciphers with very low multiplication depth. For the case of PICNIC the chosen cipher is LowMC.

1.2 LowMC Block Cipher

LowMC [ARS+15] is an efficient block cipher tailored specifically for FHE and MPC usage, aiming to minimize the number of multiplications. LowMC uses a quadratic S-box operating on 3 bit inputs, and for each output bit of the S-box, a single multiplication is needed. The low multiplication count makes LowMC a fairly suitable choice for PICNIC instantiation.

The LowMC round function is a typical SPN construction given in Figure 1.1. It consists of an n-bit block undergoing a partial substitution layer consisting of s S-boxes where $3s \leq n$. It is followed by an affine layer which consists of the multiplication of the block with an invertible $n \times n$ matrix over $\mathbb{F}_2$ and addition with an n-bit round constant. Finally, the block is xored with the round key which is again the product of the n-bit master secret key K with an $n \times n$ invertible matrix. As in most SPN constructions, a plaintext is first xored with a whitening key which for LowMC is simply the secret key K, and the round functions are executed r times to give the ciphertext. From the point of view of cryptanalysis, we emphasise that the design is completely known to the attacker, i.e. all the matrices and constants used in the round function and key update are known.

The only non-linear component of the round function is the S-box. The LowMC S-box is a 3-bit

to 3-bit function $(x_0, x_1, x_2) \to (s_0, s_1, s_2)$ described as follows:

$$s_0 = x_0 + x_1 x_2$$
$$s_1 = x_0 + x_1 + x_0 x_2$$
$$s_2 = x_0 + x_1 + x_2 + x_0 x_1$$

1.3 Quantum Information

For a more in-depth introduction to quantum information, we refer the reader to [NC16].

Quantum States

We denote by $\mathscr{H}_M$ a complex Hilbert space with label M and finite dimension $\dim M$. We use the standard bra-ket notation to work with pure states $|\psi\rangle \in \mathscr{H}_M$. The class of positive, Hermitian, trace-one linear operators on $\mathscr{H}_M$ is denoted by $\mathscr{D}(\mathscr{H}_M)$. A quantum register is a physical system whose set of valid states is $\mathscr{D}(\mathscr{H}_M)$; in this case we label by M the register itself. The maximally mixed state (i.e., uniform classical distribution) is written as $\mathbb{I}/\dim M$ on M.

The support of a quantum state ϱ is its cokernel (as a linear operator). Equivalently, this is the span of the pure states making up any decomposition of ϱ as a convex combination of pure states. We will denote the orthogonal projection operator onto this subspace by P^ϱ. The two-outcome projective measurement (to test if a state has the same or different support as ϱ) is then $\{P^\varrho, \mathbb{I} - P^\varrho\}$.

Qubits

A qubit is the most basic quantum system used throughout this thesis. The definition we opt to use is taken from [Vid22].

Definition 1. *A qubit is a tuple $(|\psi\rangle, \mathscr{H}, X, Z)$, where $\mathscr{H}$ is a Hilbert space, $|\psi\rangle \in \mathscr{H}$ is a unit vector, and X, Z are two observables (non-singular Hermitian operators on $\mathscr{H}$) such that,*

$$(XZ + ZX)|\psi\rangle = 0 \tag{1.1}$$

In words, one typically refers to this property by saying that X and Z anticommute on the support of $|\psi\rangle$.

What anticommuting means on a high level, is that there are two ways to observe $|\psi\rangle$ but it is not possible to observe $|\psi\rangle$ in both ways simultaneously. The simplest type of qubits we encounter are defined over $\mathscr{H} = \mathbb{C}^2$. A standard choice of basis for $\mathbb{C}^2$ is,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1.2}$$

According to this choice, the state $|\psi\rangle$ can be represented as $\alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$.

Pauli Operators

Important examples of observables are the Pauli matrices, σ_X and σ_Z where,

$$\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \text{ and } \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1.3}$$

We often refer to σ_Z as the computational basis measurement and σ_X as the Hadamard basis measurement. The reason these specific two observables play a special role is the following lemma.

Lemma 1. *Let $(|\psi\rangle, \mathcal{H}, X, Z)$ be a qubit. There exists a Hilbert space $\mathcal{H}'$ and an isometry $V : \mathcal{H} \to \mathbb{C}^2 \otimes \mathcal{H}'$, such that,*

$$VX|\psi\rangle = (\sigma_X \otimes \mathbb{I})V|\psi\rangle,$$
$$VZ|\psi\rangle = (\sigma_Z \otimes \mathbb{I})V|\psi\rangle.$$

This lemma states that up to an isometry every qubit can be seen as a state on $\mathbb{C}^2$ with two observables that are the Pauli matrices. There is a canonical choice for this isometry $V : \mathcal{H} \to \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathcal{H}'$ given acting as follows:

$$V|\psi\rangle = \frac{1}{2} \sum_{a,b \in \{0,1\}} (\mathbb{I} \otimes \sigma_X^a \sigma_Z^b \otimes X^a Z^b) |\phi^+\rangle |\psi\rangle, \forall |\psi\rangle \in \mathcal{H} \tag{1.4}$$

We recall the SWAP test on two quantum states $|\psi\rangle, |\phi\rangle$ which is an efficient algorithm that outputs 0 with probability $\frac{1}{2} + \frac{1}{2}|\langle\psi|\phi\rangle|^2$. In particular, if the states are equal, the output of the SWAP test is always 0.

Next, we state a well-known fact about the quantum evaluation of classical circuits.

Fact 1. *Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function which is efficiently computable by a classical circuit. Then there exists a unitary U_f on $(\mathbb{C}^2)^{\otimes n+m}$ which is efficiently computable by a quantum circuit (possibly using ancillas) such that, for all $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$,*

$$U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle.$$

Measurements

The next concept which is used extensively in this thesis is the topic of measurements. Measurements allow us to observe the properties of a quantum state. In the real world, this can be seen as measuring the energy of a state or the spin of a particle. Measurements are often modeled as observables, let us denote one by a Hermitian operator O. As O is a Hermitian it can be

decomposed as $O = \sum_i \lambda_i \Pi_i$, where Π_i is the projection onto the eigenspace corresponding to eigenvalue λ_i. The eigenvalues are referred to as measurement outcomes and the probability of observing λ_i when measuring it on $|\psi\rangle$ is given by $\langle\psi|\Pi_i|\psi\rangle$.

There are two types of measurements, namely a projector-valued measurement (PVM) and positive operator-valued measurement (POVM). A PVM is defined by a set of projections $\{\Pi_i\}$ such that $\sum_i \Pi_i = \mathbb{I}$. The probability of obtaining measurement outcome i when measuring it on $|\psi\rangle$ is given by $|\langle\psi|\Pi_i|\psi\rangle|$. POVMs are a generalization of PVMs. For a POVM $\{\Pi_i\}$, Π_i's are not necessarily projections but can be any positive operators. We still have the requirement that $\sum_i \Pi_i = \mathbb{I}$ and the law that the probability of observing outcome i is given by $\langle\psi|\Pi_i|\psi\rangle$.

Let us also talk about the post-measurement state. When measuring a POVM $\{\Pi_i\}$ on a state $|\psi\rangle$, conditioned on the outcome of the measurement being i, the post measurement state is given by $\frac{\sqrt{\Pi_i}|\psi\rangle}{\sqrt{\langle\psi|\Pi_i|\psi\rangle}}$. Another interesting fact is that any POVM can be represented by a PVM on a bigger space. This is referred to as Neimark's dilation theorem.

Circuits and Universal Gate Set

The last concepts we cover in this section are quantum circuits and a set of universal gates. For a fixed set of unitary gates with fan-in at most t, G, a quantum circuit is represented as a tuple $(n, [(G_i, r_1, r_2, \ldots, r_t)]_{i\in\mathscr{I}})$, where n is the number of input qubits, $G_i \in G$ and $r_i \in [n] \cup \{\perp\}$. The representation means that the i^{th} operation, is applying G_i to qubits indexed $(r_1, r_2, \ldots, r_t)$. r_i is set to $\perp$ if the gate does not use that input wire. For a circuit C, we represent the corresponding unitary with U_C. The size of a circuit is often considered to be the number of gates in the circuit, i.e. $|\mathscr{I}|$.

A set of gates $\mathscr{G} = \{G_i\}_{i\in I}$ is called universal, if for any unitary U acting on a Hilbert space $\mathscr{H}$ and all $\epsilon > 0$, there exists an integer m, a circuit C of size m, using only gates in $\mathscr{G}$, such that U_C is ϵ close to U in operator norm.

A usual choice for a universal gate set is $\{H, P, \text{CNOT}, T\}$. The gates are called Hadamard, Phase, Controlled NOT, and T-gate and are described as follows:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}, P = \begin{bmatrix}1 & 0\\ 0 & -i\end{bmatrix}, \text{CNOT} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}, T = \begin{bmatrix}1 & 0\\ 0 & e^{-i\frac{\pi}{4}}\end{bmatrix} \tag{1.5}$$

1.4 Cryptographic Primitives

1.4.1 Quantum-Secure Pseudorandom Functions

Throughout this thesis, we often refer to a *pseudorandom function* (PRF) first introduced in [GGM86]. This is an ensemble of functions, indexed by a key, denoted $\{f_k\}_k$, that can be

evaluated in polynomial time satisfying a certain security property. In this thesis, we often require PRF to be *quantum-secure*, which, loosely speaking says that an adversary with oracle access to f_k cannot distinguish it from a truly random function, even when allowed to make superposition queries. It is known that quantum-secure PRF can be constructed from any quantum-secure one-way function [Zha12].

Definition 2 (Quantum-secure PRF)**.** *We say that a keyed family of functions $\{f_k\}_k$ is a* quantum-secure pseudorandom function (PRF) ensemble *if, for any QPT adversary $\mathscr{A}$, we have*

$$\left|\Pr\left[1 \leftarrow \mathscr{A}(1^\lambda)^{f_k}\right] - \Pr\left[1 \leftarrow \mathscr{A}(1^\lambda)^{f}\right]\right| \leq \mu(\lambda),$$

where $k \xleftarrow{\$} \{0,1\}^\lambda$, f is a truly random function, and the oracles can be accessed in superposition, that is, they implement the following unitaries

$$|x\rangle|z\rangle \xmapsto{U_{f_k}} |x\rangle\left|z \oplus f_k(x)\right\rangle \quad \textit{and} \quad |x\rangle|z\rangle \xmapsto{U_f} |x\rangle\left|z \oplus f(x)\right\rangle,$$

respectively.

1.4.2 Post-Quantum IND-CCA Symmetric-Key Encryption

We briefly recall the definition of a symmetric-key encryption scheme (SKE).

Definition 3. *An SKE consists of 2 algorithms with the following syntax:*

1. $\mathsf{Enc}(\mathsf{sk}, pt)$*: a* PPT *algorithm, which receives a symmetric-key* $\mathsf{sk} \in \{0,1\}^\lambda$ *and a plaintext* pt*, and outputs a ciphertext* ct*.*
2. $\mathsf{Dec}(\mathsf{sk}, ct)$*: a deterministic polynomial-time algorithm, which takes a symmetric-key* sk *and a ciphertext* ct*, and outputs a plaintext* pt*.*

We say that a SKE scheme is perfectly *correct* if for every plaintext $\mathsf{pt} \in \{0,1\}^*$ and symmetric-key $\mathsf{sk} \in \{0,1\}^\lambda$, $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, \mathsf{pt})) = \mathsf{pt}$.

Definition 4. *An SKE is post-quantum IND-CCA secure if for every QPT adversary $\mathscr{A} := (\mathscr{A}_1, \mathscr{A}_2)$, there exists a negligible function ϵ such that the following holds for all λ:*

$$\Pr\left[\tilde{b} = b \;\middle|\; \begin{array}{l} \mathsf{sk} \xleftarrow{\$} \{0,1\}^\lambda \\ pt_0, pt_1 \leftarrow \mathscr{A}_1^{\mathsf{Enc}(\mathsf{sk},\cdot),\mathsf{Dec}(\mathsf{sk},\cdot)}(1^\lambda) \\ b \xleftarrow{\$} \{0,1\} \\ ct^* \leftarrow \mathsf{Enc}(\mathsf{sk}, pt_b) \\ \tilde{b} \leftarrow \mathscr{A}_2^{\mathsf{Enc}(\mathsf{sk},\cdot),\mathsf{Dec}^*(\mathsf{sk},\cdot)}(ct^*, 1^\lambda) \end{array}\right] \leq 1/2 + \epsilon(\lambda),$$

Where $\mathsf{Dec}^*(\mathsf{sk},\cdot)$ *is the same as* $\mathsf{Dec}(\mathsf{sk},\cdot)$ *but returns* $\perp$ *on input the challenge ciphertext* ct^**.*

he adversary in this security definition is not given superposition access to the Enc, Dec oracles.

1.4.3 Pseudorandom Function-Like State (PRFS) Generators

The notion of pseudorandom function-like states was first introduced by Ananth, Qian and Yuen in [AQY22]. A stronger definition where the adversary is allowed to make superposition queries to the challenge oracles was introduced in the follow-up work [AGQY22]. We reproduce their definition here:

Definition 5 (Quantum-accessible PRFS generator)**.** *We say that a QPT algorithm G is a quantum-accessible secure pseudorandom function-like state generator if for all QPT (non-uniform) distinguishers A if there exists a negligible function ϵ, such that for all λ, the following holds:*

$$\left| \Pr_{k \leftarrow \{0,1\}^{l_\lambda}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)\rangle}(\rho_\lambda) = 1 \right] - \Pr_{\mathcal{O}_{\mathsf{Haar}}} \left[A_\lambda^{|\mathcal{O}_{\mathsf{Haar}}(\cdot)\rangle}(\rho_\lambda) = 1 \right] \right| \leq \epsilon(\lambda),$$

where:

- $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$*, on input a d-qubit register* $\mathbf{X}$*, does the following: it applies an isometry channel that is controlled on the register* $\mathbf{X}$ *containing x, it creates and stores $G_{1^\lambda}(k,x)$ in a new register* $\mathbf{Y}$*. It outputs the state on the registers* $\mathbf{X}$ *and* $\mathbf{Y}$*.*

- $\mathcal{O}_{\mathsf{Haar}}(\cdot)$*, modeled as a channel, on input a d-qubit register* $\mathbf{X}$*, does the following: it applies a channel that controlled on the register* $\mathbf{X}$ *containing x, stores $|\vartheta_x\rangle\langle\vartheta_x|$ in a new register* $\mathbf{Y}$*, where $|\vartheta_x\rangle$ is sampled from the Haar distribution*[1]*. It outputs the state on the registers* $\mathbf{X}$ *and* $\mathbf{Y}$*.*

Moreover, A_{1^λ} has superposition access to $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$ and $\mathcal{O}_{\mathsf{Haar}}(\cdot)$ (denoted using the ket notation).[2]*.*

We say that G is a $(d(\lambda), n(\lambda))$-QAPRFS generator to succinctly indicate that its input length is $d(\lambda)$ and its output length is $n(\lambda)$.

1.4.4 Quantum Pseudorandomness with Proofs of Destruction

We import the definition of pseudorandom function-like states with proofs of destruction (PRFSPD) from [BBSS23].

[1] Meaning $|\vartheta_x\rangle = U|0\rangle$, where U is sampled according to the Haar measure. The Haar measure is a canonical measure that is left invariant with respect to the action of the unitary group.

[2] As the oracle is not necessarily a unitary the action can not be defined generically by only defining it on the basis vectors. However, what we mean by accessed in superposition is $\mathcal{O}_{\mathsf{PRFS}}(k,\cdot)$ returning $\sum_x \alpha_x |x\rangle \left|\psi_k^x\right\rangle$ when queried on $\sum_x \alpha_x |x\rangle |0\rangle$

Definition 6 (PRFS generator with proof of destruction)**.** *A PRFSPD scheme with key-length $w(\lambda)$, input-length $d(\lambda)$, output length $n(\lambda)$ and proof length $c(\lambda)$ is a tuple of QPT algorithms* $\mathsf{Gen}, \mathcal{D}estruct, \mathcal{V}er$ *with the following syntax:*

1. $\left|\psi_k^x\right\rangle \leftarrow \mathsf{Gen}(k,x)$*: takes a key* $k \in \{0,1\}^w$*, an input string* $x \in \{0,1\}^{d(\lambda)}$*, and outputs an* n*-qubit pure state* $\left|\psi_k^x\right\rangle$*.*

2. $p \leftarrow \mathcal{D}estruct(\left|\phi\right\rangle)$*: takes an* n*-qubit quantum state* $\left|\phi\right\rangle$ *as input, and outputs a* c*-bit classical string,* p*.*

3. $b \leftarrow \mathcal{V}er(k,x,q)$*: takes a key* $k \in \{0,1\}^w$*, a* d*-bit input string* x*, a* c*-bit classical string* p *and outputs a boolean output* b*.*

Correctness. *A PRFSPD scheme is said to be correct if for every* $x \in \{0,1\}^d$*,*

$$\Pr_{k \xleftarrow{u} \{0,1\}^w}[1 \leftarrow \mathcal{V}er(k,x,p) \mid p \leftarrow \mathcal{D}estruct(\left|\psi_k^x\right\rangle); \left|\psi_k^x\right\rangle \leftarrow \mathsf{Gen}(k,x)] = 1$$

Security.

1. ***Pseudorandomness:*** *A PRFSPD scheme is said to be (adaptively) pseudorandom if for any QPT adversary* $\mathcal{A}$*, and any polynomial* $m(\lambda)$*, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$\left|\Pr_{k \leftarrow \{0,1\}^w}[\mathcal{A}^{|\mathsf{Gen}(k,\cdot)\rangle}(1^\lambda) = 1]\right.$$
$$\left. - \Pr_{\forall x \in \{0,1\}^d, |\phi^x\rangle \leftarrow \mu_{(\mathbb{C}^2)^{\otimes n}}}[\mathcal{A}^{\left|\mathcal{H}aar^{\{|\phi^x\rangle\}_{x\in\{0,1\}^d}}(\cdot)\right\rangle}(1^\lambda) = 1]\right| = \mathsf{negl}(\lambda)$$

   where $\forall\, x \in \{0,1\}^d$*,* $\mathcal{H}aar^{\{|\phi^x\rangle\}_{x\in\{0,1\}^d}}(x)$ *outputs* $\left|\phi^x\right\rangle$*. Here* $\mathcal{A}$ *is granted quantum access to the oracles.*

2. ***Unclonability-of-proofs****: A PRFSPD scheme satisfies* Unclonability-of-proofs *if for any QPT adversary* $\mathcal{A}$ *in cloning game (see Game 1), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr[\mathsf{Cloning\text{-}Exp}_\lambda^{\mathcal{A},PRFSPD} = 1] = \mathsf{negl}(\lambda).$$

1.4.5 Claw-Free Functions with Adaptive Hardcore Bit Property

We import the definition of claw-free families with adaptive hardcore bit property from [BCM+18].

Game 1 $\mathsf{Cloning\text{-}Exp}_{\lambda}^{\mathscr{A},\mathsf{PRFSPD}}$

1: Given input 1^{λ}, Challenger samples $k \leftarrow \{0,1\}^{w(\lambda)}$ uniformly at random.
2: Initialize an empty set of variables, S.
3: $\mathscr{A}$ gets oracle access to $\mathsf{Gen}(k,\cdot)$, $\mathcal{V}er(k,\cdot,\cdot)$ as oracle, i.e. oracle access to the algorithms Gen and $\mathcal{V}er$ with the first input set to k
4: **for** Gen query x made by $\mathscr{A}$ **do**
5: **if** $\exists$ variable $t_x \in S$ **then** $t_x = t_x + 1$.
6: **else** Create a variable t_x in S, initialized to 1.
7: **end if**
8: **end for**
9: $\mathscr{A}$ outputs $x, c_1, c_2, \ldots, c_{t_x+1}$ to the challenger.
10: Challenger rejects if c_i's are not distinct.
11: **for** $i \in [m+1]$ **do** $b_i \leftarrow \mathcal{V}er(k,x,c_i)$
12: **end for**
13: Return $\wedge_{i=1}^{m+1} b_i$.

Definition 7 (Adaptive hardcore bit property)**.** *For parameter λ, and a finite set of keys $\mathcal{K}$ a family of functions $\{f_{\mathsf{pk}}\}_{\mathsf{pk}\in\mathcal{K}}$ is called a claw-free family with adaptive hardcore bit property if,*

1. *There exists an efficient randomized generation algorithm* $(\mathsf{pk}, td) \leftarrow \mathsf{Gen}(1^{\lambda})$

2. *For all* $\mathsf{pk} \in \mathcal{K}$, $f_{\mathsf{pk}} : \{0,1\}^{m(\lambda)+1} \to \{0,1\}^{m(\lambda)}$ *is a 2-to-1 function and can be evaluated in polynomial time*

3. *For all* $\mathsf{pk} \in \mathcal{K}$, $b \in \{0,1\}$, $f_{\mathsf{pk},b} : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m(\lambda)}$ *acting as follows:*

$$x \xrightarrow{f_{\mathsf{pk},b}} f_{\mathsf{pk}}(b||x)$$

   is a bijection.

4. *For all* $\mathsf{pk} \in \mathcal{K}$, *there exists an efficient inversion algorithm* $\{x_0, x_1\} \leftarrow f_{\mathsf{pk}}^{-1}(td, y)$, *for all* $y \in \{0,1\}^{m(\lambda)}$

5. ***Correctness:***[3]

$$\Pr\left[f_{\mathsf{pk}}(0||x_0) = f_{\mathsf{pk}}(1||x_1) = y \;\middle|\; \begin{array}{l} (\mathsf{pk}, td) \leftarrow \mathsf{Gen}(1^{\lambda}) \\ y \xleftarrow{\$} \{0,1\}^{m(\lambda)} \\ (x_0, x_1) \leftarrow f_{\mathsf{pk}}^{-1}(td, y) \end{array} \right] = 1$$

6. ***Adaptive Hardcore Bit Property:*** *For all* $\mathsf{pk} \in \mathcal{K}$ *and* $b \in \{0,1\}$ *define* $G_b^{\mathsf{pk},td} \subset \{0,1\}^{m(\lambda)} \times$

[3] In the correctness and security definitions the probability is taken over the randomness of the Gen algorithm, y and the randomness of the adversary

$\{0,1\}^{m(\lambda)} \times \{0,1\}^{m(\lambda)+1}$ *as follows:*

$$G_b^{\mathsf{pk},td} = \{(x,y,d) | d \neq 0 \wedge f_{\mathsf{pk}}(x) = y \wedge d.(0||x_0 + 1||x_1) = b, \text{ where } (x_0,x_1) \leftarrow f_{\mathsf{pk}}^{-1}(td,y)\}$$

The family satisfies the adaptive hardcore property if for all QPT(λ) *adversaries* $\mathscr{A}$ *returning* $(x,y,d) \leftarrow \mathscr{A}(\mathsf{pk})$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that the following holds on average over the random coins of the* Gen *algorithm and* $\mathscr{A}$[4]*:*

$$\begin{aligned} &(\mathsf{pk}, td) \leftarrow \mathsf{Gen}(1^\lambda) \\ &|\Pr[\mathscr{A}(\mathsf{pk}) \in G_0^{\mathsf{pk},td}] - \Pr[\mathscr{A}(\mathsf{pk}) \in G_1^{\mathsf{pk},td}]| \leq \mathsf{negl}(\lambda) \end{aligned} \tag{1.6}$$

To give a high-level description of the property, the security definition guarantees that it is not possible for an adversary to provide one preimage and a single bit of information about the other preimage at the same time for an image y of their choice.

1.5 The Collapsing Property

The final security definition we cover in this section is the collapsing property. Defined by Unruh [Unr16], the collapsing property can be seen as the quantum variant of collision resistance.

Definition 8. *A family of functions* $\{f_k\}_k$ *is called collapsing if for all large* λ *any QPT adversary* $\mathscr{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

$$|\Pr[\textit{Collapsing-Exp}_\lambda^{\mathscr{A},\{f_k\}_k} = 1] - 1/2| \leq \mathsf{negl}(\lambda)$$

1.6 Probabilities and Learning Theory: Basic Facts and Definitions

For $k \in \mathbb{N}$ we denote by $[k]$ the set $\{1,\ldots,k\}$ and by $\mathfrak{D}(n)$ the family of distributions on n-bit strings. For $\mathscr{P},\mathscr{Q} \in \mathfrak{D}(n)$ we define their Hellinger distance as $d_H(\mathscr{P},\mathscr{Q}) := \frac{1}{\sqrt{2}}\|\sqrt{\mathscr{P}} - \sqrt{\mathscr{Q}}\|_2$.

We denote the total variation distance of $\mathscr{P},\mathscr{Q}$ as $\Delta(\mathscr{P},\mathscr{Q}) = \frac{1}{2}\|\mathscr{P} - \mathscr{Q}\|_1$. The two similarity measures satisfy $d_H^2(\mathscr{P},\mathscr{Q}) \leq \Delta(\mathscr{P},\mathscr{Q}) \leq \sqrt{2}d_H(\mathscr{P},\mathscr{Q})$. A direct calculation yields the useful identity $1 - d_H^2(\mathscr{P},\mathscr{Q}) = \sum_{x\in\{0,1\}^n}\sqrt{\mathscr{P}(x)\mathscr{Q}(x)}$, where $\mathscr{P}(x)$ is the probability mass function of $\mathscr{P}$.

Fact 2 (Chernoff-Hoeffding)**.** *Let* $X_1,\ldots,X_k$ *be independent Bernoulli variables with parameter* p*. Then for every* $0 < \epsilon < 1$

$$Pr\left[\left|\frac{1}{k}\sum_{i=1}^{k} X_i - p\right| > \epsilon\right] \leq 2e^{-\frac{\epsilon^2 k}{2}}.$$

[4]The probabilities in equation 1.6 are taken over the randomness of $\mathscr{A}$ and equation 1.6 holds on average over the randomness of Gen

Game 2 Collapsing-EXP$^{\mathscr{A},\{f_k\}_k}_{\lambda}$

1: Given input 1^λ, Challenger samples $k \leftarrow \{0,1\}^{w(\lambda)}$ uniformly at random.
2: $|\phi\rangle \leftarrow \mathscr{A}(k, 1^\lambda)$, let us write $|\phi\rangle = \sum_x \alpha_x |x\rangle$
3: Evaluate f_k on superposition on $|\phi\rangle$, i.e. $|\phi'\rangle = \sum_x \alpha_x |x\rangle |f_k(x)\rangle_{\text{out}}$
4: Measure the out register to obtain y and a state $|\phi'\rangle = \frac{1}{\sum_{x:f_k(x)=y} |\alpha_x|^2} (\sum_{x:f_k(x)=y} \alpha_x |x\rangle)$
5: $b \xleftarrow{\$} \{0,1\}$
6: **if** $b = 0$ **then** $\phi_{\text{challenge}} = |\phi'\rangle$
7: **else**
8: Measure $|\phi'\rangle$ in computational basis and set $\phi_{\text{challenge}}$ to be the post-measurement state $\frac{1}{\sum_{x:f_k(x)=y} |\alpha_x|^2} (\sum_{x:f_k(x)=y} |\alpha_x|^2 |x\rangle \langle x|)$
9: **end if**
10: $\tilde{b} \leftarrow \mathscr{A}(y, |\phi_{\text{challenge}}\rangle)$
11: **return** $1_{b=\tilde{b}}$

The learning tasks we consider in this document are classification tasks. Given a relation $R \subseteq \mathscr{X} \times \mathscr{C}$, on query $x \in \mathscr{X}$, the goal of a classifier is to output $c \in \mathscr{C}$ such that $(x,c) \in R$. The elements of $\mathscr{X}$ are referred to as samples and the elements of $\mathscr{C}$ are referred to as classes. A classification task is called separable if each $x \in \mathscr{X}$ belongs to exactly one class $c \in \mathscr{C}$, i.e. R is a function. Furthermore, the classification task is a binary classification if R is a boolean function, i.e. $\mathscr{C} = \{1,-1\}$. For separable tasks, the class relation R is often called the ground truth and we often denote it by a boolean function g, i.e. $g(x) = b$ if and only if $(x,b) \in R$.

A *supervised* learning algorithm for a boolean classification task g, consists of two phases:

1. The training phase: in this phase, the learning algorithm gets labelled samples from a distribution $\mathscr{D}$, i.e. $(x, g(x))_{x\sim\mathscr{D}}$. At the end of the training phase, it outputs a hypothesis function f. The number of samples that the learning algorithm is given is often referred to as the sample complexity.

2. The testing phase: in this phase, the learning algorithm is queried on examples from a distribution $\mathscr{D}'$, i.e. the hypothesis function f is queried.

The training distribution $\mathscr{D}$ is often referred to as the *nature*. In most cases, the error of the learning algorithm is measured with respect to the samples coming from nature, i.e. the training distribution and the test distribution are considered to be the same $\mathscr{D} = \mathscr{D}'$. The standard risk of a learning algorithm outputting a hypothesis function f is defined as follows.

Definition 9 (**Standard Risk**). *For a separable binary classification task with distribution $\mathscr{D}$ and a ground truth g we define the standard risk of f as*

$$R^g_{\mathscr{D}}(f) := Pr_{x\sim\mathscr{D}}[f(x) \neq g(x)].$$

Post-Quantum Cryptography Part I

2 Ruining a PICNIC, Act 1

In this chapter, we explore cryptanalysis of a post-quantum signature scheme PICNIC. We present the first known cryptanalysis results on LowMC in a scenario where the adversary has access to only a single plaintext/ciphertext pair, which corresponds to the security guarantee required for the usecase of PICNIC. The personal contributions of this chapter are mainly taken from joint work with Subhadeep Banik, Betül Durak and Serge Vaudenay published at IACR-ToSC 2020 [BBDV20].

Structure of the Chapter: In section 2.2 we briefly introduce the LowMC paramterization of interest for the security of PICNIC based on the LowMC cryptnalysis challenge. In section 2.3 we provide a review of the previously known cryptanalysis attempts on LowMC. In section 2.4 we introduce a technique to linearize the LowMC S-box which is used later in section 2.5 to realize an attack on two sets of parameters for LowMC. In section 2.6 we introduce a meet-in-the-middle attack (MITM) employing the linearization technique. We later show the attack can be optimized using the 3-xor problem in section 2.7. We conclude this chapter in section 2.8.

2.1 LowMC as an Attribute of PICNIC

PICNIC is a post-quantum signature scheme built based on the MPC-in-the-head interactive proofs for statements regarding evaluations of boolean circuits [GMO16]. On a highlevel, to prove their identity, a signer proves the knowledge of a key K such that $\mathsf{PRP}_K(x) = y$ for a public pair (x, y), where PRP is a pseudo-random permutation. As the size of the proof and the prover efficiency from [GMO16] heavily relies on the number of AND gates required to compute $\mathsf{PRP}_K(\cdot)$, the framework is often instanciated with PRPs with minimal multiplication count. A way to do this is by decomposing the the PRP into linear layers and t-fan-in, t-fan-out non-linear operations. The smallest non-linear operation that fits this framework is a 3-fan-in 3-fan-out quadratic gate. This is exactly the philosophy taken in the design of LowMC. A more detailed description of PICNIC and LowMC in sections 1.1 and 1.2.

An important question regarding the signing efficiency of such signatures is how low can the number of the non-linear operations be such that the security is still intact? One important observation is that for the derived signature to be secure, we do not necessarily need the full security of the PRP. Indeed, the adversary is only given the evaluation of the PRP on a single point x rather than being provided an evaluation oracle. This would mean that one might hope to reduce the number of non-linear operations further than normally allowed without harming the usual PRP security, for the specific use-case of digital signatures. The goal of this segment of the dissertation is to establish that even in this setting, reducing the number of non-linear operations excessively would lead to serious security problems.

2.2 LowMC Cryptanalysis Challenge and Paramters

In the pursuit of finding the optimal number of rounds for PICNIC, Rechberger et al. announced a cryptanalysis challenge for LowMC family of block ciphers, specifically for the PICNIC use case [ARS+15].

From the point of view of cryptanalysis, the design is completely known to the attacker, i.e. all the matrices and constants used in the round function and key update are known. We denote the number of rounds by r, the number of S-boxes in each layer by s and the block size by n.

The LowMC challenge specifies 9 challenge scenarios for key recovery given only 1 plaintext-ciphertext pair, i.e. the data complexity $d = 1$.

- $n = 128,\ s = 1$
- $n = 128,\ s = 10$
- $n = 129,\ s = 43$ (full S-box layer)
- $n = 192,\ s = 1$
- $n = 192,\ s = 10$
- $n = 192,\ s = 64$ (full S-box layer)
- $n = 256,\ s = 1$
- $n = 256,\ s = 10$
- $n = 255,\ s = 85$ (full S-box layer)

The number of rounds r for instances with the full S-box layer is either 2, 3, or 4 and for instances with a partial S-box layer can vary between $0.8 \times \lfloor \frac{n}{s} \rfloor$, $\lfloor \frac{n}{s} \rfloor$ and $1.2 \times \lfloor \frac{n}{s} \rfloor$. The key length k for all instances is n bits. In general instantiations of LowMC, the key size and block size are not the same. The whitening key and all the round keys are extracted by multiplying

Instance	n	s	r	Type of Attack	Complexity	Section
Full S-box layer	129	43	2	Linearization	2^{86}	4
	192	64			2^{128}	
	255	85			2^{170}	
Partial S-box layer	128	1	$0.8 \times \lfloor \frac{n}{s} \rfloor$	Linearization	2^{102}	4
	192	1			2^{154}	
	256	1			2^{205}	
Partial S-box layer	128	10	$0.8 \times \lfloor \frac{n}{s} \rfloor$	Linearization	2^{100}	4
	192	10			2^{150}	
	256	10			2^{200}	
Full S-box layer	129	43	2	Linearization + MITM	2^{109}	5
	192	64			2^{161}	
	255	85			2^{214}	
Partial S-box layer	128	1	$0.8 \times \lfloor \frac{n}{s} \rfloor$	Linearization+ MITM	2^{124}	5*
	192	1			2^{186}	
	256	1			2^{248}	
Partial S-box layer	128	10	$0.8 \times \lfloor \frac{n}{s} \rfloor$	Linearization + MITM	2^{124}	5*
	192	10			2^{186}	
	256	10			2^{248}	
Full S-box layer	129	43	2	Linearization + MITM+3-xor	2^{106}	6
	192	64			2^{158}	
	255	85			2^{211}	

Table 2.1: Summary of results. Note for the Linearization+MITM (+3-xor) approaches the complexity is given in "evaluations of a quadratic expression". For the Linearization only approach, the complexity is in "number of Gaussian eliminations." * As explained in Section 4, these are best case complexities that occur for around 29% of the LowMC instances.

the master key with full rank matrices over GF(2). However, for all the instances of LowMC used in the LowMC challenge the block size and key size are the same. This being so, the lengths of the master key, whitening key and all the subsequent round keys are the same. Effectively, this makes all these keys related to each other by multiplication with an invertible matrix over GF(2). Thus all round keys can be extracted by multiplying the whitening key with an invertible matrix. So for all practical purposes used in this work, the whitening key can also be seen as the master secret key. This is true since given any candidate whitening key, all round keys can be generated from it, and thus given any known plaintext-ciphertext pair, it is possible to verify if that particular candidate key has been used to generate the corresponding pt/ct pair. As such we use the terms master key/whitening key interchangeably.

2.3 Previous Work

In ICISC 2015 Dobraunig et al. [DEM15] proposed an attack on LowMC family of block ciphers, based on cube attack strategies. The authors proposed an algorithm which successfully recovers the key of the round reduced version of the cipher, aiming for 80-bit security. Dinur et al. [DLMW15] showed that around 2^{-38} fraction of its 80-bit key instances could be broken

2^{23} times faster than exhaustive search. Moreover, all instances that claimed to provide 128-bit security could be broken about 1000 times faster. In [DKP+19], the authors showed that for the LowMC instances that employ partial linear layers, each instance belonged to a large class of equivalent instances that differ in their linear layers. This led to a more efficient implementation of the cipher that reduces the evaluation time and storage of computing the linear layers. In FSE 2018, Rechberger et al. [RST18] proposed a meet-in-the-middle style attack, based on possible output differentials, given an input differential, which affects the security of the variants of LowMCv2 with partial S-box layers drastically. In [LIM20] some results on LowMC were reported building on the techniques of [RST18], albeit with higher data complexities, which naturally do not apply to the PICNIC scenario. In [DN19] the authors proposed multi-target attacks on the PICNIC signature scheme. For a survey of key recovery attacks on LowMC, readers may check the survey done by Rechberger et al. [DKRS].

2.4 Linearizing the LowMC S-box

Let us for example take f to be the majority function computed on the inputs of the 3 input bits, i.e. $f = x_0 \cdot x_1 + x_1 \cdot x_2 + x_0 \cdot x_2$, where all the operations are over $\mathsf{GF}(2)$. Then the expressions of the S-box can be rewritten as

$$\begin{aligned} s_0 &= f \cdot (x_1 + x_2 + 1) + x_0, \\ s_1 &= f \cdot (x_0 + x_2 + 1) + x_0 + x_1, \\ s_2 &= f \cdot (x_0 + x_1 + 1) + x_0 + x_1 + x_2 \end{aligned}$$

This means that if we guess the value of the single expression f (0 or 1), then the entire S-box becomes an affine function in the input bits. The same holds for the inverse S-box. In fact we can replace f with any balanced 3-variable Boolean function of degree 2, and still get the same results as we prove in the following lemma.

Lemma 2. *Consider the LowMC S-box S defined over the input bits x_0, x_1, x_2. If we guess the value of any 3-variable quadratic Boolean function f which is balanced over the input bits of the S-box, then it is possible to re-write the S-box as an affine function of its input bits.*

Proof. The general expression for a 3 variable quadratic Boolean function is

$$f = A + Bx_0 + Cx_1 + Dx_2 + Ex_0 \cdot x_1 + Fx_1 \cdot x_2 + Gx_0 \cdot x_2.$$

The only non-linear terms in the expression of the LowMC S-box are $x_0 \cdot x_1$, $x_1 \cdot x_2$, $x_0 \cdot x_2$. Thus if there exists a Boolean function of the above form, which when multiplied with different linear functions can produce each of the terms $x_0 \cdot x_1$, $x_1 \cdot x_2, x_0 \cdot x_2$, then we are done. Thus the necessary and sufficient conditions required to achieve the above is to prove the existence

of 3 affine Boolean functions $g_i = a_i x_0 + b_i x_1 + c_i x_2 + d_i,\ \forall i \in [0,2]$, such that

$$f \cdot g_0 = x_0 \cdot x_1 + l_0(x_0, x_1, x_2)$$
$$f \cdot g_1 = x_1 \cdot x_2 + l_1(x_0, x_1, x_2)$$
$$f \cdot g_2 = x_0 \cdot x_2 + l_2(x_0, x_1, x_2)$$

where l_0, l_1, l_2 are some affine functions on x_0, x_1, x_2. If these functions g_i exist, we can write each of the three output bits of the LowMC S-box as

$$x_0 + f \cdot g_1 + l_1,\ x_0 + x_1 + f \cdot g_2 + l_2,\ x_0 + x_1 + x_2 + f \cdot g_0 + l_0$$

So in order for the first equation to be satisfied, we need that the product of f and g_0 produces coefficients $0,1,0,0$ for the terms $x_0 \cdot x_1 \cdot x_2,\ x_0 \cdot x_1,\ x_1 \cdot x_2,\ x_0 \cdot x_2$ respectively. In matrix form this can be written as $\mathbf{M} \cdot [a_0,\ b_0,\ c_0,\ d_0]^T = [0,1,0,0]^T$, where

$$\mathbf{M} = \begin{bmatrix} F & G & E & 0 \\ C+E & B+E & 0 & E \\ 0 & D+F & C+F & F \\ D+G & 0 & B+G & G \end{bmatrix}$$

Similarly the other 2 equations can be written as $\mathbf{M} \cdot [a_1,\ b_1,\ c_1,\ d_1]^T = [0,0,1,0]^T$ and $\mathbf{M} \cdot [a_2,\ b_2,\ c_2,\ d_2]^T = [0,0,0,1]^T$. It is therefore clear that for the equations to have a solution we need $\mathbf{M}$ to be invertible. Since the number of 3-variable quadratic Boolean functions f is just 2^7, we can perform the following small computer exercise: we can construct the matrix $\mathbf{M}$ for each function f and test whether it is invertible or not. We found that all functions f for which $\mathbf{M}$ is invertible, are exactly the functions that are balanced.

□

For example, if we take $f = s_0 = x_0 + x_1 \cdot x_2$, the S-box functions can be written as

$$s_0 = f,$$
$$s_1 = f \cdot (x_2 + 1) + x_1,$$
$$s_2 = f \cdot (x_1 + 1) + x_1 + x_2$$

2.5 Cryptanalysis by Linearization

The first technique to break LowMC by linearization is for instances for which the total number of S-boxes is less than the key length. This occurs for the following cases:

1. All instances of full S-box layer with number of rounds = 2.
2. All instances of partial S-box layer with number of rounds = $0.8 \times \lfloor \frac{n}{s} \rfloor$.

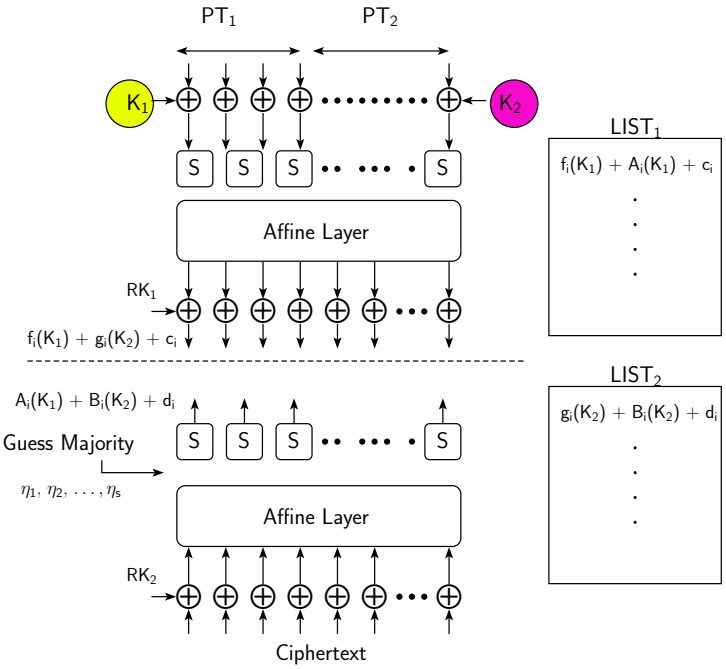


Figure 2.1: Meet in the Middle

The idea is as follows. We guess the value of the majority function at the input of all the S-boxes in the encryption circuit. When we do so the expression relating the plaintext and ciphertext becomes a linear expression in the key variables, i.e. of the form

$$A \cdot [k_0, k_1, \ldots, k_{n-1}]^T = \text{const},$$

where A is an $n \times n$ matrix over $\mathsf{GF}(2)$. Thus the key can be found using Gaussian elimination. After this a wrong key can be discarded by simply recalculating the encryption function with the derived key and plaintext and checking if the result equals the given ciphertext or not. Of course, we need not compute the full encryption: a key can be discarded as soon as the majority function computed at the input of one of the s-boxes differs from the value used to linearize the circuit. If the total number of s-boxes in the circuit is t, then the worst case complexity of the process is 2^t gaussian eliminations calculations. For example this is 2^{86} for the `LowMC` instance with $n = 129, s = 43, r = 2$. However, there is an added cost in this process. For any guess of the majority values, the matrix A computed above may not necessarily be invertible. If the dimension of the kernel of the matrix A is d_A, then we can see that $O(2^{d_A})$ keys would satisfy any equation of the form $A \cdot K = \text{const}$. Thus the verification would require running the verification for 2^{d_A} candidate keys. Moreover, we did not find any easy way to find a closed form for any bound on d_A.

2.6 Meet-in-the-Middle approach

2.6.1 2 round full S-box layer

The complexity of the attack in the previous section was measured in terms of number of Gaussian eliminations. Even while bypassing the Gaussian Elimination method, the algorithm will still require an additional computational step (evaluating all elements in the kernel of A). In this section, we present attacks whose complexity is measured in a much simpler and more tangible metric: "number of evaluations of a quadratic expression in keybits". We describe a meet-in-the-middle approach for the two-round variant of LowMC. The idea is to first split the key into two parts $K_1 = [k_0, \ldots, k_{t-1}]^T$ and $K_2 = [k_t, \ldots, k_{n-1}]^T$, each of around $t \approx \frac{n}{2}$ bits. By guessing the majority bits (or any other balanced quadratic function) of the second layer S-box we can make the second round linear as described above. After this, it is possible to adopt a meet-in-the-middle approach, by guessing first the K_1 value and making a list based on each guess. We later independently guess K_2 and create a list based on the guessed values and search for a collision in the obtained lists.

The idea is as follows. As proven in Lemma 2, if we know the value of a balanced quadratic boolean function in the input bits of each S-box, i.e. the majority, we can write the S-box as an affine function in the input bits. The same argument holds for the inverse S-box (since the inverse S-box is also a quadratic permutation over $\{0,1\}^3$). Again let us denote by R_1, R_2 the first and second round functions i.e. $R_1(\mathsf{pt} + \mathsf{RK}_0, \mathsf{RK}_1) = x$ and $R_2(x, \mathsf{RK}_2) = \mathsf{ct}$, where x denotes the n-bit input to the second round and $\mathsf{RK}_1, \mathsf{RK}_2$ denotes the first, second round keys, respectively, which are of course linear functions of the original key $K = \mathsf{RK}_0$. As shown in Figure 2.1, we start with the ciphertext backwards and try to reach the state at the input to the second round. To do this we first perform the inverse affine function operation on the vector $\mathsf{ct} \oplus \mathsf{RK}_2$ (where RK_2 is expressed in terms of K_1 and K_2). Thereafter we guess the s majority bits $\eta_1, \ldots, \eta_s$ at the input of the second round inverse S-boxes to linearize R_2. After this, each bit of x can be written as an affine function of the key and the ciphertext. In fact denoting each bit of x as x_i, we can further write $x_i = A_i(K_1) + B_i(K_2) + d_i$, $\forall\ i \in [0, n-1]$, where each A_i, B_i are linear functions over K_1, K_2 and d_i is a single bit constant.

Similarly it is possible to compute x from the plaintext in the forward direction. Even if we do not guess the majority of the first round s-boxes, K_1 and K_2 can be chosen such that the bits of K_1 and K_2 are never multiplied in the first round function. For example for $n = 129$, K_1 can be taken to be the first $t = 3 \times \lfloor s/2 \rfloor = 63$ bits of the key and K_2 to be the remaining 66 bits. The only source of non-linearity in the first round are the S-boxes, and each S-box either gets the bits of K_1 or K_2 as inputs and so K_1 and K_2 are not mixed in a multiplicative sense in this round. This being the case, after the affine layer and addition of RK_1, each bit x_i can be written as $f_i(K_1) + g_i(K_2) + c_i$ where each f_i, g_i are at most quadratic functions over K_1, K_2 and c_i is a single bit constant. Given the equality $x_i = f_i(K_1) + g_i(K_2) + c_i = A_i(K_1) + B_i(K_2) + d_i$, we can rearrange the terms to get

$$f_i(K_1) + A_i(K_1) + c_i = g_i(K_2) + B_i(K_2) + d_i, \ \forall \ i \in [0, n-1]$$

We are now ready to state the attack. Let the plaintext be $\mathsf{pt} = [\mathsf{pt}_0, \mathsf{pt}_1 \ldots, \mathsf{pt}_{n-1}]$, and $\mathsf{ct} = [c_0, c_1, \ldots, c_{n-1}]$ be the corresponding ciphertext. Take $t = 3 \times \lfloor s/2 \rfloor \approx \frac{n}{2}$. We proceed as follows:

1. Calculate the functional forms of f_i, g_i and c_i for all $i \in [0, n-1]$.
2. Guess the values $\eta_1, \ldots, \eta_s$. This step is done 2^s times in the worst case.
   - Compute A_i, B_i, d_i for all $i \in [0, n-1]$ using the guessed values.
   - For all possible values of K_1, create a hash table LIST_1 indexed by the n-bit vector $[f_i(K_1) \oplus A_i(K_1) \oplus c_i], \ \forall \ i \in [0, n-1]$. We need 2^t operations in this step.
   - For all possible values of K_2, create a hash table LIST_2 indexed by the n-bit vector $[g_i(K_2) \oplus B_i(K_2) \oplus d_i], \ \forall \ i \in [0, n-1]$. We need 2^{n-t} operations in this step.
   - Find a collision between LIST_1 and LIST_2.
   - When a collision is found for K_1 and K_2 check if the majority bits are consistent with the guess of the key. If yes, this key is in fact the encryption key. Otherwise try another guess of $\eta_1, \ldots, \eta_s$.

In practice, 2 hash tables are not necessary. The attacker can insert each new vector of LIST_1 and LIST_2 into a single hash table and wait until a collision between elements of LIST_1 and LIST_2 is found. For each set of majority guesses, the complexity of the attack is dominated by finding a collision between two lists of length 2^t and 2^{n-t} each. So for $n = 129$, we can take $t = 63$ (key bits added before the first 21 S-boxes) and $n - t = 66$. The total complexity of the attack is $O\left(2^s \times (2^t + 2^{n-t})\right)$, which for the $n = 129$ bit version is around $2^{43+66} = 2^{109}$.

2.6.2 MITM on partial S-box layers

In order to perform a MITM on the partial S-box layer instances of LowMC, we rearrange the first r_1 and final r_3 rounds so that the total number of different key bits involved in these rounds is $3s$ per round. The transformations are shown in Figures 2.2, 2.3 and are similar to the ones used in [RST18]. In fact the transform used in the backward direction (see Fig 2.3) is exactly same as the one used in [RST18, Fig.1]. The idea is that the affine layer and key addition are interchangeable. Since if L is a linear function, we have $L(x) + K = L(x + L^{-1}(K))$ and similarly $L(x + K) = L(x) + L(K)$. Hence the key addition can be moved before or after the affine layer as required, by multiplying the round key by the appropriate matrix. Fig 2.2 further shows how to transform the first r_1 rounds.

We partition the $r = r_1 + r_2 + r_3$ rounds of LowMC into the first r_1, middle r_2 and final r_3 rounds, and further transform the first r_1 and the final r_3 rounds so that each round has only $3s$

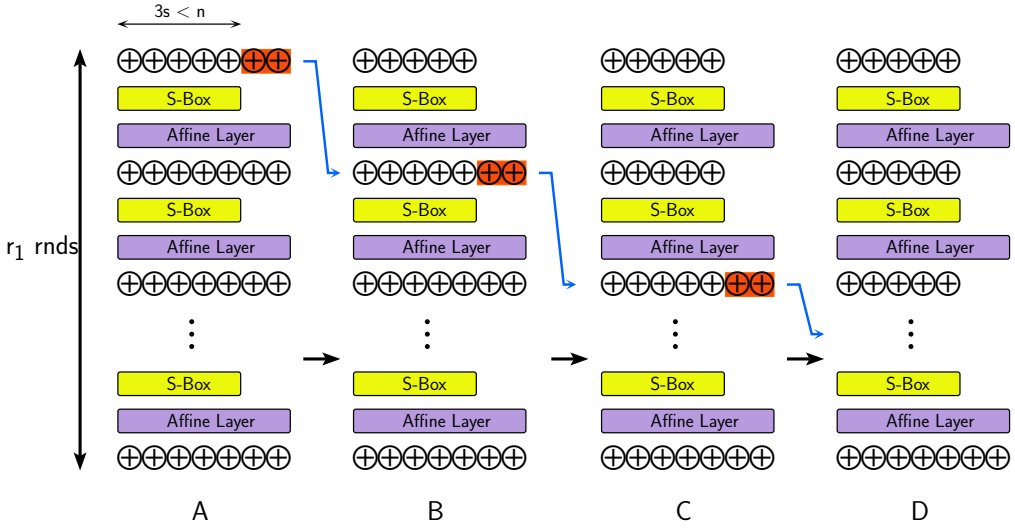


Figure 2.2: Transforming the round function in the first r_1 rounds. From A → B, the key material not added to bits input to the S-box in round 1 (shown in orange background) are carried to the next round, through the affine layer and merged with the round key in round 2. B → C → D do the same from the second round onwards.

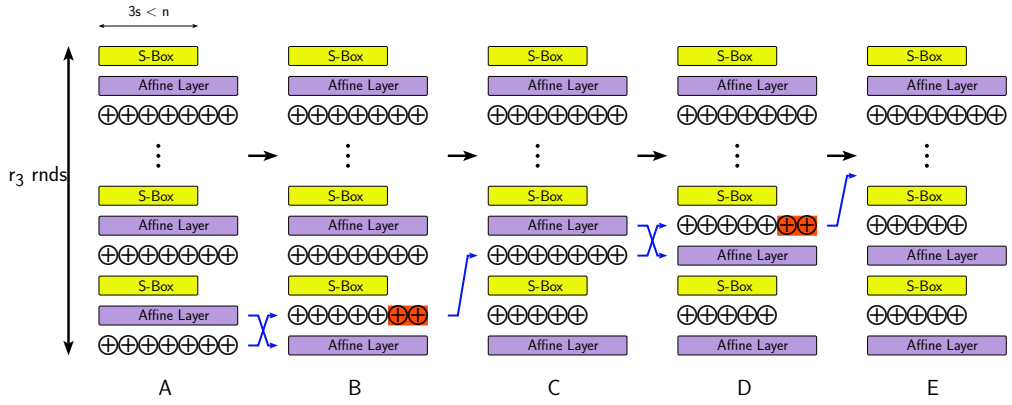


Figure 2.3: Transforming the round function in the final r_3 rounds. A → B flips the order of the last round Affine layer and round key xor. B → C takes the bits of the last round key that are not added to S-box outputs (shown in orange background), and brings them back by 1 round and merges it with the penultimate round key. C → D flips the order of the Affine layer and round key of the penultimate round, and D → E generalizes the process from this point onwards.

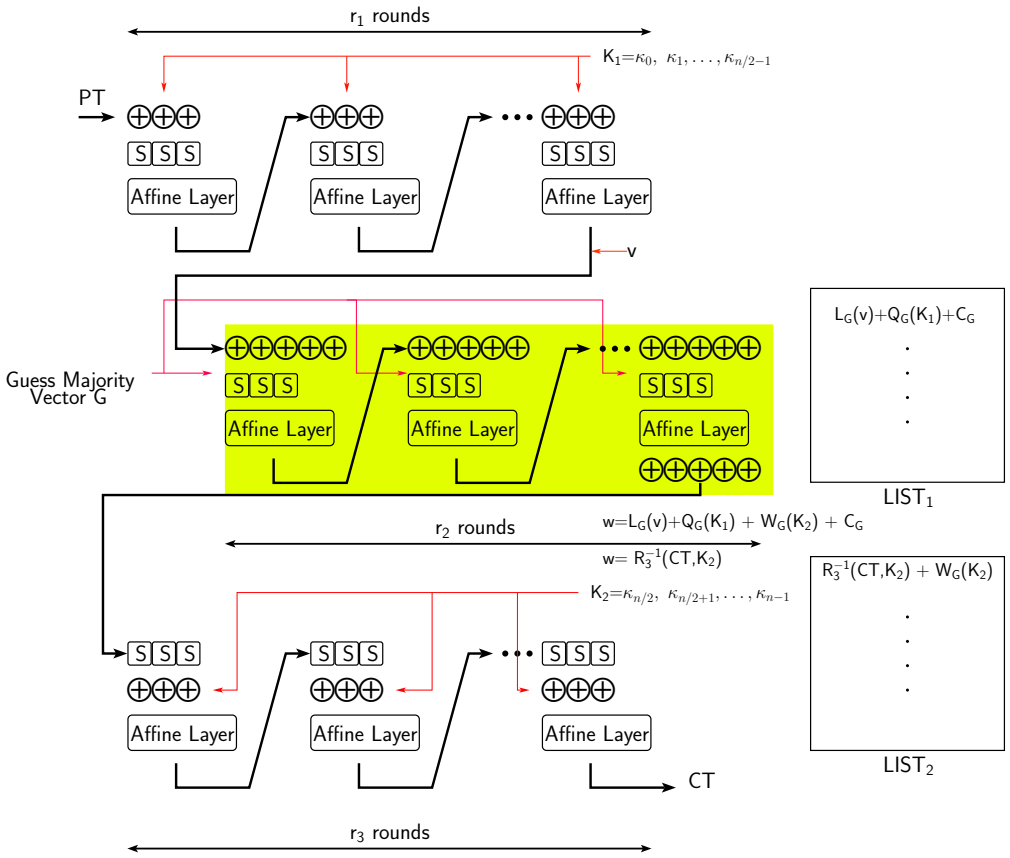


Figure 2.4: MITM on $r_1 + r_2 + r_3$ rounds with partial S-box layers. In fact the first r_1 and last r_3 rounds have been transformed as per the procedures explained in Figures 2.2, 2.3. We guess the majority bits at the S-box inputs in the middle r_2 rounds (shown against yellow background) so that they become affine.

keybits. If $r_1 = r_3 = \lfloor \frac{n}{6s} \rfloor$, then there are a total of n keybits in these rounds. Naming these keybits as $\kappa_0, \kappa_1, \ldots, \kappa_{n-1}$. Let us assume that these n keybits result from linearly independent expressions on the master key bits (in the next subsection we will see what happens when this is not the case). Then it is not difficult to see that all the keybits in the middle $r_2 = 0.8\lfloor \frac{n}{s} \rfloor - \lfloor \frac{n}{3s} \rfloor$ rounds can be written as linear functions of $\kappa_0, \kappa_1, \ldots, \kappa_{n-1}$. Now let us divide the keybits into $K_1 = [\kappa_0, \kappa_1, \ldots, \kappa_{n/2-1}]$ and $K_2 = [\kappa_{n/2}, \kappa_{n/2+1}, \ldots, \kappa_{n-1}]$ where K_1 and K_2 are the keybits used in the first r_1 and the final r_3 rounds respectively.

Now if all the $s \cdot r_2$ majority bits of the middle r_2 rounds are guessed, then the transformation in the middle r_2 rounds becomes completely affine. If G is the vector of these $s \cdot r_2$ majority bits, let us denote this affine transformation in the middle rounds as $L_G(x) + Q_G(K_1) + W_G(K_2) + C_G$, where L_G is a linear function from $\{0,1\}^n \to \{0,1\}^n$ and Q_G, W_G are linear functions over $\{0,1\}^{n/2} \to \{0,1\}^n$ and C_G is an n-bit constant. Let v be the n-bit vector obtained by executing the r_1 forward rounds by guessing some value of K_1, and let w be the vector obtained after $r_1 + r_2$ rounds. Then after guessing G we have $w = L_G(v) + Q_G(K_1) + W_G(K_2) + C_G$. Now w

can also be obtained by guessing K_2 and executing the inverse of the final r_3 rounds on the ciphertext. If R_3 denotes the transformation in the last r_3 rounds, we have $w = R_3^{-1}(\mathsf{ct}, K_2)$ So we have $R_3^{-1}(\mathsf{ct}, K_2) = L_G(v) + Q_G(K_1) + W_G(K_2) + C_G$. Rearranging terms we have

$$R_3^{-1}(\mathsf{ct}, K_2) + W_G(K_2) = L_G(v) + Q_G(K_1) + C_G$$

Then our meet-in-the-middle algorithm will proceed as follows.

1. Guess the vector G of the $s \cdot r_2$ majority values in the middle rounds. Find the functions L_G, W_G, K_G and C_G. This step is done $2^{s \cdot r_2}$ times in the worst case.
   - For all possible values of K_2, create a hash table LIST_2 indexed by the n-bit vector $R_3^{-1}(\mathsf{ct}, K_2) + W_G(K_2)$. We need $2^{n/2}$ operations in this step.
   - For all possible values of K_1, create a hash table LIST_1 indexed by the n-bit vector $L_G(v) + Q_G(K_1) + C_G$. We need $2^{n/2}$ operations in this step.
   - Find a collision between LIST_1 and LIST_2.
   - When a collision is found for K_1 and K_2 check if the majority of bits are consistent with the guess of the key. If yes, this key is in fact the encryption key. Otherwise try another guess of G.

The procedure has been explained diagrammatically in Figure 2.4. Again as explained before, 2 hash tables are not necessary in practice. The attacker can insert each new vector of LIST_1 and LIST_2 in a single hash table and wait till a collision between elements of LIST_1 and LIST_2. The majority of the computational complexity is taken by the guessing of G and computing $R_3^{-1}(\mathsf{ct}, K_2) + W_G(K_2)$ for each guess of K_2 and $L_G(v) + Q_G(K_1) + C_G$ for each guess of K_1. This part takes $2^{s \cdot r_2} \cdot 2^{1+n/2} \approx 2^{s \cdot r - n/3 + n/2} = 2^{rs + n/6}$. For $r = 0.8\left\lfloor \frac{n}{s} \right\rfloor$, this complexity is around $2^{29n/30}$.

2.6.3 When all the key expressions $\kappa_i,\ i \in [0, n-1]$ are not linearly independent

Note that each κ_i is a linear expression in the n master key bits, and so it may turn out that the n linear expressions for $\kappa_i,\ i \in [0, n-1]$ are not linearly independent. Assuming each κ_i is a random linear expression, the probability that they are linearly independent is the same as the probability that a random $n \times n$ matrix over $\mathsf{GF}(2)$ is invertible. In fact it is a well known result in discrete mathematics, that this probability is around 0.29 as n becomes large.

When all the κ_i's are not linearly independent, then we can not write the round keys in the middle r_2 rounds as linear expressions of the κ_i's. And if this happens, then naturally the attack as outlined in the previous subsection can not be applied. In that case how do you proceed with the attack?

1. Let us assume that for some $r_1,\ r_3$, the total rank of the $3 \cdot s \cdot (r_1 + r_3) \times n$ matrix containing the linear expressions (in terms of the master key) for all the keybits κ_i used in these

rounds be equal to λ. We have already seen that that when $3 \cdot s \cdot (r_1 + r_3) = n$, the probability that $\lambda = n$ is 0.29. The probability that $\lambda = t$ is given by the expression $2^{-n \cdot t} \prod_{i=0}^{t-1} \left(1 - \frac{2^i}{2^n}\right)$. Therefore the probability that $t \geq n-1,\ n-2,\ n-3$ is around 0.58, 0.77, 0.88 respectively (for large enough n).

2. In such an event the attacker should choose suitable values of $r_1,\ r_3$ such that the value of $\lambda = 3 \cdot s \cdot (r_1 + r_3)$.

3. Let $K = [\kappa_0, \kappa_1, \ldots, \kappa_{\lambda-1}]$ be the corresponding keybits whose linear expressions are linearly independent. Let K_1 be the subset of these keybits used in the first r_1 rounds, K_2 be the subset of these keybits used in the last r_3 rounds. Choose $K_3 = [\kappa_\lambda, \kappa_{\lambda+1}, \ldots, \kappa_{n-1}]$ as random linear expressions of the master key such that the expressions for $K_1,\ K_2,\ K_3$ are linearly independent. After this step, all round keybits can be written as linear expressions in $K_1,\ K_2,\ K_3$.

4. After guessing G, the vector of the middle $s \cdot r_2$ majority bits, the middle r_2 rounds become completely affine. Again if v is the vector that is the output of the first r_1 rounds, the output vector w of the first $r_1 + r_2$ rounds can be written as $w = L_G(v) + Q_G(K_1) + W_G(K_2) + E_G(K_3) + C_G$, where L_G, Q_G, E_G are linear functions and C_G is an n-bit constant.

5. Since w can be computed from the ciphertext backwards as $w = R_3^{-1}(\mathsf{ct}, K_2)$ So we have $R_3^{-1}(\mathsf{ct}, K_2) = L_G(v) + Q_G(K_1) + W_G(K_2) + E_G(K_3) + C_G$. Rearranging terms we have $R_3^{-1}(\mathsf{ct}, K_2) + W_G(K_2) = L_G(v) + Q_G(K_1) + E_G(K_3) + C_G$. Let us partition K_3 into two disjoint sets K_{31} and K_{32} so that the number of bits in $K_1 \cup K_{31}$ and $K_2 \cup K_{32}$ are almost same. We write $E_G(K_3) = E_G^1(K_{31}) + E_G^2(K_{32})$. Rearranging terms further we have

$$R_3^{-1}(\mathsf{ct}, K_2) + W_G(K_2) + E_G^2(K_{32}) = L_G(v) + Q_G(K_1) + E_G^1(K_{31}) + C_G$$

After this our meet-in-the-middle algorithm will proceed as follows.

1. Choose suitable values of $r_1,\ r_3$ such that the value of $3 \cdot s \cdot (r_1 + r_3) = \lambda$.

2. Choose $K_3 = [\kappa_\lambda, \kappa_{\lambda+1}, \ldots, \kappa_{n-1}]$ as random linear expressions of the master key such that the expressions for $K_1,\ K_2,\ K_3$ are linearly independent.

3. Partition K_3 into two disjoint sets K_{31} and K_{32} so that the number of bits in $K_1 \cup K_{31}$ and $K_2 \cup K_{32}$ are almost same.

4. Guess the vector G of the $s \cdot r_2$ majority values in the middle rounds. Find the functions $L_G, W_G, K_G, E_G^1, E_G^2$ and C_G. This step is done $2^{s \cdot r_2}$ times in the worst case.

   - For all possible values of $K_2 \cup K_{32}$, create a hash table LIST_2 indexed by the n-bit vector $R_3^{-1}(\mathsf{ct}, K_2) + W_G(K_2) + E_G^2(K_{32})$. We need around $2^{n/2}$ operations in this step.
   - For all possible values of $K_1 \cup K_{31}$, create a hash table LIST_1 indexed by the n-bit vector $L_G(v) + Q_G(K_1) + E_G^1(K_{31}) + C_G$. We need around $2^{n/2}$ operations in this step.

- Find a collision between LIST_1 and LIST_2.
- When a collision is found check if the majority bits are consistent with the guess of the key. If yes, this key is in fact the encryption key. Otherwise, try another guess of G.

Again the majority of the computational complexity is taken by the guessing of G and computing $R_3^{-1}(\text{ct}, K_2) + W_G(K_2) + E_G^2(K_{32})$ for each guess of $K_2 \cup K_{32}$ and $L_G(v) + Q_G(K_1 + E_G^1(K_{31})) + C_G$ for each guess of $K_1 \cup K_{31}$. This part takes $2^{s \cdot r_2} \cdot 2^{1+n/2}$. If $r_1 + r_3 = \lfloor \frac{n}{3s} \rfloor - \Delta$ then the complexity can be rewritten as $2^{s \cdot r + s \cdot \Delta - n/3 + n/2} = 2^{sr + s\Delta + n/6}$. For $r = 0.8\lfloor \frac{n}{s} \rfloor$, this complexity is around $2^{29n/30 + s\Delta}$. Thus the procedure becomes a valid attack if and only if $s\Delta < n/30$. Thus since Δ is at least 1 when the first and last keybits are not all linearly independent, the procedure does not work for all challenge instances when $s = 10$.

2.7 Improving Complexities using the 3-xor problem

The 3-xor problem in a nutshell is as follows: given 3 lists L_1, L_2, L_3 of binary strings over $\{0,1\}^n$, the task is to find 3 elements $x_1 \in L_1$, $x_2 \in L_2$, $x_3 \in L_3$ such that $x_1 \oplus x_2 \oplus x_3 = 0$. This problem has been extensively studied in the literature. Wagner studied in [Wag02], the generalized k-xor problem and showed that for the 4-xor problem if we have lists of size $2^{n/3}$ then a solution can be found in time $O(2^{n/3})$. However the 3-xor problem still required $O(2^{n/2})$ time using his approach. In [Nan15] a forgery attack was mounted against the COPA mode of operation requiring only $2^{n/3}$ encryption queries and about $2^{2n/3}$ time. This attack was later refined in [NS15], using an improved 3-xor algorithm, to $2^{n/2-\epsilon}$ queries and $2^{n/2-\epsilon}$)operations, for small ϵ. In [LS19], the authors attacked the 2-round Even Mansour algorithm using this problem with data and time both lower than 2^n. However, the algorithm we use was proposed by Joux [Jou09, Section 8.3.3.1], which is the best algorithm for the 3-xor problem to this day. A generalization for the above algorithm for variable sized lists was proposed in [BDF18], however since we will use lists of fixed size in this section, Joux's algorithm is more relevant here.

Before we discuss the details of the attack it is best to summarize the algorithm in a few words. We begin with the following lemma.

Lemma 3. *Given $n/2$ randomly generated vectors over $\{0,1\}^n$, then with high probability, they are linearly independent.*

Proof. The above probability is given by $p = 2^{-n^2/2} \cdot \prod_{i=0}^{n/2-1}(2^n - 2^i)$. For large n, this equals

$$p = \prod_{i=0}^{n/2-1}\left(1 - \frac{2^i}{2^n}\right) \approx 1 - \frac{\sum_{i=0}^{n/2-1} 2^i}{2^n} = 1 - \frac{2^{n/2}-1}{2^n} \approx 1 - 2^{-n/2}$$

□

The algorithm proceeds with 3 lists $L_1,\ L_2,\ L_3$ of size $2^{n/2}/\ell,\ 2^{n/2}/\ell,\ \ell^2$ respectively where $\ell = \sqrt{n/2}$. The list L_3 has $n/2$ random vectors which span at most a subspace of rank $n/2$. It is possible to choose vectors $\mathbb{B} = \{b_1, b_2, \ldots, b_{n/2},\ b_{n/2+1}, b_{n/2+2}, \ldots, b_n\}$ such that all vectors in L_3 belong to the subspace generated by $b_{n/2+1}, b_{n/2+2}, \ldots, b_n$. Now designate M to be the $n \times n$ binary matrix that changes the basis of all vectors in $L_1,\ L_2,\ L_3$ to $\mathbb{B}$. In fact, in the modified basis all elements in L_3 will begin with $n/2$ zeros. From the previous lemma, we know that the elements in L_3 are linearly independent with very high probability. In that case $b_{n/2+1}, b_{n/2+2}, \ldots, b_n$ can be simply taken as the elements of L_3 which ensures that in the modified basis the elements of L_3 have hamming weight exactly equal to 1, i.e. it has 1 in one of the positions from $n/2+1$ to n. In fact, if there exist 3 vectors $x_1 \in L_1,\ x_2 \in L_2,\ x_3 \in L_3$ such that $x_1 \oplus x_2 \oplus x_3 = 0$, then $Mx_1 \oplus Mx_2 \oplus Mx_3 = 0$ for any $n \times n$ binary matrix M. Once M is fixed, it can be used to transform L_1 and L_2. After this, all we need to do is to search for pairs of elements $(x_1, x_2) \in L_1 \times L_2$ such that $M \cdot x_1 \oplus M \cdot x_2$ equals 0 on the first $n/2$ bits, (and when all the vectors in L_3 are linearly independent we simply have to check if the sum has hamming weight 1) and this of course can be done efficiently in the following way.

1. After transforming all elements of L_3 in the new basis $\mathbb{B}$, insert the elements in hash table J.
2. After transforming all elements of L_1 in the new basis $\mathbb{B}$, insert the elements in hash table H indexed by first $n/2$ bits. All cells of the table should be able to hold multiple elements.
3. After transforming all elements of L_2 in the new basis $\mathbb{B}$, insert the elements in the same hash table H indexed by first $n/2$ bits. In fact, if any cell of H has more than one elements then their sum in the first $n/2$ bits must be 0. By standard randomness assumptions there will be $\frac{2^{n+1} \cdot 2^{-n/2}}{n} = \frac{2^{n/2+1}}{n}$ such pairs left whose sum needs to be tested for membership in L_3.
4. Assuming that testing for membership in J can be done in constant time, we need $\frac{2^{n/2+1}}{n}$ tests. One can see that most of the time L_3 is linearly independent and so testing for membership in L_3 can be done by simply checking whether the hamming weight of the full vector is 1, and whether it begins with $n/2$ zeros. In this case, it is neither necessary to change the basis of vectors in L_3 nor store them anywhere.

Since the complexity of preparing each list is $O(2^{n/2}/\sqrt{n/2})$ and around $O(2^{n/2+1}/n)$ membership tests are required the complexity of the algorithm is $O(2^{n/2+1}/\sqrt{n/2} + 2^{n/2+1}/n) \approx O(2^{n/2+1}/\sqrt{n/2})$. This gives a speedup of around $\sqrt{n/2}$ compared to the basic birthday algorithm of Wagner.

2.7.1 MITM on 2-round full S-box layer

The improved algorithm closely follows the one presented in Sec 2.6.1 earlier. The basic idea is still the same: this time we partition the key K into 3 sets $K_1 = \{k_0, k_1, k_2, \ldots, k_{m-1}\},\ K_2 =$

$\{k_m, k_{m+1}, k_{m+2}, \ldots, k_{2m-1}\}$ and $K_3 = \{k_{2m}, k_{2m+1}, \ldots, k_{n-1}\}$, where the value of m is given by $\lfloor \log_2(2^{n/2}/\sqrt{n/2}) \rfloor$, and so the size of K_3 is considerably smaller and only around $\lfloor \log_2(n/2) \rfloor$.

Our strategy will be, as before, to guess the s majority bits $\eta_1, \ldots, \eta_s$ at the input of the second round inverse S-boxes to linearize R_2 (and of course its inverse). Borrowing the terminology from Sec 2.6.1, where x denotes the n-bit input to the second round and $\mathsf{RK}_1, \mathsf{RK}_2$ denote the first, second round keys which are linear functions of the original key $K = \mathsf{RK}_0$, we have $R_1(\mathsf{pt} + \mathsf{RK}_0, \mathsf{RK}_1) = x$ and $R_2(x, \mathsf{RK}_2) = \mathsf{ct}$. Since after guessing the majority bits η_I the inverse of R_2 becomes linear, we can write each bit x_i of x as $x_i = A_i(K_1) + B_i(K_2) + C_i(K_3) + d_i, \ \forall \ i \in [0, n-1]$, where all A_i, B_i, C_i are linear functions and d_i is a constant.

Similarly in R_1, the set of keybits in K_1, K_2, K_3 can be partitioned in a manner so that they are not combined multiplicatively in the first round. Hence computing R_1 in the forward direction from the plaintext input it is possible to write each x_i as $f_i(K_1) + g_i(K_2) + h_i(K_3) + e_i, \ \forall \ i \in [0, n-1]$, where all f_i, g_i, h_i are quadratic functions and e_i is a constant. Equating these expressions we have $A_i(K_1) + B_i(K_2) + C_i(K_3) + d_i = f_i(K_1) + g_i(K_2) + h_i(K_3) + e_i$. Rearranging terms we have:

$$\underbrace{[A_i(K_1) + f_i(K_1) + d_i]}_{L_1} + \underbrace{[B_i(K_1) + g_i(K_1) + e_i]}_{L_2} + \underbrace{[C_i(K_3) + h_i(K_3)]}_{L_3} = 0$$

We see that, if 3 lists are enumerated for the terms in the square braces, then we arrive exactly at the scenario of the 3-xor problem. We need to find 3 elements form these lists that sum to 0. So our modified algorithm will be as follows:

1. Calculate the functional forms of f_i, g_i, h_i and e_i for all $i \in [0, n-1]$.
2. Guess the values $\eta_1, \ldots, \eta_s$. This step is done 2^s times in the worst case.
   - Compute A_i, B_i, C_i, d_i for all $i \in [0, n-1]$ using the guessed values.
   - For all possible values of K_3, create a hash table L_3 indexed by the n-bit vector $[C_i(K_3) + h_i(K_3)], \ \forall \ i \in [0, n-1]$. From here find the matrix M that would transform basis the basis $\mathbb{B} = \{b_1, b_2, \ldots, b_{n/2}, \ b_{n/2+1}, b_{n/2+2}, \ldots, b_n\}$ such that L_3 is spanned by $b_{n/2+1}, b_{n/2+2}, \ldots, b_n$. With high probability the list L_3 is linearly independent so that $b_{n/2+1}, b_{n/2+2}, \ldots, b_n$ can be taken to be the vectors in L_3. Multiply all vectors in L_3 by M and store in a hash table J. Note there are around $n/2$ steps here.
   - For all possible values of K_1, create a hash table L_1 indexed by the n-bit vector $M \cdot [A_i(K_1) + f_i(K_1) + d_i], \ \forall \ i \in [0, n-1]$. We need $2^{n/2}/\sqrt{n/2}$ operations in this step.
   - For all possible values of K_2, create a hash table L_2 indexed by the n-bit vector $M \cdot [B_i(K_1) + g_i(K_1) + e_i], \ \forall \ i \in [0, n-1]$. We need $2^{n/2}/\sqrt{n/2}$ operations in this step.

- In practice, 2 different hash tables are not necessary. We can instead use one single hash table H in which all elements of L_1, L_2 are inserted indexed by the first $n/2$ bits as explained in the previous subsection.
- For all pairs in $x_1, x_2 \in H$ which are in the same cell

  A: Discard if the sum is not in L_3. For most cases this can easily be verified by checking if the hamming weight of the sum is 1, i.e. if L_3 is linearly independent.
- Once a solution for K_1, K_2 and K_3 is found, check if the majority bits are consistent with the guess of the key. If yes, this key is in fact the encryption key. Otherwise try another guess of $\eta_1, \ldots, \eta_s$.

For each majority guess, the complexity of the attack is dominated by finding a collision between two lists of length $O(2^{n/2}/\sqrt{n/2})$. So the total complexity of the attack is $O(2^s \times 2 \cdot 2^{n/2}/\sqrt{n/2}) = O(n^{-1/2} \cdot 2^{s+n/2+1})$. This gives a speed up of around $\sqrt{n/2}$ over the attack in Section 2.6.1.

There are some further issues to be discussed. We ideally want the lists L_1 and L_2 of the same size, but it is often not possible due to the algebraic structure of LowMC. Since we have to partition the keybits such that the cardinality of each set should be a multiple of 3, it is not always possible to get lists of size $2^{n/2}/\sqrt{n/2}$, $2^{n/2}/\sqrt{n/2}$ and $n/2$. For $n = 129$ we have $n/2 = 64.5 \approx 2^6$, and so we can take K_3 to be the last 6 bits of the key, and K_1 and K_2 may contain the first 60 and the next 63 bits of the key respectively. In that case, the cost of preparing the lists is around $2^{60} + 2^{63} \approx 2^{63}$. The sum of the transformed vectors in L_1 and L_2 would need to be zero in the first $129 - 64 = 65$ bits and so after filtering $2^{60+63-65} \approx 2^{58}$ vector sums need to be tested for membership in L_3. So the total cost is around $2^{63} + 2^{60} + 2^{58} \approx 2^{63}$. Multiplying this by the 2^{43} times we need to guess majority bits, this comes to $2^{63+43} = 2^{106}$, which results in a speed up of factor 8 compared to the basic MITM in Section 2.6.1. For $n = 192$, we have $n/2 = 96 \approx 2^{6.58}$. The only feasible choice of the size of K_3 is again 6, which forces K_1 and K_2 to be of size 93 each. The cost of preparing lists is around $2^{93} + 2^{93} = 2^{94}$. However the number of pairs needed to be tested for membership in L_3 is $2^{93+93-(192-64)} = 2^{58}$. So the total complexity for list matching is around $2^{94} + 2^{58} \approx 2^{94}$. Multiplying by the number of majority guesses, we get the total complexity as $2^{64+94} = 2^{158}$ which also results in a speed up of 8 compared to Section 2.6.1. Similarly for $n = 255$, we have to take K_1, K_2, K_3 of sizes 123, 126, 6 respectively. A similar calculation yields the total complexity as $2^{85+126} = 2^{211}$ which results again in a speedup of 8 compared to the basic MITM.

2.8 Conclusion

In this work we describe attacks on instances of LowMC where the number of S-boxes is less than the security level, when we use only one plaintext/ciphertext pair. A cryptanalysis of this kind is important as it results in a forgery on the post-quantum signature scheme PICNIC.

Since our attacks are in the KPA/KCA scenario and since we use only one plaintext/ciphertext pair, it is not possible to apply traditional symmetric cryptanalytic techniques like differential, linear or any other higher order differential attacks. We begin by showing how to efficiently linearize the LowMC S-box by guessing only one single balanced quadratic expression in its input bits. We leverage this fact to present two types of attacks. First is a simple linearization attack where the attacker obtains a set of linear equations on the key bits relating the plaintext and ciphertext. The second is a meet in the middle attack, which takes advantage of the fact that in a single LowMC round, all key bits are not combined multiplicatively. We then show how to improve the attack on the 2-round full S-box layer variant of LowMC with the help of Joux's algorithm to solve the 3-xor problem.

3 Ruining a PICNIC, Act 2

In this chapter, we present improved cryptanalysis results on instantiations of LowMC block-cipher relevant to the PICNIC signature scheme. We build on the attacks presented in Chapter 2 and show how these attacks can be extended to more number of rounds both in the full, and partial S-box layer scenario. The personal contributions of this chapter are taken from joint work with Subhadeep Banik, Serge Vaudenay and Hailun Yan published at IACR-ASIACRYPT 2021 [BBVY21].

Structure of the Chapter: We begin this chapter by providing some additional related work in section 3.1. We proceed by providing a more precise complexity analysis of the linearization attack and its proof in section 3.2. Then, we present improved attacks on both **a)** the 2 and 3-round complete non-linear layer instance in Section 3.3, and **b)** the $0.8 \cdot \lfloor \frac{n}{s} \rfloor$ and $\lfloor \frac{n}{s} \rfloor$-round LowMC instance with partial non-linear layers in Section 3.4. We show that the attack complexity can be reduced if we perform the MITM in two separate stages: the first stage reduces the set of possible key candidates of a fraction of key bits to a smaller set. A second MITM stage is then performed on this reduced candidate set and the candidates in the remaining fraction of the key bits. This result shows that the combined computational complexity of the 2 attack stages is significantly lower than the complexities reported in [BBDV20]. Table 3.1 tabulates in detail the complexities of the attacks reported in this work and compares them to the corresponding complexities reported in [BBDV20]. In Section 3.5, we present some experimental results on LowMC instances with smaller blocksizes. This is done to prove that the attacks presented in Section 3.3, 3.4 can indeed be applied to full-size LowMC instances. Section 3.6 concludes the chapter.

3.1 Additional Related Work

The LowMC cryptanalysis challenge asked for cryptanalysis of several instances of LowMC (in which the blocksize and keysize are equal), with both partial and complete non-linear layers given only one plaintext and ciphertext pair. We presented the results from [BBDV20] in chapter 2 where we demonstrated how some parameterizations of LowMC do not provide

the intended security level. The main observation was that after guessing the value of any balanced quadratic Boolean function on the inputs of the LowMC S-box, the transformation becomes completely linear. We chose the 3-variable majority function for this purpose, but we show that any balanced quadratic function can be used. Using this fact, we showed various attacks on

A 2-round LowMC with complete non-linear layers.

B $0.8 \cdot \left\lfloor \frac{n}{s} \right\rfloor$-round LowMC with partial non-linear layers. Here n denotes the blocksize of the LowMC instance, and s denotes the number of S-boxes in each round.

In [BBDV20], we reported the attack complexities in the number of linear/quadratic expression evaluations. However, it is always preferable to have computational complexity reported in terms of the number of encryptions. We show that the best complexity of these attacks is equivalent to $\frac{n}{2r} \times 2^{rs}$ encryptions (r denotes the number of rounds used in the encryption), as will be discussed later in this chapter.

In [Din21], the authors showed an ingenious method of finding roots of multiple polynomial systems over GF(2). The n variables of the equation system are partitioned into two disjoint sets $y = y_0, y_1, \ldots, y_{m-1}$ and $z = z_0, z_1, \ldots, z_{p-1}$ (with $n = m + p$). It is argued that any random linear combination of the polynomials in the original equation system has an isolated solution with high probability, i.e. if $(\hat{y}, \hat{z})$ is an isolated solution then $(\hat{y}, z')$ is not a solution for all $z' \neq \hat{z}$. The authors then observed that all such isolated solutions could be recovered bit-by-bit by computing $p + 1$ partial sums for each candidate solution $\hat{y} \in \{0,1\}^m$. The first step is to randomly combine the original equation system into a system with a smaller number of equations whose solutions can be found by brute force. These solutions are then used to compute partial sums and construct a candidate solution of the original equation system. This generic method of solving equations works quite well if the algebraic degree of the system is small and so it was applied to attack 3, 4 and 5 round LowMC with complete non-linear layers for some specific block-lengths. However, the method can not be applied to LowMC instances with partial non-linear layers, since the number of rounds in such instances are generally much higher, and the degree of the internal state variables (as a function of the key) doubles every round. [LIM21] reports an algebraic attack on LowMC. However, the authors use the $n^{2.8}$ estimate (ignoring constant factors) to solve Gaussian elimination, to report the complexity of their attack. As such it is unclear if the complexity bounds they report are tight.

3.2 Linearization Attack

The starting point of the attack in [BBDV20] was lemma 2 which enabled us to linearize the LowMC S-box by guessing only one balanced quadratic expression on its input bits, e.g. the 3-bit majority function.

Using this lemma, the first attack we proposed in [BBDV20] used only the linearization tech-

nique to obtain affine equations relating plaintext and ciphertext. The idea is as follows. The values of the majority function at the input of all the S-boxes in the encryption circuit were guessed: this made the expression relating the plaintext and ciphertext completely linear in

Table 3.1: Summary of results. Note for the complexity is given in #Encryptions

Instance	n	s	r	Type of Attack	Recalculated Complexity	Reference
Full S-box layer	129	43	2	Linearization	2^{91}	[BBDV20]*
	192	64			2^{134}	
	255	85			2^{176}	
Partial S-box layer	128	1	$0.8 \times \lfloor \frac{n}{s} \rfloor$	Linearization	2^{102}	[BBDV20]*
	192	1			2^{153}	
	256	1			2^{204}	
Partial S-box layer	128	10	$0.8 \times \lfloor \frac{n}{s} \rfloor$	Linearization	2^{103}	[BBDV20]*
	192	10			2^{163}	
	256	10			2^{203}	
Full S-box layer	129	43	2	Equation solving	2^{102}	[Din21]**
			3		2^{108}	
			4		2^{113}	
Full S-box layer	192	64	2	Equation solving	2^{153}	[Din21]**
			3		2^{162}	
			4		2^{170}	
			5		2^{175}	
Full S-box layer	255	85	2	Equation solving	2^{204}	[Din21]**
			3		2^{216}	
			4		2^{226}	
			5		2^{232}	
Full S-box layer	129	43	2	2-Stage MITM	$\mathbf{2^{81}}$	Sec 3.3
	192	64			$\mathbf{2^{122}}$	
	255	85			$\mathbf{2^{164}}$	
Full S-box layer	129	43	3	2-Stage MITM	2^{123}	Sec 3.3
	192	64			2^{186}	
	255	85			2^{248}	
Partial S-box layer	128	1	$0.8 \times \lfloor \frac{n}{s} \rfloor$	2-Stage MITM	$\mathbf{2^{101}}$	Sec 3.4
	192	1			$\mathbf{2^{151}}$	
	256	1			$\mathbf{2^{202}}$	
Partial S-box layer	128	1	$\lfloor \frac{n}{s} \rfloor$	2-Stage MITM	$\mathbf{2^{125}}$	Sec 3.4
	192	1			$\mathbf{2^{189}}$	
	256	1			$\mathbf{2^{253}}$	
Partial S-box layer	128	10	$0.8 \times \lfloor \frac{n}{s} \rfloor$	2-Stage MITM	$\mathbf{2^{91}}$	Sec 3.4
	192	10			$\mathbf{2^{149}}$	
	256	10			$\mathbf{2^{188}}$	
Partial S-box layer	128	10	$\lfloor \frac{n}{s} \rfloor$	2-Stage MITM	$\mathbf{2^{111}}$	Sec 3.4
	192	10			$\mathbf{2^{179}}$	
	256	10			$\mathbf{2^{238}}$	

*Complexities recalculated and do not always match those reported in [BBDV20]

**[Din21] reports complexities in bit operations. We recalculate them in number of encryptions.

the key variables, i.e. of the form:

$$A \cdot [k_0, k_1, \ldots, k_{n-1}]^T = \mathsf{const}, \tag{3.1}$$

where A is an $n \times n$ matrix over $\mathsf{GF}(2)$. Thereafter the key could be found by using Gaussian elimination. A wrong key found by this method could be discarded by recalculating the encryption and checking if the given plaintext mapped to the given ciphertext.

The above method would work if the total number of S-boxes in the encryption circuit is strictly less than the size of the key in bits. This happens for **a)** 2-round `LowMC` with complete non-linear layers and **b)** $0.8 \times \left\lfloor \frac{n}{s} \right\rfloor$-round `LowMC` with partial non-linear layers. However, we pointed out 2 issues in this approach:

1. If the total number of S-boxes in the encryption circuit is t, then the algorithm requires in the worst case at least 2^t computations of the encryption function (for the verification of each computed candidate key). It additionally requires 2^t Gaussian elimination calculations.

2. For any guess of the majority values, the matrix A computed above may not necessarily be invertible. If the dimension of the kernel of the matrix A is d_A, then we can see that $O(2^{d_A})$ keys would satisfy any equation of the form $A \cdot K = \mathsf{const}$. Thus the verification would require running the verification for 2^{d_A} candidate keys.

We start by giving a closed-form expression of the complexity of the linearization algorithm in terms of the number of encryptions.

First of all, the expected number of solutions for the system $A \cdot [k_0, k_1, \ldots, k_{n-1}]^T = \mathsf{const}$ is 1 if the system is random. If const lies in the image of the linear transformation defined by A then the system has 2^{d_A} solutions, and it has 0 solutions otherwise. Now the probability that const lies in the image of A is exactly 2^{-d_A} and so the average number of solutions by Bayes theorem is $2^{d_A} \cdot 2^{-d_A} + (1 - 2^{-d_A}) \cdot 0 = 1$, and testing this solution costs us one encryption.

Multiplying an $n \times n$ matrix with an n-bit column vector requires n^2 bit operations. Every `LowMC` round therefore requires at least $2n^2$ bit operations (n^2 for computing the affine layer and another n^2 for generating the round key). Assuming calculation of the S-box layer can be done in linear time using a lookup table and also since key xor with state also takes linear time, the sum total of all the other bit operations in the round are linear in n. Suppressing these, the total bit operations required in performing a `LowMC` encryption is around $2rn^2$. Solving a system of linear equations by Gaussian elimination (GE) costs around n^3 bit operations which is equivalent to $\frac{n^3}{2rn^2} = \frac{n}{2r}$ encryptions.

Also note the computational complexity required to formulate the linear system of form $A \cdot [k_0, k_1, \ldots, k_{n-1}]^T = \mathsf{const}$. We argue that this is equivalent to n encryptions. After guessing the majority bits, the system becomes completely linear. Therefore finding the i-th column of

A and the i-th bit of const is equivalent to performing one encryption with the basis key vector $[0,0,\ldots,k_i,\ldots,0,0]$. Hence the result follows. Therefore the total computational complexity required to perform the attack using only linearization in terms of number of encryptions is

$$2^{rs} \text{ (Guessing majority bits)} \times [\ n \text{ (Formulating the linear system)} + \frac{n}{2r} \text{ (Solving the linear system)} + 1 \text{ (Testing one solution on average)}].$$

We can simplify this to $n \cdot 2^{rs}$ encryptions.

3.2.1 Improving complexity using Gray-Code based approach

The above complexity can be significantly improved if one were to make the majority guesses in a Gray-code-like manner. Recall that the encoding is defined as follows: $\textbf{Graycode}(i) = i \oplus (i \gg 1)$. An important observation is that, the hamming difference between $\textbf{Graycode}(i)$ and $\textbf{Graycode}(i+1)$ is always 1 for all values of i. The idea is instead of ordering the majority guesses in lexicographic order, we use the order defined by the Gray-code, i.e. in the i-th step the majority guess sequence is the binary string defined by the bits of $\textbf{Graycode}(i)$. When this is done the matrix A defined above, changes very little from iteration i to $i+1$. Thus having already constructed A in the i-th iteration, the corresponding construction in the $i+1$-th iteration can be done much faster and so the cost of formulating the linear system of equations defined by Eqn (3.1) can be amortized over all the majority guesses.

Let us state the algorithm formally. Let $M = m_0, m_1, \ldots, m_{s-1},\ m_s, m_{s+1}, \ldots, m_{2s-1}, \ldots, m_{(r-1)s}, m_{(r-1)s+1}, \ldots, m_{rs-1}$ be the rs majority guesses for the s number of S-boxes in each of the r rounds. Let M_i denote the value of the string M at the i-th iteration which we want to be equal to $\textbf{Graycode}(i)$. Let the linearized system of equations at the i-th iteration be denoted as $A_i \cdot k = c_i$. We want to determine how A_{i+1}, c_{i+1} relate with respect to A_i, c_i. Let $x \rightarrow Tx \oplus v$ be the linear map from $\{0,1\}^n \rightarrow \{0,1\}^n$ that is obtained as a result of linearizing the S-boxes in any single round with the majority value string Str (T is an $n \times n$ matrix and v is a n-element vector). Let $x \rightarrow T'x + v'$ be the corresponding map when the majority string is $\mathsf{Str} \oplus \mathbf{e}_t$ (here $\mathbf{e}_t$ denotes the t-th unit vector of length s and $0 \leq t < s$). Then we define $\Delta_t = T \oplus T'$ and $\lambda_t = v \oplus v'$, so that $\Delta_t x + \lambda_t$ denotes the change of linear map when the majority guess changes at the t-th S-box.

Let L_a denote the $n \times n$ matrix used in the linear layer in the a-th round (with $1 \leq a \leq r$). Also, let $\textbf{Graycode}(i) \oplus \textbf{Graycode}(i+1) = \mathbf{e}_j$ for some j (by slight abuse of notation $\mathbf{e}_j$ here denotes the j-th unit vector of length rs). If $j < s$, then it can be deduced that $A_i \oplus A_{i+1} = (\prod_{a=1}^{r} L_a) \cdot \Delta_j := B_j$ (say) and $c_i \oplus c_{i+1} = (\prod_{a=1}^{r} L_a) \cdot \lambda_j := b_j$. If $j \in [(u-1)s, us-1]$, which means that the change of majority guess occurs in the u-th round, then denote $j' = j - (u-1)s$. B_j is now defined as $A_i \oplus A_{i+1} = (\prod_{a=u}^{r} L_a) \cdot \Delta_{j'}$ and $b_j = (\prod_{a=u}^{r} L_a) \cdot \lambda_{j'}$. In fact, it is thus possible to precompute for all $j \in [0, rs-1]$ the matrix-vector pair (B_j, b_j) before the linearization step

begins. Thus the linearization attack can be restated as follows:

1. For all $j \in [0, rs-1]$ precompute the matrix-vector pair (B_j, b_j).
2. Compute A_0, c_0 and try to solve the system $A_0 \cdot k = c_0$ using GE.
3. For $i = 1 \rightarrow 2^n - 1$ do
   - The majority guess is $M_i = \textbf{Graycode}(i)$.
   - Let $\textbf{Graycode}(i) \oplus \textbf{Graycode}(i-1) = \mathbf{e}_j$.
   - Calculate $A_i = A_{i-1} \oplus B_j$ and $c_i = c_{i-1} \oplus b_j$.
   - Try to solve the system $A_i \cdot k = c_i$ using GE.

Since none of the B_j's are sparse matrices, we can not devise a quicker method of doing GE on A_i from the knowledge of steps involved in the GE of A_{i-1}. The additional complexity of constructing A_i, c_i at each step is given by a matrix and vector addition and so equal to $n^2 + n$ bit operations which roughly corresponds to $\frac{n^2+n}{2rn^2} \approx \frac{1}{2r}$ encryption operations. Thus if P denotes the cost involved in pre-computation (which is at most a polynomial in rs) then the total complexity of the method can be written as $P + 2^{rs} \cdot (\frac{n}{2r} + 1 + \frac{1}{2r}) \approx \frac{n}{2r} \cdot 2^{rs}$ encryptions which gives us an improvement of a factor of $2r$ over the naive linearization method of the previous subsection. We have recalculated the complexities in Table 3.1 using this expression.

3.3 2-stage MITM attack on 2-rounds with full S-box layer

In Section 2.6.1 we already showed that after guessing the majority bits of the second round and linearizing it the algebraic relation between the plaintext and ciphertext can be written as

$$f_i(K_1) + A_i(K_1) + c_i = g_i(K_2) + B_i(K_2) + d_i, \ \forall \ i \in [0, n-1]. \tag{3.2}$$

The functions $A_i,\ B_i$ are linear and $f_i,\ g_i$ are quadratic. It can be seen that for Equation (3.2) to hold we need not split K in such a way that K_1 and K_2 have approximately $n/2$ bits. We can, for example, also split K so that K_1 has around $n/3$ and K_2 has around $2n/3$ bits. The only condition that must be satisfied is that the sizes of K_1 and K_2 are chosen so that they are never mixed multiplicatively in the first round. It is easy to see that if we choose $t = |K_1|$ and $n - t = |K_2|$ to be multiples of 3 then this condition is automatically satisfied.

The important observation is that $f_i,\ g_i$ can be expressed as affine functions in an extension of the input of double size. This comes from the structure of the S-box: $S(x_0, x_1, x_2)$ is an affine function on $(x_0, x_1, x_2, x_0x_1, x_1x_2, x_2x_0)$. Let $\overline{f}_i, \overline{g}_i$ be the affine functions associated with $f_i,\ g_i$. Therefore the above set of equations can be written as

$$\overline{f}_i(\overline{K}_1) + A_i(\overline{K}_1) + c_i + d_i = g_i(K_2) + B_i(K_2), \ \forall \ i \in [0, n-1], \tag{3.3}$$

where if $K_1 = [k_0, k_1, k_2, \ldots, k_{3w-3}, k_{3w-2}, k_{3w-1}]$, we define

$$\overline{K}_1 = [k_0, k_1, k_2, k_0k_1, k_1k_2, k_2k_0, \ldots\ldots\ldots, k_{3w-3}, k_{3w-2}, k_{3w-1},$$
$$k_{3w-3}k_{3w-2}, k_{3w-2}k_{3w-1}, k_{3w-1}k_{3w-3}].$$

Since K_1 only has the first $t = 3w$ bits of the master key and so $\overline{K}_1$ is of size $6w$. Since $F_i = \overline{f}_i + A_i$ is an affine function over $\overline{K}_1$, the map $\phi : \overline{K}_1 \to [F_0, F_1, \ldots, F_{n-1}]$ can be seen as a linear code of length n and dimension $6w$. Let w be such that K_1 contains around $n/3$ key bits i.e. $w \approx n/9$ and hence K_2 contains the remaining $2n/3$ key bits. Since ϕ is seen as a linear code, let $\mathbf{G}$ be the corresponding generator matrix (of size $n \times 6w \approx n \times 2n/3$), which can be efficiently constructed from the algebraic forms of the functions F_i. Let $\mathbf{H}$ be the parity check matrix of the code (of size $(n-6w) \times n \approx n/3 \times n$). The parity check matrix is essentially obtained from the generator matrix by employing one Gaussian elimination. Define Con to be the vector $[c_0 + d_0, c_1 + d_1, \ldots c_{n-1} + d_{n-1}]^T$. The left side of Equation (3.3), when written in matrix notation for all $i = 0, 1, \ldots, n-1$ is essentially $\phi(\overline{K}_1) + \mathsf{Con}$. Therefore we have $\mathbf{H} \cdot [\phi(\overline{K}_1) + \mathsf{Con}] = \mathbf{H} \cdot [\mathbf{G}\overline{K}_1 + \mathsf{Con}] = \mathbf{H} \cdot \mathsf{Con} = e$ (say). This follows from the fact that since $\mathbf{G}$ and $\mathbf{H}$ are the generator and parity check matrices of a linear code, we must have $\mathbf{H} \cdot \mathbf{G} = 0$.

We can split K_2 into two halves K_{21} and K_{22} such that both halves contain approximately $n/3$ key bits each. Let's say $|K_{21}| = 3u$ and $|K_{22}| = n - 3w - 3u$ (our strategy would be to have $3u \approx n - 3w - 3u$ so that the halves are of equal size). We can rewrite $g_i(K_2) + B_i(K_2)$ as $g_i^1(K_{21}) + B_i^1(K_{21}) + g_i^2(K_{22}) + B_i^2(K_{22})$ for all $i \in [0, n-1]$, where g_i^j are quadratic and B_i^j are linear for $j = 1, 2$. Again this is possible if we take $|K_{21}|$ and $|K_{22}|$ to be multiples of 3, so that the bits of K_{21} and K_{22} after xor with the plaintext are input to different S-boxes. Due to the structure of `LowMC`, the quadratic terms from adjacent S-boxes do not combine multiplicatively after one round and so the separation into the 2 expressions is possible. Define the n-bit vectors:

$$M_1 = [g_0^1(K_{21}) + B_0^1(K_{21}), \ldots, g_{n-1}^1(K_{21}) + B_{n-1}^1(K_{21})]^T, \text{ and}$$
$$M_2 = [g_0^2(K_{22}) + B_0^2(K_{22}), \ldots, g_{n-1}^2(K_{22}) + B_{n-1}^2(K_{22})]^T.$$

One can see that, if Eqn (3.3) for $i = 0, 1, \ldots, n-1$, is written together as a vector equation, The right-hand side of the vector equation is essentially $M_1 + M_2$. We have already seen that the left-hand side of the vector equation when multiplied by $\mathbf{H}$ results in the vector $\mathbf{H} \cdot \mathsf{Con} = e$. Multiplying the right side of the vector equation by $\mathbf{H}$, we get the matrix equation:

$$\mathbf{H} \cdot (M_1 + M_2) = e, \Rightarrow \mathbf{H} \cdot M_1 = \mathbf{H} \cdot M_2 + e.$$

Pre-computation: In this phase we try and compute some expressions that remain constant over different majority guesses. We compute the following vectorial functions over all points over its input space: **(a)** $f_i(K_1)$, $\forall i \in [0, n-1]$ over input space of K_1 i.e $\{0,1\}^{3w}$, **(b)** $g_i^1(K_{21})$, $\forall i \in [0, n-1]$ over input space of K_{21} i.e $\{0,1\}^{3u}$ and **(c)** $g_i^2(K_{22})$, $\forall i \in [0, n-1]$ over input space

of K_{22} i.e $\{0,1\}^{n-3u-3w}$. Using Möbius transform the number of bit-operations required are

$$n \cdot \left(\frac{3w}{2} \cdot 2^{3w} + \frac{3u}{2} \cdot 2^{3u} + \frac{n-3u-3w}{2} \cdot 2^{n-3u-3w} \right).$$

This follows since any t-variable Boolean polynomial can be evaluated over all its input space using Möbius transform using $t \cdot 2^{t-1}$ bit operations.

1st MITM stage: As M_1 and M_2 only contain expressions on the key bits in the sets K_{21} and K_{22} respectively, we can conduct a first MITM stage in which we create 2 lists L_1, L_2. L_1 contains the $(n-6w)$, n-bit vector pairs $\mathbf{H} \cdot M_1,\ M_1$ for all 2^{3u} values of K_{21}. And similarly the list L_2 contains the $(n-6w), n$-bit vector pairs $\mathbf{H} \cdot M_2 + e, M_2$ for all $2^{n-3w-3u}$ values of K_{22}. We look for a collision in the $n-6w$ co-ordinates of these lists. We are expected to get around $2^{3u+(n-3w-3u)-(n-6w)} \approx 2^{3w}$ collisions. Thus in the process we get 2^{3w} key values for the key bit set $K_2 = (K_{21}, K_{22})$. For computing each entry in the list L_1 we do the following:

1. Compute the vectorial linear functions $B_0^1, B_1^1, \ldots, B_{n-1}^1$ over a given point k in K_{21}. Each such computation takes $|K_{21}| \cdot n = 3un$ bit operations.
2. Add to the corresponding precomputed vector $g_i^1(k)$, $\forall i \in [0, n-1]$. This requires n bit operations.
3. Multiply by $\mathbf{H}$. Each such computation takes $(n-6w) \cdot n$ bit operations.

This is computationally equivalent to $\frac{3un+n+(n-6w)n}{2rn^2} \approx \frac{3u+n-6w}{4n}$ of an encryption for $r = 2$. A similar argument holds for L_2. Hence the total computational cost incurred in this step is $\frac{3u+n-6w}{4n} \cdot 2^{3u} + \frac{2n-9w-3u}{4n} \cdot 2^{n-3w-3u}$ encryptions.

2nd MITM: Let us now turn to Eqn (3.2). The left side of this equation is defined over approximately the $3w$-bit set K_1 which can have 2^{3w} values in total. And we have just reduced K_2 to a set of 2^{3w} values. Thus the next MITM is making two more lists $L_3,\ L_4$ of size 2^{3w} each in the following way. L_3 contains all 2^{3w} n-bit vectors $[f_i(K_1) \oplus A_i(K_1) \oplus c_i \oplus d_i],\ \forall\ i \in [0, n-1]$ enumerated for all the 2^{3w} values of K_1. For all the 2^{3w} values of K_2 that have passed the previous MITM step we make the list L_4 containing the n-bit vector $[g_i(K_2) \oplus B_i(K_2)],\ \forall\ i \in [0, n-1]$. We now look for a collision between L_3 and L_4. On average we have $2^{3w+3w-n} = 2^{6w-n} < 1$ collisions. This means that the correct key K will necessarily by the output of one of these MITM steps for the correct guess of majority bits in the second round. For constructing L_3 we need to compute the n linear functions $A_i(K_1)$ over the $3w$-bit variable K_1 which by the previous logic, requires $3wn$ bit operations each and then n bit operations for addition to the precomputed vector $f_i(K_1)$. Populating L_4 requires computing $[g_i(K_2) \oplus B_i(K_2)]$ for all the K_2 that have passed the previous MITM step. However we can compute this vector by simply adding the $M_1,\ M_2$ vectors that have collided in the previous MITM stage. This stage therefore requires $\frac{3wn+n}{4n^2} \cdot 2^{3w} + \frac{n}{4n^2} \cdot 2^{3w} \approx \frac{3w}{4n} \cdot 2^{3w}$ encryptions. We are now ready to state the attack formally:

1. Calculate the functional forms of $f_i, g_i, \overline{f}_i, g_i^1, g_i^2$ and c_i for all $i \in [0, n-1]$.
2. Pre-compute $f_i(K_1)$, $g_i^1(K_{21})$, $g_i^2(K_{22})$, $\forall\, i \in [0, n-1]$ over their respective input spaces.
3. Guess the majority values $\eta_1, \ldots, \eta_s$ at the output of 2nd round S-box layer as in the previous attack. This step is done 2^s times in the worst case (note $s = n/3$).
   - Compute A_i, B_i, d_i for all $i \in [0, n-1]$ using the guessed values.
   - Compute the functions $F_i = \overline{f}_i + A_i$ for all $i \in [0, n-1]$.
   - Using the F_i's, construct the generator matrix $\mathbf{G}$.
   - Using Gaussian elimination, construct the parity check matrix $\mathbf{H}$.
   - Construct $\mathsf{Con} = [c_0 + d_0, c_1 + d_1, \ldots c_{n-1} + d_{n-1}]^T$, and $e = \mathbf{H} \cdot \mathsf{Con}$.
   - For all possible values of K_{21}, create a hash table L_1 indexed by the $(n - 6w)$-bit vector $\mathbf{H} \cdot M_1$.
   - For all possible values of K_{22}, create a hash table L_2 indexed by the $(n - 6w)$-bit vector $\mathbf{H} \cdot M_2 + e$.
   - Find all collisions between L_1 and L_2. Store all values of K_{21}, K_{22} extracted from the collision in a list L.
   - For all possible values of K_1, create a hash table L_3 indexed by the n-bit vector $[f_i(K_1) \oplus A_i(K_1) \oplus c_i \oplus d_i],\ \forall\ i \in [0, n-1]$.
   - For all values of $K_2 \in L$, create a hash table L_4 indexed by the n-bit vector $[g_i(K_2) \oplus B_i(K_2)],\ \forall\ i \in [0, n-1]$.
   - When a collision is found for K_1 and K_2 check if the majority bits are consistent with the guess of the key. If yes, this key is in fact the encryption key. Otherwise try another guess of $\eta_1, \ldots, \eta_s$.

Complexity Estimation: We first consider the time complexity. For each guess of $2^s = 2^{n/3}$ majority values, we have to perform a Gaussian elimination and 2 MITM steps. The cost of Gaussian elimination and the linear terms required to formulate A_i, B_i, d_i and pre-computation may be ignored in comparison with $2^{n/3}$. Hence the total time complexity for this attack is around

$$2^{n/3} \cdot \left(\frac{3u + n - 6w}{4n} \cdot 2^{3u} + \frac{2n - 9w - 3u}{4n} \cdot 2^{n-3w-3u} + \frac{3w}{4n} \cdot 2^{3w} \right). \tag{3.4}$$

For $w = u = n/9$, the above evaluates to $2^{n/3} \cdot ((\frac{1}{6} + \frac{1}{6} + \frac{1}{12}) \cdot 2^{n/3}) = \frac{5}{12} \cdot 2^{2n/3} \approx 2^{2n/3 - 1.26}$ encryptions.

Memory Complexity: In the first MITM stage, we created 2 lists L_1, L_2 which contain $(n - 6w)$, n-bit vector pairs for 2^{3u} possible values of K_{21} and $(n - 6w)$, n-bit vector pairs for $2^{n-3w-3u}$ possible values of K_{22}, respectively. As a matter of fact, in practice, 2 different lists are not necessary. We can instead insert each new vector of L_1 and L_2 into a single hash table.

The memory complexity here is $(2n-6w)\cdot(2^{3u}+2^{n-3w-3u})$ bits. In the second MITM stage, we create 2 more lists L_3, L_4, both containing 2^{3w} n-bit vectors. By similar logic, memory complexity here is thereby $2n\cdot 2^{3w}$ bits. The pre-computation part generates n-bit vectors over the input spaces of K_1, K_{21}, K_{22}. Hence the memory complexity here is $n\cdot(2^{3w}+2^{3u}+2^{n-3u-3w})$ bits. The total memory complexity for this attack is around

$$(2n-6w)\cdot(2^{3u}+2^{n-3w-3u})+2n\cdot 2^{3w}+n\cdot(2^{3w}+2^{3u}+2^{n-3u-3w}) \text{ bits.} \tag{3.5}$$

If we look at concrete parameters, for $n=129$ and $s=43$, we can choose the parameters in the following manner: we can choose $w=u=14$, which makes $|K_1|=42$ and $|K_2|=87$ and hence $|K_{21}|=42$ and $|K_{22}|=45$. The parity check matrix $\mathbf{H}$ is of size $(n-6w)\times n = 45\times 129$, which makes $\mathbf{H}\cdot M_1$ and $\mathbf{H}\cdot M_2+e$ both 45-bit vectors. After the first MITM stage the number of remaining candidates for K_2 is $\approx 2^{|K_{21}|+|K_{22}|-45}=2^{42}$. The complexity of the first MITM stage is thus $\frac{1}{6}\cdot(2^{45}+2^{42})\approx\frac{1}{6}\cdot 2^{45}\approx 2^{42.4}$ encryptions. The second MITM stage requires $\frac{1}{12}\cdot 2^{42}=2^{38.4}$ encryptions. Hence the total attack complexity is $2^s\cdot(2^{42.4}+2^{38.4})\approx 2^{85}$ encryptions and around 2^{53} bits of memory. This is lower than the linearization attack by a factor of 2^6 for this `LowMC` instance.

3.3.1 Extending attack to 3-rounds

The attack can be extended to 3-round `LowMC` in which we keep the basic character of the algorithm and run it by guessing the majority values of the last 2 rounds and linearizing both of them simultaneously. Hence a total of 2^{2s} values would need to be guessed in stead of 2^s. All other steps remain the same. Thus the computational complexity will be given by:

$$2^{2n/3}\cdot\left(\frac{3u+n-6w}{6n}\cdot 2^{3u}+\frac{2n-9w-3u}{6n}\cdot 2^{n-3w-3u}+\frac{3w}{6n}\cdot 2^{3w}\right).$$

This is so since encryption is now given by $2rn^2=6n^2$ bit operations. The memory complexity is essentially the same as in the 2-round attack. For $w=u=n/9$, the above evaluate of computational complexity is $2^{2n/3}\cdot((\frac{1}{9}+\frac{1}{9}+\frac{1}{18})\cdot 2^{n/3})\approx\frac{5}{18}\cdot 2^n$ encryptions, which is better than exhaustive search by a factor equal to approximately 2 bits. For $n=129$ and $s=43$, using the values $w=14$, $|K_1|=42$, $|K_{21}|=42$ and $|K_{22}|=45$, we get $\frac{1}{9}\cdot(2^{45}+2^{42})\approx 2^{41.8}$ encryptions for the first MITM. The second MITM requires $\frac{1}{18}\cdot 2^{42}\approx 2^{37.8}$ encryptions. The total complexity is therefore $2^{2s}\cdot(2^{41.8}+2^{37.8})\approx 2^{128}$ encryptions.

3.3.2 Speedup using Gray-Codes

There are 3 places in the above process where a speed-up may be applied using a Gray-code like approach.

1. By ordering the majority guesses in a Gray-code like manner as in Sec 3.2.1 so that the

affine expressions formed after linearizing the S-boxes can be generated more efficiently. But we have already seen that this does not result in significant speed-up when employed along with MITM.

2. Using a Gray-code like approach to do the pre-computations.

3. Using a Gray-code like approach to generate the values of the expressions that are inserted in the tables in each of the MITM stages. We will see how optimizing this stage results in significant speed-up.

There are several methods of evaluating an n-variable Boolean function over all the 2^n points of its input space, given its algebraic expression. One such method, as we have already seen is the Möbius transform which evaluates the function in-place by performing around $n \cdot 2^{n-1}$ bit operations. However the method we will use for this method is the Gray-code based approach suggested by [BCC+10] which finds all roots of a polynomial over GF(2) by evaluating it over all points of its input space by traversing the space in a Gray-code like manner. We start with the following theorem from [BCC+10].

Theorem 1. *[BCC+10] All the zeroes of a single multivariate polynomial f in n variables of degree d can be found in essentially $d \cdot 2^n$ bit operations (plus a negligible overhead), using n^{d-1} bits of read-write memory, and accessing n^d bits of constants, after an initialization phase of negligible complexity $O(n^{2d})$.*

We present a top-level overview of the approach used in this paper. Consider the derivative $\frac{\delta f}{\delta i} : \mathbf{x} \rightarrow f(\mathbf{x}+\mathbf{e}_i) \oplus f(\mathbf{x})$. Then for any vector $\mathbf{x}$, we have $f(\mathbf{x}+\mathbf{e}_i) = f(\mathbf{x}) \oplus \frac{\delta f}{\delta i}(\mathbf{x})$. If the algebraic degree of f is d then $\frac{\delta f}{\delta i}$ is of degree $d-1$. For our use case, f is either a quadratic or linear function. Meaning $\frac{\delta f}{\delta i}$ is an affine (or constant) function and can be represented by a vector $D_i \in (\mathrm{GF}(2))^n$ and a constant c_i. Now we have that,

$$c_i = \frac{\delta f}{\delta i}(0) = f(e_i) + f(0) \tag{3.6}$$

$$D_i[j] = f(e_i + e_j) + f(e_i) + f(e_j) + f(0) \tag{3.7}$$

Note that the $D_i[j]$ and c_i values only depend on the evaluation of f on unite vectors. The main idea of the optimization is to precompute and store these values and use them to evaluate the function on the rest of the space faster. Let us elaborate on how this is done. The first observation is that:

$$f(x + e_i) = f(x) + D_i \cdot x + c_i \tag{3.8}$$

Now when enumerating the input space in a gray-code order, for every two consecutive input values, they differ by a unit vector e_i. Hence, f can be evaluated on the later input using equation 3.8.

Employing this method for the case of our attack, for the pre-computation part, we can evaluate each t-variable quadratic Boolean function in 2^{t+1} bit-operations. As a result the pre-computation cost can be brought down to $2n \cdot \left(2^{3w} + 2^{3u} + 2^{n-3u-3w}\right)$ bit-operations. However, the pre-computation is not the most dominant term in the total computational cost, and so this gives only a slight improvement.

We now see how we can improve the complexity of the MITM stages by using this approach. As we only evaluate linear functions inside the iterations for each majority guess, since only 2^t bit-operations are required to evaluate any linear function, using the Gray-code approach we can accelerate this part considerably. Note that in L_1 we need to store both $\mathbf{H} \cdot M_1$ and M_1. To do this, we begin by computing the quadratic expressions each one of the n bits M_1 and then each of the $(n-6w)$-bits given by $\mathbf{H} \cdot M_1$. We use the Gray-code approach of [BCC+10], to evaluate these functions over all the points of their input domains. The number of bit operations required are therefore $n \cdot 2^{3u+1} + (n-6w) \cdot 2^{3u+1} \approx \frac{2n-6w}{2rn^2} \cdot 2^{3u+1}$ encryptions. Similarly the list L_2 would require around $\frac{2n-6w}{2rn^2} \cdot 2^{n-3u-3w+1}$ encryptions.

The lists L_3, L_4 are simpler to construct. For L_3 we need to compute the n linear functions $A_i(K_1)$ which requires $n \cdot 2^{3w}$ bit operations each and then add to the precomputed vector $f_i(K_1)$. Populating L_4, as before can be done by simply adding the M_1, M_2 vectors that have collided in the previous MITM stage. This stage therefore requires $\frac{2n}{2rn^2} \cdot 2^{3w} + \frac{n}{2rn^2} \cdot 2^{3w} \approx \frac{3n}{2rn^2} \cdot 2^{3w}$ encryptions. This reduces the main terms of the computational complexity to

$$T = 2^{\frac{(r-1)n}{3}} \cdot \left(\frac{n-3w}{rn^2} \cdot 2^{3u+1} + \frac{n-3w}{rn^2} \cdot 2^{n-3u-3w+1} + \frac{3n}{2rn^2} \cdot 2^{3w} \right) \text{ encryptions}$$

For $n = 129$, $r = 2$ and $u = w = 14$, we have $T = 2^{80.7}$ encryptions. For $n = 129$, $r = 3$ and $u = w = 14$, we have $T = 2^{123.2}$ encryptions. The memory complexity of this attack is the same as the attack in the previous sub-section plus the additional cost for storing tables required for fast Gray-code based evaluations. Using Theorem 1, this additional memory is $(3u)^2 \cdot (2n-6w) + (n-3u-3w)^2 \cdot (2n-6w) + (3w) \cdot n$ bits which is negligible when compared to the space occupied by the lists.

3.4 2-Stage MITM attack on partial S-box layers

In order to perform a MITM on the partial S-box layer instances of `LowMC`, we use a trick used in both [BBDV20, RST18] to transform some of the initial and final rounds so that the total number of different key bits involved in these rounds is $3s$ per round. The transformations are shown in Figures 2.2, 2.3 and are similar to the ones used in [RST18]. In fact the transform used in the backward direction (see Figure 2.3) is exactly same as the one used in [RST18, Figure 1]. The idea is that the affine layer and key addition are interchangeable. Since L is a linear function, we have $L(x) + K = L(x + L^{-1}(K))$ and similarly $L(x + K) = L(x) + L(K)$. Hence the key addition can be moved before or after the affine layer as required, by multiplying the round key by the appropriate matrix. Figure 2.2 further shows how to transform the first r_1

rounds. To mount this attack let us split the LowMC into 4 parts as shown in Figure 3.1:

1. First $a+b$ rounds which have been transformed as per Figure 2.2.
2. Final c rounds which have been transformed as per Figure 2.3.
3. The remaining $d = r-a-b-c$ rounds which lie in between.

Let the set of round key bits in the first a, b and the last c rounds be denoted as

$$K_a = [\kappa_0, \kappa_1, \ldots, \kappa_{3sa-1}],\ K_b = [\kappa_{3sa}, \kappa_{3sa+1}, \ldots, \kappa_{3sa+3sb-1}],\ \text{and}$$
$$K_c = [\kappa_{n-3sc}, \kappa_{n-3sc+1}, \ldots, \kappa_{n-1}].$$

Denote by K_{rem} the remaining $n-3s(a+b+c)$ key bits such that K_a, K_b, K_c, and K_{rem} are linearly independent expressions of the master key and so any key bit can be expressed as a linear function of them. It is important to notice that, we implicitly assume here that $n \geq 3s(a+b+c)$.

Let $X = [x_0, x_1, x_2, \ldots, x_{n-1}]$ be the output of the first a rounds, $W = [\omega_0, \omega_1, \ldots, \omega_{n-1}]$ be the output of the first $a+b$ rounds and $Y = [y_0, y_1, \ldots, y_{n-1}]$ be the input to the last c rounds as shown in Figure 3.1. Observe the middle b and $d = r-a-b-c$ rounds closely, as seen in Figure 3.2. Let us introduce $6b \cdot s$ new variables $U = [u_0, u_1, \ldots, u_{3bs-1}]$ and $Z = [z_0, z_1, \ldots, z_{3bs-1}]$ such that they represent the input and output bits of the $b \cdot s$ S-boxes in the middle b rounds. Our first aim is to find a linear expression relating the x_i's, y_i's and z_i's and the key bits. Let $D = [D_0, D_1, \ldots, D_{n-1}]$ be the output of the first of the b rounds (see Figure 3.2). Then we can write $D = \mathbf{Lin}_1(z_0, z_1, \ldots, z_{3s-1}, x_{3s}, x_{3s+1}, \ldots, x_{n-1})$, where $\mathbf{Lin}_1$ denotes a set of n affine functions. Similarly, if $E = [E_0, E_1, \ldots, E_{n-1}]$ is the output of the next round we can write E as a set of linear functions on $(z_{3s}, z_{3s+1}, \ldots, z_{6s-1}, D_{3s}, D_{3s+1}, \ldots, D_{n-1})$ which means that we can write $E = \mathbf{Lin}_2(z_0, z_1, \ldots, z_{6s-1}, x_{3s}, x_{3s+1}, \ldots, x_{n-1})$ as a set of linear functions on X and the first $6s$ z_i's. Iterating upto all the b rounds, it can be seen that W can be written as a set of linear functions on the entire Z and $x_{3s}, x_{3s+1}, \ldots, x_{n-1}$. Now if we guess the majority bits at the inputs of the following d rounds, they become completely linear. In that case Y itself becomes linear in W and K_a, K_b, K_c, K_{rem} (since the key bits used in these d rounds can be seen as linear expressions in K_a, K_b, K_c, K_{rem}). Hence we have

$$Y = \mathbf{Lin}(Z, x_{3s}, x_{3s+1}, \ldots, x_{n-1}, K_a, K_b, K_c, K_{rem}). \tag{3.9}$$

The above equation denotes a system of n affine equations (one for each bit in Y) in all the n bits of the key. Our aim is to get a reduced set of equations by somehow eliminating Z, K_b, K_{rem} from this set. The set $\Lambda = \{Z, K_b, K_{rem}\}$ comprises a total of $\theta = 3sb + 3sb + (n-3s(a+b+c))$ variables. Consider the system of n equations given in Equation 3.9. Apart from the θ variables the system has n (for Y) $+ (n-3s)$ (for X) $+ (3as+3cs)$ (for K_a, K_c) $= 2n+3(a+c-1)s$ variables. So the above system can be written in matrix notation as $\mathbb{M} \cdot \mathbf{v} = \mathbf{a}$, where $\mathbf{v}$ is the set of $2n+3(a+c-1)s+\theta = (3n+3sb-3s)$ variables, $\mathbb{M}$ is a matrix over GF(2) of size $n \times (3n+3sb-3s)$,

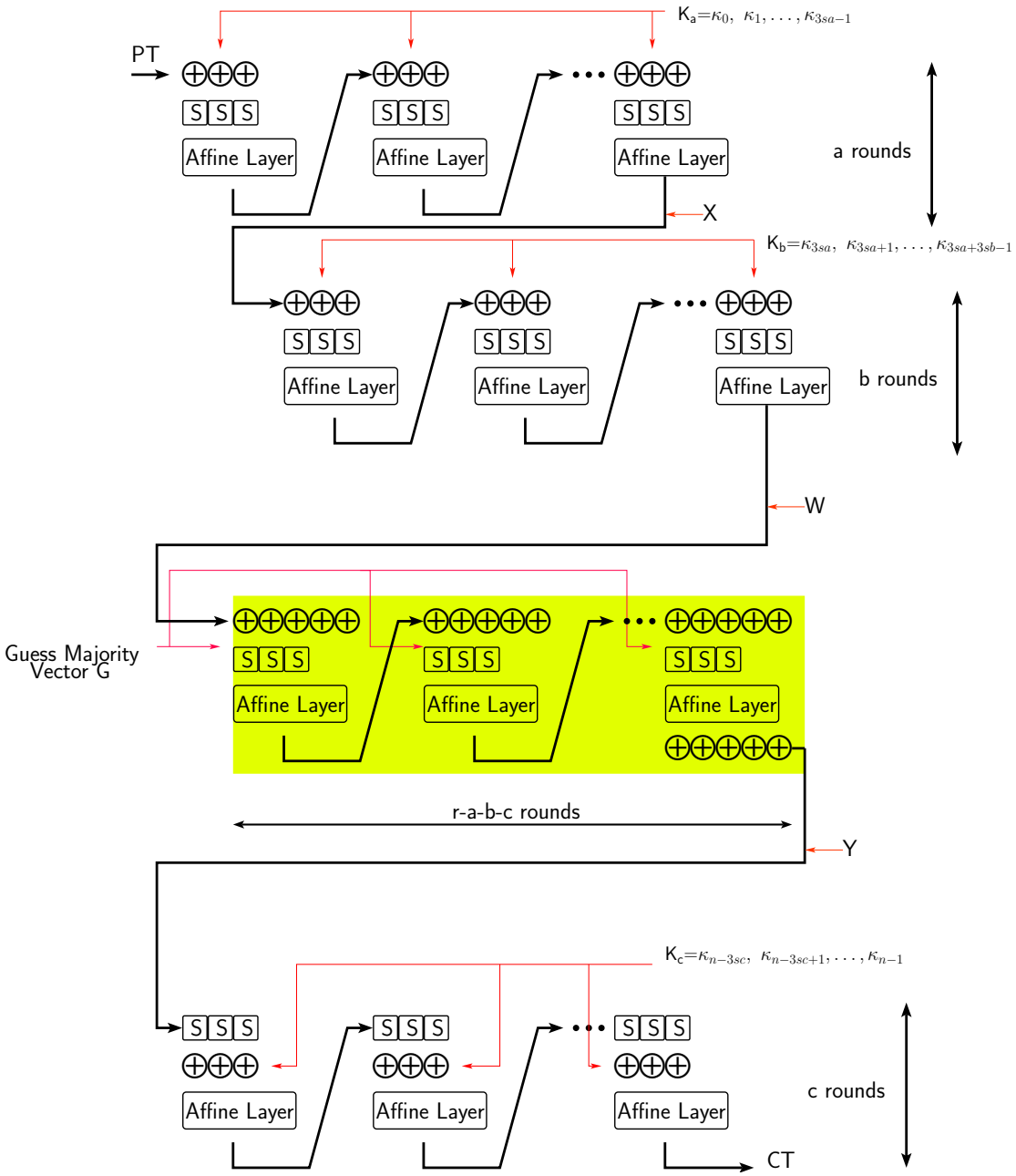


Figure 3.1: Splitting LowMC into 4 sections

and **a** is a constant vector. Rearrange **v** so that the variables in Λ are the first θ elements of **v**. Then we use Gaussian elimination to sweep out at least the first θ columns of $\mathbb{M}$. Then the last $n-\theta$ rows of the matrix would then have the entries in the first θ columns all equal to 0 and thus these are the linear equations in K_a, K_c, X, Y that we get from this process. Note we have a total of $n-\theta = 3sa+3sc-3sb$ equations of this form.

First MITM: The equations so obtained can be rearranged and written as $\mathbf{Aff}_1(K_a, X) = \mathbf{Aff}_2(K_c, Y)$, where $\mathbf{Aff}_1, \mathbf{Aff}_2$ are the set of $3sa+3sc-3sb$ affine functions on K_a, X and K_c, Y respectively, obtained above. We now state the first MITM step: the observation is that, if we guess the value of K_a, we can easily obtain the value of X by computing the forward a rounds from the plaintext. If we guess K_c we can similarly compute Y, by computing backward the last c rounds from the ciphertext. Hence for all the 2^{3sa} values of K_a we make the first list L_1 that contains all the $(3sa-3sb+3sc)$-bit vectors calculated from $\mathbf{Aff}_1(K_a, X)$. Similarly for all the 2^{3sc} values of K_c we make the second list L_2 that contains all the $3sa-3sb+3sc$-bit vectors calculated from $\mathbf{Aff}_2(K_c, Y)$. We look for collisions in the two lists. We can expect around $2^{3sa+3sc-(3sa-3sb+3sc)} = 2^{3sb}$ collisions. We store all the 2^{3sb} tuples (K_a, K_c) so obtained in a list L.

Second MITM: The second part of the attack focuses on getting an affine relation between

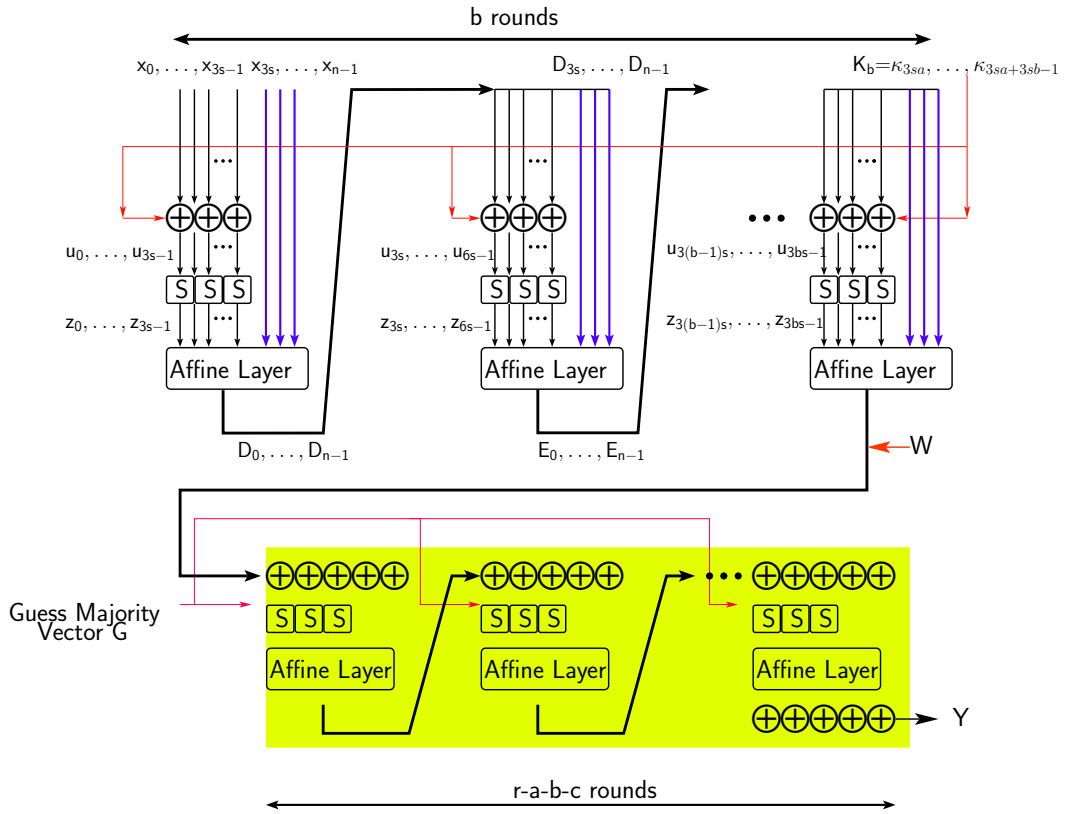


Figure 3.2: The middle $b+d$ rounds

U, Z and K_b. From Figure 3.2, we can see that $u_i = x_i + \kappa_{3sa+i}$, $\forall i \in [0, 3s-1]$. For the second round we have

$$\begin{aligned} u_{3s+i} &= D_i + \kappa_{3sa+3s+i}, \ \forall \ i \in [0, 3s-1] \\ &= \mathbf{Lin}_{1,i}(z_0, \ldots, z_{3s-1}, x_{3s}, \ldots, x_{n-1}) + \kappa_{3sa+3s+i}, \ \forall \ i \in [0, 3s-1] \end{aligned}$$

where $\mathbf{Lin}_{1,i}$ is the i-th linear function of $\mathbf{Lin}_1$ described above. The above holds since we have already seen that all D_i's are linear functions in $(z_0, \ldots, z_{3s-1}, x_{3s}, \ldots, x_{n-1})$. Similarly for the third round we have

$$\begin{aligned} u_{6s+i} &= E_i + \kappa_{3sa+6s+i}, \ \forall \ i \in [0, 3s-1] \\ &= \mathbf{Lin}_{2,i}(z_0, \ldots, z_{6s-1}, x_{3s}, \ldots, x_{n-1}) + \kappa_{3sa+6s+i}, \ \forall \ i \in [0, 3s-1] \end{aligned}$$

where $\mathbf{Lin}_{2,i}$ is similarly the i-th linear function of $\mathbf{Lin}_2$. Iterating over all the b rounds we can write the vector equation, $U = K_b + \ \mathbf{P}(Z, x_{3s}, \ldots, x_{n-1})$, where $\mathbf{P}$ denotes the set of $3bs$ linear expressions obtained by putting together the linear expressions $\mathbf{Lin}_{1,i}, \mathbf{Lin}_{2,i}$ etc. We can now replace K_b in Equation (3.9) to get

$$\begin{aligned} Y &= \mathbf{Lin}(Z, x_{3s}, x_{3s+1}, \ldots, x_{n-1}, K_a, U + \ \mathbf{P}(Z, x_{3s}, \ldots, x_{n-1}), K_c, K_{rem}) \\ &= \mathbf{Lin}'(Z, x_{3s}, x_{3s+1}, \ldots, x_{n-1}, K_a, U, K_c, K_{rem}). \end{aligned}$$

This time we eliminate K_{rem} from the above set of linear equations using the same Gaussian elimination method as in the previous stage. There are $n - 3s(a+b+c)$ variables in K_{rem} that we eliminate, which leaves us with $3s(a+b+c)$ equations in $Z, x_{3s}, x_{3s+1}, \ldots, x_{n-1}, K_a, U, K_c$. We can rearrange the terms in the equation to get $\mathbf{Aff}_3(Z, U) = \mathbf{Aff}_4(X, K_a, K_c)$, where $\mathbf{Aff}_3, \mathbf{Aff}_4$ are a set of $3s(a+b+c)$ affine functions on Z, U and K_a, K_c, X respectively.

In fact, if Z is guessed, one can compute U since the S-box is bijective, and we have already seen that guessing K_a lets us compute X by computing the a forward rounds from the plaintext. Thus in the next MITM stage we make 2 lists L_3, L_4. In L_3 we store the $3s(a+b+c)$-bit vector given by the expressions $\mathbf{Aff}_3(Z, U)$ for each of 2^{3bs} values of Z. In L_4 we store the $3s(a+b+c)$-bit vector given by the expressions $\mathbf{Aff}_4(X, K_a, K_c)$ for each of 2^{3bs} values of (K_a, K_c) in L. We again look for collisions in the 2 lists. The expected number of collisions is $2^{3bs+3bs-3s(a+b+c)} = 2^{3sb-3sa-3sc}$. However the correct value of the key K_a, K_c is guaranteed to be the outcome of the collision finding stage for the correct guess of the majority values.

Once we get a candidate solution K_a, K_c, Z, U we can compute the vectors X, Y by computing the a, c rounds forwards/backwards from the plaintext/ciphertext. We can then compute $K_b = U + \ \mathbf{P}(Z, x_{3s}, \ldots, x_{n-1})$. As we know the majority of the inputs of the S-boxes in $r-a-b-c$ middle rounds, we can solve an affine equation of form $\mathbf{Aff}_{rem}(W, K_{rem}) = Y$ to recover the value of K_{rem}, which was the only part of the key which remained unknown. After this one can check if the key so obtained produces the required majority values guessed at the beginning. If not the attacker can restart the process with another set of majority values. The expected number of such checks is around $2^{s(r-a-b-c)+3sb-3sa-3sc} = 2^{rs-4sa-4sc+2sb}$. We formally state

the attack:

1. Separate the first $a+b$ and last c rounds of the cipher
2. Denote the output of the first a rounds by X, the output of the b rounds by W and the input of the last c rounds by Y.
3. Denote the inputs/outputs of the S-boxes in the b rounds by U/Z
4. Guess majority bits of the inputs of the S-boxes of $r-a-b-c$ middle rounds.
5. For every majority guess do:

   First MITM:

   - Compute the relation $Y = \mathbf{Lin}(Z, x_{3s}, \ldots, x_{n-1}, K_a, K_b, K_c, K_{rem})$
   - Eliminate K_b, K_{rem}, Z from the relation and form and equation of form $\mathbf{Aff}_1(K_a, X) = \mathbf{Aff}_2(K_c, Y)$.
   - By exhausting all possible values of K_a keep a list of $\mathbf{Aff}_1(K_a, X)$, where X is computed knowing K_a and plaintext pt.
   - Try all possible values of K_c and find collisions between $\mathbf{Aff}_2(K_c, Y)$ and the list computed in the previous step. Keep a list L of (K_a, K_c) values satisfying the condition.

   Second MITM:

   - Compute the relation $Y = \mathbf{Lin}'(Z, x_{3s}, x_{3s+1}, \ldots, x_{n-1}, K_a, U, K_c, K_{rem})$ by replacing K_b.
   - Eliminate K_b, K_{rem} to get a relation of form $\mathbf{Aff}_3(Z, U) = \mathbf{Aff}_4(X, K_a, K_c)$.
   - For every pair (K_a, K_c) in the list L, compute $\mathbf{Aff}_4(X, K_a, K_c)$.
   - For every possible value of Z, compute $\mathbf{Aff}_3(Z, U)$, where U can be computed efficiently from Z, and look for occurrence with $\mathbf{Aff}_3(Z, U)$ in the list from the previous step.
   - For every (K_a, K_c, Z, U) satisfying the relation, compute K_b, W, Y as shown before.
   - Linearize the middle $r-a-b-c$ rounds using the majority guess and compute K_{rem} from $\mathbf{Aff}_{rem}(K_{rem}, K_a, K_b, K_c, W) = Y$.
   - After the entire key is found, check if they result in the same majority values assumed at the beginning of the attack or else retry with another set of majority values.

Complexity Estimation: Before we state our analysis to calculate the computational complexity, let us state a few observations:

1. The number of variables on the right side of Equation (3.9) is $2n+3sb-3s$. Hence using the basis vector logic, forming Equation (3.9) is equivalent to $2n+3sb-3s$ encryptions limited to $r-a-c$ rounds, hence equivalent to $(2n+3sb-3s)\cdot\frac{(r-a-c)}{r}$ encryptions.

2. For the first MITM, eliminating $\theta = n-3s(a-b+c)$ variables in an $n\times(3n+3sb-3s)$ matrix using the sweeping out method costs around $\frac{n\cdot\theta\cdot(3n+3sb-3s)}{2rn^2}$ encryptions.

3. Computing U from X and K is equivalent to the encryption of $2n$ base vectors (for the n bits of X and the n bits of K) in b rounds instead of r. So, this costs $2n\cdot\frac{b}{r}$ encryptions

4. For the 2nd MITM, eliminating $3sb$ (K_b) and $n-3s(a+b+c)$ (K_{rem}) variables in a $n\times(3n+6sb-3s)$ matrix requires $\frac{(n-3s(a+c))\cdot n\cdot(3n+6sb-3s)}{2rn^2}$ encryptions.

5. Solve the system of linear equations to get K_{rem} from $\mathbf{Aff}_{rem}(K_{rem},K_a,K_b,K_c,W)=Y$. This requires one Gaussian Elimination which is equivalent to $\frac{(n-3s(a+b+c))^3}{2rn^2}$ encryptions.

Both MITM steps should be done for each majority guess for the middle rounds, hence should be repeated $2^{s(r-a-b-c)}$ times. To evaluate $\mathbf{Aff}_1(K_a,X)$ we need to evaluate the first a encryption rounds to get X from the plaintext. Thereafter we evaluate $(3sa-3sb+3sc)$ linear expressions in $(3sa+n)$ bits of $K_a,\ X$, which requires around $(3sa+3sb-3sc)\cdot(3sa+n)$ bit-operations. Similarly to evaluate $\mathbf{Aff}_2(K_c,Y)$ we need to evaluate the last c decryption rounds to get Y from the ciphertext, followed by evaluation of linear expressions that take $(3sa+3sb-3sc)\cdot(3sc+n)$ bit-operations. Hence the first MITM takes time equivalent to $T_1=\left(\frac{a}{r}+\frac{(3sa-3sb+3sc)\cdot(3sa+n)}{2rn^2}\right)\cdot 2^{3sa}+\left(\frac{c}{r}+\frac{(3sa-3sb+3sc)\cdot(3sc+n)}{2rn^2}\right)\cdot 2^{3sc}$ encryptions. The number of pairs stored in the first MITM is around 2^{3sb} as mentioned before.

Later on we replace K_b in the linear equation and eliminate K_b,K_{rem}, this can also be seen as a matrix multiplication followed by a Gaussian elimination. Next we compute the values of $\mathbf{Aff}_3(Z,U)$ and $\mathbf{Aff}_4(X,K_a,K_c)$ having values of K_a,K_c and Z. Computing the value of U from Z takes time less than required in the b encryption rounds. Thereafter, evaluating $3s(a+b+c)$ linear expressions in $6bs$ bits requires $3s(a+b+c)\cdot 6bs$ bit-operations. Again for $\mathbf{Aff}_4$ computing X from K_a requires evaluating the first a encryption rounds. Then evaluation of linear expressions requires $3s(a+b+c)\cdot(3sa+3sc+n)$ bit-operations. Hence the 2nd MITM takes $T_2=\left(\frac{b}{r}+\frac{(3sa+3sb+3sc)\cdot(6bs)}{2rn^2}+\frac{a}{r}+\frac{(3sa+3sb+3sc)\cdot(3sa+3sc+n)}{2rn^2}\right)\cdot 2^{3sb}$ encryptions. The expected number of collisions in this procedure is $2^{3sb-3sa-3sc}$ which the attacker needs to filter whenever it is greater than 1. Hence the total complexity of the attack is estimated as:

$$
\begin{aligned}
2^{s(r-a-b-c)}\times\Big[\ & T_1+T_2\ (\text{The 2 MITMs})\ +(2^{3s(b-a-c)})\ (\text{Filter Solutions})+\\
& (2n+3sb-3s)\cdot\frac{(r-a-c)}{r}+\frac{n\cdot\theta\cdot(3n+3sb-3s)}{2rn^2}+\\
& 2n\cdot\frac{b}{r}+\frac{(n-3s(a+c))\cdot n\cdot(3n+6sb-3s)}{2rn^2}+\\
& \frac{(n-3s(a+b+c))^3}{2rn^2}\Big].
\end{aligned}
$$

As n and s go to infinity, the optimal parameters become $a = b = c = 1$ and the asymptotic complexity is equivalent to $\frac{4}{r} * 2^{sr}$, which is an improvement by a factor $n/8$ compared to the linearization attack. When s remains small (e.g. $s = 1$), the optimal parameters can be larger. With $a = b = c = \frac{\log_2(2n)}{3s}$, the complexity is asymptotically $\frac{4\log_2(n)}{3sr} \cdot 2^{sr}$. If we take $sr = n$, this is better than exhaustive search by a factor $\Omega\left(\frac{n}{\log(n)}\right)$. The memory complexity is dominated by the space required for the 2 MITM stages. It can be seen that the total memory complexity in bits can be computed as

$$(3sa - 3sb + 3sc) \cdot (2^{3as} + 2^{3cs}) + (3sa + 3sb + 3sc) \cdot 2^{3bs+1}.$$

For the $\lfloor \frac{n}{s} \rfloor$-round instances, we get the following results. For $n = 128,\ s = 1, r = 128$, if we take $a = b = c = 5$, we get the total complexity around 2^{125} encryptions with 2^{22} bits of memory. For $n = 128,\ s = 10, r = 12$, if we take $a = b = c = 1$, we get the total complexity around 2^{119} encryptions with 2^{38} bits of memory. For the $0.8 \times \lfloor \frac{n}{s} \rfloor$-round instances, we get the following results. For $n = 128,\ s = 1, r = 103$, if we take $a = b = c = 5$, we get the total complexity around 2^{101} encryptions. For $n = 128,\ s = 10, r = 10$, if we take $a = b = c = 1$, we get the total complexity around 2^{99} encryptions. The memory complexity is the same as the corresponding $\lfloor \frac{n}{s} \rfloor$-round attacks.

3.4.1 Speed-up using Gray-Codes

We emphasize that the technique outlined in [BCC+10] to evaluate a function over all points of its input domain, works best for linear or quadratic functions. As such, it is best to employ the attack when the set of functions for which we want to evaluate over the input space is quadratic/linear. This is only possible if we restrict $a = c = 1$. Let us see why. The first MITM procedure finds a collision between two lists using the equation $\mathbf{Aff}_1(K_a, X) = \mathbf{Aff}_2(K_c, Y)$. Notice that, thus far, X (rep. Y) has been computed from the plaintext (resp. ciphertext) by guessing K_a (resp. K_c) and evaluating the first a rounds in the forward direction (resp. last c rounds in the backward direction). In order to apply Gray-code-based speed-up we need to express X and Y as functions of K_a and K_c. These functions happen to be quadratic only when $a = c = 1$. This condition automatically ensures that in the second MITM equations are also quadratic. This is true since the second MITM essentially equates $\mathbf{Aff}_3(Z, U) = \mathbf{Aff}_4(X, K_a, K_c)$, and we know that the relation between $U,\ Z$ is quadratic since these are the input-output bits of the LowMC S-box in the middle b rounds. However, unlike in the MITM for the complete non-linear layers, there is no pre-computation in the first MITM that helps us reduce the steps in the second MITM. $\mathbf{Aff}_4(X, K_a, K_c)$ needs to be only evaluated for the 2^{3sb} pairs of K_a, K_c that survive the 1st MITM. However, to employ Gray-code-based speed up we need to evaluate $\mathbf{Aff}_4$ over all points of its input space. We could split $\mathbf{Aff}_4$ into $\mathbf{Aff}_5(K_a, X) + \mathbf{Aff}_6(K_c)$ and then evaluate each of the $\mathbf{Aff}_5$ and $\mathbf{Aff}_6$ separately. Thus the time required for the first MITM would be $T_{G_1} = \frac{3sa-3sb+3sc}{2rn^2} \cdot (2^{3as+1} + 2^{3sc+1})$ encryptions. The 2nd MITM requires $T_{G_2} = \frac{3sa+3sb+3sc}{2rn^2} \cdot (2^{3bs+1} + 2^{3as+1} + 2^{3sc})$ encryptions.

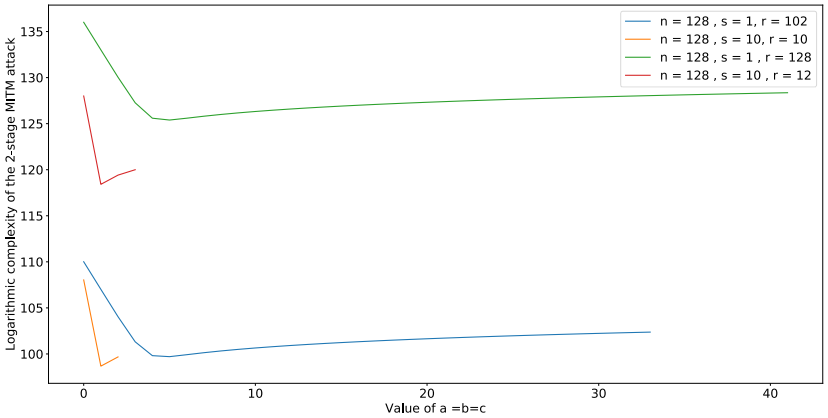


Figure 3.3: The base 2 logarithm of the complexity of the 2-stage MITM attack when $n = 128$ and $s = 1, 10$, for $n = 0.8 \times \lfloor \frac{n}{s} \rfloor$, when a, b, c are kept equal and varied.

It only makes sense to employ Gray-codes if $T_{G_1} + T_{G_2} < T_1 + T_2$. For $s = 1$, the optimal values of a, b, c are considerably higher and it does not make sense to attempt the Gray-code speed-up using this algorithm. In fact even if we attempt to use this method by forcing $a = b = c = 1$, the complexity is many times higher. Intuitively this makes sense, if a, c and s are both 1 then the lists require exhaustive search over only $3as = 3sc = 3$ variables, for which employing even a non-Gray-code approach would take only 2^3 function evaluations. However when $s = 10$, using such Gray-codes to execute the MITM stages is beneficial. For $n = 128$, $s = 10, r = \lfloor \frac{n}{s} \rfloor = 12$, if we take $a = b = c = 1$, we get the total complexity around $2^{110.6}$ encryptions which is better than the previous estimate by a factor of around 2^9. For $r = 0.8 \times \lfloor \frac{n}{s} \rfloor = 10$ using the same parameters we get the total complexity around $2^{90.8}$ encryptions which again outperforms the previous estimate by a factor of around 2^8.

3.5 Experimental Results

In this section we present experimental data to showcase how our new attacks stack up in comparison to the attacks proposed in [BBDV20] on instances of `LowMC` with smaller blocksizes. Our results indicate that for all instances targeted in our work, there is a significant speedup compared to the previous attacks. Moreover, we provide experimental evidence that our attacks successfully recover the key with a better complexity than exhaustive search for both 3-round with full S-box layer and n/s-round with partial S-box layer variants.

All the attacks and variants of the encryption function were implemented in Sage and ran on an Intel Xeon E5-2680 processor with 256 GB of memory. Each attack was run for several randomly generated instances. The complexity figures are reported by computing the base 2 logarithm of the amount of time taken by the attack, divided by the amount of time one

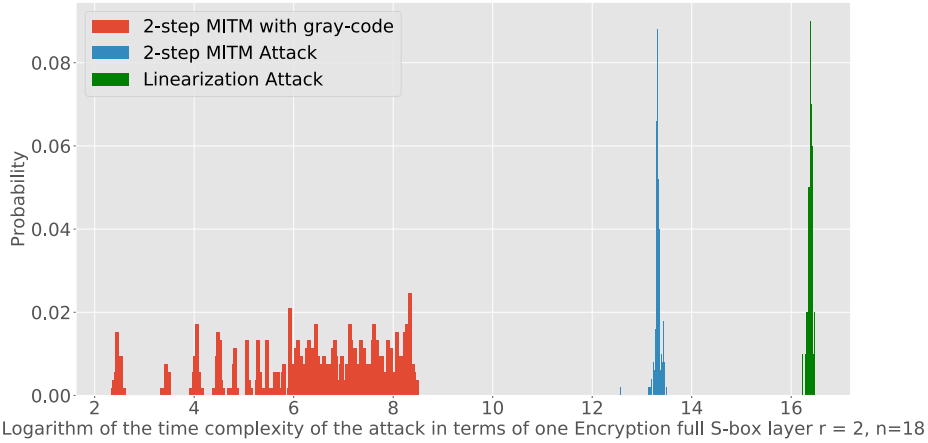


Figure 3.4: The histogram of base 2 logarithm of the time complexity of all linearization, 2-stage MITM and 2-stage MITM with gray-code enumeration attacks for $n = 18$, $s = 6$, $r = 2$, in terms of the time it takes to perform a single encryption with the same key, the same affine layers, and the same key update functions.

encryption takes.[1]

Full S-box Layer: For the 2-round full S-box layer variant of the cipher, we implemented all three Linearization, 2-step MITM and 2-step MITM with gray-code enumeration attacks for $n = 18$. The results are presented in Figure 3.4. On average, the linearization attack required $2^{16.38}$ encryptions to recover the key, where as the 2-stage MITM, and the gray code enumeration attacks required $2^{13.31}$ and $2^{6.42}$ encryptions to yield a solution respectively.

We also implemented the attack using Gray-code enumeration for 3-round variants of block size 12. Figure 3.5 show cases the complexity of this attack for several randomly generated samples. Our experimental results indicate that the 3-round variant of this attack yields a solution faster than exhaustive search for all the samples we ran the attack for and the average complexity of our experiments was $2^{5.88}$ encryptions for $n = 12, s = 4, r = 3$.

Partial Non-Linear Layer: For the partial S-box layer variant of the cipher with the number of rounds equal to $r = \lfloor \frac{n}{s} \rfloor \times 0.8$, we implemented the 2-stage MITM attack described in section 3.4, the linearization method described in [BBDV20] and in addition, the special case gray-code enumeration attack described at the end of 3.4. For $n = 16$, $s = 1$ and $r = 12$ the linearization attack yielded a complexity of $2^{10.29}$ encryptions, and the two-step MITM and the gray-code enumeration attacks yielded a solution in $2^{8.46}$ and $2^{8.50}$ encryptions respectively.

[1] The source code of the attacks can be found at https://gitlab.epfl.ch/barooti/lowmc-challenge-round-3

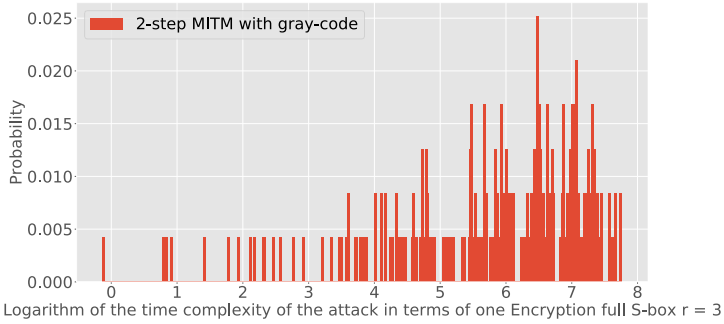
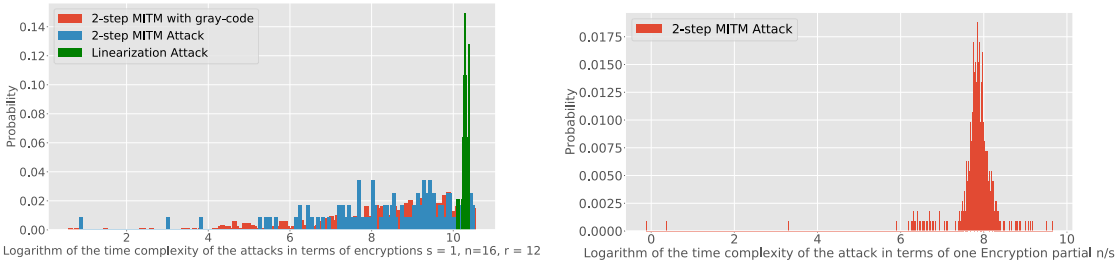


Figure 3.5: The histogram of base 2 logarithm of the time complexity of the gray-code enumerated 2-stage MITM attack for $n = 12$, $s = 4$, $r = 3$, in terms of the time it takes to perform a single encryption



(a) (b)

Figure 3.6: (a) The logarithm of the complexity of 2-step MITM, 2-step MITM with gray-code enumeration and linearization attacks for the partial S-box layer variant with parameters $n = 16$, $s = 1$ and $r = 12$, (b) The logarithm of the complexity of the two-step MITM attack for $n = 12, s = 1, r = 12$.

For the 2-step MITM attack, we ran the experiments for 3 instances of $a = b = c = 1$, $a = b = c = 2$ and $a = b = c = 3$. According to our experimental results, the best performance was when $a = b = c = 1$. The results of the 3 attacks are demonstrated in Figure 3.6a, and it is evident that both our new attacks are significantly faster than the linearization method.

We also experimented the attack for $n = 12$, $s = 1$ and $r = n/s = 12$ and $a = b = c = 1$. According to our experimental results demonstrated in Figure 3.6b, this attack had an average complexity of $2^{7.402}$ encryptions, indicating a speed up over exhaustive search.

3.6 Conclusion

In this chapter, we present a 2-stage MITM on several instances of `LowMC` using only a single plaintext/ciphertext. The first MITM stage reduces the key candidates corresponding to a fraction of key bits of the master key. The second MITM stage between this reduced candidate

set and the remaining fraction of key bits successfully recovers the master key. We have shown with experimental evidence on smaller versions of LowMC that the combined computational complexity of both these stages is significantly lower than those reported in [BBDV20].

Quantum Cryptography Part II

4 Public-Key Encryption With Quantum Keys

This chapter is dedicated to the study of the notion of Public-Key encryption with quantum keys (qPKE). We show that, unlike the classical counterpart, qPKE can be constructed from MiniCrypt assumptions. We further show although qPKE can not provide unconditional security, it is possible to build this primitive in a world in which one-way functions do not. The personal contribution of this work is taken from joint work with Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter [BGHD+23].

Structure of the Chapter: We begin this chapter by motivating the problem and summarizing the results presented throughout the chapter in section 4.1. We continue by giving a technical overview of the contributions done in this work in section 4.1.2. We summarize the related and concurrent work on this problem in sections 4.1.3 and 4.1.4. The formal definitions and security notions for qPKE are provided in section 4.2. Later, in section 4.3 we present two constructions for qPKE, based on the existence of one-way functions and pseudorandom function-like state generators. We continue by presenting another construction of qPKE based on pseudorandom function-like states with proof of destruction in section 4.4. Section 4.5 contains the impossibility proof for information-theoretically secure qPKE. We ultimately conclude the chapter in section 4.6.

4.1 Introduction

The use of quantum resources to enable cryptographic tasks under weaker assumptions than classically needed (or even *unconditionally*) were actually the first concrete proposals of quantum computing, with the seminal quantum money protocol of Wiesner [Wie83] and the key-exchange protocol of Bennett and Brassard [BB84]. Ever since, the field of quantum cryptography has seen a surge of primitives that leverage quantum information to perform tasks that classically require stronger assumptions, or are downright impossible. Recent works [BCKM21, GLSV21] have shown that there exist quantum protocols for oblivious transfer, and therefore arbitrary multi-party computation (MPC), based solely on the existence of one-way functions (OWF) [BCKM21, GLSV21], or pseudorandom states (PRS) [JLS18], which

potentially entail even weaker computational assumptions [Kre21, KQST22]. It is well-known that, classically, oblivious transfer and MPC are "Cryptomania" objects, i.e., they can only be constructed from more structured assumptions that imply public-key encryption (PKE). Thus, the above results seem to challenge the boundary between Cryptomania and MiniCrypt, in the presence of quantum information. Motivated by this state of affairs, in this work we investigate the notion of *PKE itself*, the heart of Cryptomania, through the lenses of quantum computing. That is, we ask the following question:

Does public-key encryption (PKE) belong to MiniQCrypt?

Known results around this question are mostly negative: It is known that PKE cannot be constructed in a black-box manner from OWFs [IR90], and this result has been recently re-proven in the more challenging setting where the encryption or decryption algorithms are quantum [ACC+22]. However, a tantalizing possibility left open by these works is to realize PKE schemes from OWFs (or weaker assumptions), where public-keys or ciphertexts are quantum states.

4.1.1 Our results

In this work we initiate the systematic study of quantum public-key encryption (qPKE), i.e., public-key encryption where public-keys and ciphertexts are allowed to be quantum states. We break down our contributions as follows.

1. Definitions. We provide a general definitional framework for qPKE, where both the public-key and the ciphertext might be general quantum states. In the classical setting, there is no need to provide oracle access to the encryption, since the public-key can be used to implement that. In contrast, if the public-key is a quantum state, it might be measured during the encryption procedure, and the ciphertexts might depend on the measurement outcome. In fact, this is the approach taken in some of our constructions. This motivates a stronger security definition, similar to the classical counterpart, in which the adversary gets additional access to an encryption oracle that uses the same quantum public-key that is used during the challenge phase. We define IND-CPA-EO (respectively, IND-CCA-EO) security by adding the encryption oracle (EO) to the standard IND-CPA (respectively, IND-CCA) security game.

2. Constructions. With our new security definition at hand, we propose three protocols for implementing qPKE from OWF and potentially weaker assumptions, each with its own different advantages and disadvantages. More concretely, we show the existence of:

1. A qPKE scheme with quantum public-keys and classical ciphertexts that is IND-CCA-EO[1] secure, based on post-quantum OWF, in section 4.3.1.

[1] Throughout this chapter, unless explicitly specified, by IND-CCA we refer to the notion of adaptive IND-CCA2.

2. A qPKE scheme with quantum public-key and quantum ciphertext that is IND-CCA1 secure, based on pseudo-random function-like states (PRFS) with super-logarithmic input-size[2], in section 4.3.2. Since this scheme is not EO secure, each quantum public-key enables the encryption of a single message.

3. A qPKE scheme with quantum public-key and classical ciphertext that is IND-CPA-EO secure based on pseudo-random function-like states with proof of destruction (PRFSPDs), in section 4.4.

We wish to remark that it has been recently shown that OWF imply PRFS with super-logarithmic input-size [AQY22] and PRFSPDs [BBSS23]. Therefore, the security of the second and third protocols is based on a potentially weaker cryptographic assumption than the first one. Furthermore, PRFS with super-logarithmic input-size are *oracle separated* from one-way functions [Kre21]; therefore, our second result shows a black-box separation between a certain form of quantum public-key encryption and one-way functions. On the other hand, for the other two constructions, even if the public-key is a quantum state, the ciphertexts are classical and, furthermore, one quantum public-key can be used to encrypt multiple messages. The first protocol is much simpler to describe and understand since it only uses standard (classical) cryptographic objects. Moreover, we show that this scheme guarantees the notion of adaptive CCA2 security and is the only scheme that achieves perfect correctness.

3. Lower Bounds. To complete the picture, we demonstrate that *information-theoretically secure* qPKE does not exist. Due to the public-keys being quantum states, this implication is not trivial like the classical case. In fact, some of the existing constructions of qPKE [Got05] have been conjectured to be unconditionally secure, a conjecture that we invalidate in this work. While this general statement follows by recent and independent implications in the literature (see section 4.5 for more details), in this work we present a novel proof of this fact, borrowing techniques from shadow tomography, which we consider to be of independent interest.

4.1.2 Technical overview

In this section, we provide a technical overview of our results. In Section 4.1.2, we explain the challenges and choices to define qPKE and its security definition. In Section 4.1.2, we present 3 instantiations of qPKE, each based on a different assumption and with different security guarantees. Ultimately, Section 4.1.2 is dedicated to the impossibility of information-theoretically secure qPKE and a high-level overview of the proof technique.

[2] It is worth mentioning that PRS implies PRFS with logarithmic size inputs, but no such implication is known for super-logarithmic inputs.

Definitions of qPKE

In order to consider public-key encryption schemes with quantum public-keys, we need to revisit the traditional security definitions. In the case of quantum public-keys, there are several immediate issues that require revision.

The first issue is related to the access the adversary is given to the public-key. In the classical-key case (even with quantum ciphertexts), the adversary is given the classical public-key pk. Given a single quantum public-key, one cannot create arbitrary number of copies of the quantum public-key, due to no-cloning. Hence, to naturally extend notions such as IND-CPA security, we provide multiple copies of the quantum public-key to the adversary (via the mean of oracle access to the quantum public-key generation algorithm).

The second issue concerns the quantum public-key's *reusability*. Classically, one can use the public-key to encrypt multiple messages. With quantum public-keys, this might not be the case: the quantum public-key might be consumed during the encryption. In a non-reusable scheme, the user needs a fresh quantum public-key for every plaintext they wish to encrypt. In fact, in the PRFS-based construction (see section 4.3.2), part of the quantum public-key is sent as the (quantum) ciphertext, so clearly, this construction is *not* reusable.

Thirdly, it could be the case that in a reusable scheme, each encryption call changes the public-key state ρ_{qpk} in an irreversible way. Hence, we make a syntactic change: $\mathcal{E}nc(\rho_{qpk}, m)$ outputs (c, ρ'_{qpk}), where c is used as the ciphertext and ρ'_{qpk} is used as the key to encrypt the next message. It is important to notice that in this scenario, the updated public-key is not publicly available anymore and is only held by the party who performed the encryption.

Lastly, the syntactic change mentioned above also has security effects. Recall that classically, there is no need to give the adversary access to an encryption oracle, since the adversary can generate encryption on their own. Alas, with quantum public-keys, the distribution of ciphers might depend on the changes that were made to the quantum public-key by the challenger whenever the key is used to encrypt several messages. Therefore, for reusable schemes, we define two new security notions, denoted CPA-EO and CCA-EO, that are similar to CPA and CCA security but the adversary is given additional access to an encryption oracle (EO). We note there are several works considering the notions of chosen-ciphertext security in the quantum setting, because it is not clear how to prevent the adversary from querying the challenge ciphertext, if it contains a quantum states. However, we only consider CCA-security for schemes with classical ciphertexts, and therefore this issue does not appear in this work.

Pure vs Mixed States. We mention explicitly that we require our public-keys to be *pure states*. This is motivated by the following concern: there is no general method to authenticate/sign quantum states. One proposal to ensure that the certificate authority (CA) is sending the correct state is to distribute various copies of the keys to different CAs and test whether they are all sending the same state [Got05]. This ensures that, as long as at least one CA is

honest, the user will reject a malformed key with a probability correlated to the distance of the tampered key from the real one. However, this argument crucially relies on the public-key being a pure state (in which case comparison can be implemented with a SWAP-test). On the other hand, if the public-key was a mixed state, there would be no way to run the above test without false positives.

It has been shown that authenticating arbitrary quantum states is impossible [BCG+02a]. However, follow-up work has shown how tampering with the public-key can be prevented using the structure of the key [KMNY23, MW23].

We also mention that, if mixed states are allowed, then there is a trivial construction of qPKE from any given symmetric encryption scheme $(\mathsf{SKE.key\text{-}gen}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$, as also observed in [MY22a, Theorem C.6], which we describe in the following. To generate the keys, we use the output of $\mathsf{SKE.key\text{-}gen}$ as the secret-key and use it to create the uniform mixture

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes |\mathsf{Enc}_{\mathsf{sk}}(x)\rangle\langle \mathsf{Enc}_{\mathsf{sk}}(x)| \tag{4.1}$$

as the public-key. The ciphertext corresponding to a message m is given by $(\mathsf{Enc}_x(m), \mathsf{Enc}_{\mathsf{sk}}(x))$. To decrypt, the decryptor would first recover x by decrypting the second element in the ciphertext using sk, and then recover m by decrypting the first item using x as the secret key.

Constructions for qPKE

As previously mentioned, we propose in this work three schemes for qPKE, based on three different assumptions, each providing a different security guarantee.

qPKE from OWF. Our first instantiation of qPKE is based on the existence of post-quantum OWFs. For this construction, we aim for the strong security notion of indistinguishability against adaptive chosen ciphertext attacks with encryption oracle referred to as IND-CCA-EO. We start with a simple bit-encryption construction that provides IND-CCA security and we discuss how one can modify the scheme to encrypt multi-bit messages and also provide EO security.

Our first scheme assumes the existence of a *quantum-secure pseudorandom function (PRF)*, which can be built from quantum-secure one-way functions [Zha12]. Given a PRF ensemble $\{f_k\}_k$, the public key consists of a pair of pure quantum states $\mathsf{qpk} = (|\mathsf{qpk}_0\rangle, |\mathsf{qpk}_1\rangle)$ and the secret key consists of a pair of bit-strings $\mathsf{dk} = (\mathsf{dk}_0, \mathsf{dk}_1)$ such that, for all $b \in \{0,1\}$,

$$|\mathsf{qpk}_b\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, f_{\mathsf{dk}_b}(x)\rangle,$$

where f_k denotes the quantum-secure PRF keyed by k. To encrypt a bit b, one simply measures all qubits of $|\mathsf{qpk}_b\rangle$ in the computational basis. The result takes the form $(x, f_{\mathsf{dk}_b}(x))$ for some

uniformly random $x \in \{0,1\}^n$ and this is returned as the ciphertext, i.e., $(qc_0, qc_1) = (x, f_{\mathsf{dk}_b}(x))$.

To decrypt a ciphertext (qc_0, qc_1), we apply both f_{dk_0} and f_{dk_1} to qc_0 and return the value of $b \in \{0,1\}$ such that $f_{\mathsf{dk}_b}(qc_0) = qc_1$. In case this does happen for neither or both of the keys, the decryption aborts.

The IND-CCA security of the simple bit-encryption scheme can be proven with a hybrid argument (see appendix B.1). However, there are a few caveats to the scheme that can be pointed out. First, the scheme is not reusable. It can be easily noticed that after using a public-key for encryption, public-key state collapses, meaning that all the subsequent encryption queries are derandomized. This would mean if the same public-key is reused, it can not even guarantee IND-CPA security as the encryption is deterministic.

The second issue is lifting this CCA-secure bit-encryption scheme to a many-bit CCA-secure encryption scheme. In fact, although not trivial, as proven by Myers and Shelat [Ms09], classically it is possible to construct CCA-secure many-bit encryption from CCA-secure bit-encryption. However, the argument cannot be extended to qPKE in a generic way. The main issue is that the construction from [Ms09], similar to the Fujisaki-Okamoto transform, derandomizes the encryption procedure for some fixed random coins. Later these fixed random coins are encrypted and attached to the ciphertext, so that the decryptor can re-encrypt the plaintext to make sure they were handed the correct randomness. Looking at our construction, it is quite clear that it is not possible to derandomize the encryption procedure as the randomness is a consequence of the measurement.

Let us show how the same approach can be modified to circumvent the issues mentioned. Our main observation is that we can use public-keys of the form mentioned before for a key agreement stage and then use the agreed key to encrypt many-bit messages with a symmetric-key encryption scheme (SKE). Let us elaborate. Let $\{f_k\}_k$ be a PRF family and $(\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ be a symmetric-key encryption scheme. It has been proven that quantum-secure one-way functions imply a quantum-secure PRF [Zha12], and post-quantum IND-CCA symmetric encryption [BZ13a]. Consider the following scheme: the secret key dk is a uniformly random key for the PRF, and for a fixed dk, the quantum public-key state is

$$\left|qp\mathcal{K}_{\mathsf{dk}}\right\rangle = \frac{1}{\sqrt{2^\lambda}} \sum_{x\in\{0,1\}^\lambda} |x\rangle \left|f_{\mathsf{dk}}(x)\right\rangle. \tag{4.2}$$

The encryption algorithm will then measure $\left|qp\mathcal{K}_{\mathsf{dk}}\right\rangle$ in the computational basis leading to the outcome $(x^*, y^* = f_{\mathsf{dk}}(x^*))$. The ciphertext of a message m is given by $(x^*, \mathsf{SKE.Enc}(y^*, m))$. To decrypt a ciphertext $(\hat{x}, \hat{c})$, we first compute $\hat{y} = f_{\mathsf{dk}}(x)$ and return $\hat{m} = \mathsf{SKE.Dec}(f_{\mathsf{dk}}(\hat{x}), \hat{c})$.

We emphasize that this scheme is reusable since it allows the encryption of many messages using the same measurement outcome $(x^*, f_{\mathsf{dk}}(x^*))$. Using a hybrid argument, it can be shown that if the underlying SKE guarantees IND-CCA security, this construction fulfills our strongest security notion, i.e. IND-CCA-EO security. A formal description of the scheme, along with a

security proof can be found in section 4.3.1.

QPKE from PRFS. The second construction we present in this work is an IND-CCA1 secure public-key scheme based on the existence of pseudorandom function-like state generators. Our approach is based on first showing bit-encryption, and the discussion regarding how to lift that restriction is discussed in section 4.3.2. The ciphertexts generated by our scheme are quantum states, and as the public-keys of this construction are not reusable, we do not consider the notion of EO security. A family of states $\{|\psi_{k,x}\rangle\}_{k,x}$ is pseudo-random function-like [AQY22] if

1. There is a quantum polynomial-time algorithm Gen such that
$$\mathsf{Gen}(k, \sum_x \alpha_x |x\rangle) = \sum_x \alpha_x |x\rangle |\psi_{k,x}\rangle, \text{ and}$$
2. No QPT adversary can distinguish $(|\psi_1\rangle, ..., |\psi_\ell\rangle)$ from $(|\phi_1\rangle, ..., |\phi_\ell\rangle)$, where $|\psi_i\rangle = \sum_x \alpha_x^i |x\rangle |\psi_{k,x}\rangle$, $|\phi_i\rangle = \sum_x \alpha_x^i |x\rangle |\phi_x\rangle$, and $\{|\phi_x\rangle\}_x$ are Haar random states and the states $|\sigma_i\rangle = \sum_x \alpha_x^i |x\rangle$ are chosen by the adversary.

We continue by providing a high-level description of the scheme. The key generation algorithm picks a uniform PRFS key dk and generates the corresponding public-keys as stated below:

$$\frac{1}{\sqrt{2^\lambda}} \sum_{x\in\{0,1\}^\lambda} |x\rangle |\psi_{\mathsf{dk},x}\rangle^{\otimes n}, \tag{4.3}$$

where $\{|\psi_{k,x}\rangle\}_{k,x}$ is a PRFS family, the size of the input x is super-logarithmic in the security parameter and n is a polynomial in the security parameter.

To encrypt a bit m, the encryptor will then measure the first register of $|qpk\rangle$ to obtain x^* and the residual state after this measurement will be of form $|x^*\rangle |\psi_{\mathsf{dk},x^*}\rangle^{\otimes n}$. They also sample a uniform key dk_1 and compute the state $|\psi_{\mathsf{dk}_1,x^*}\rangle$ then compute the ciphertext $c = (x^*, \rho)$ where

$$\rho = \begin{cases} |\psi_{\mathsf{dk},x^*}\rangle^{\otimes n}, & \text{if } m = 0 \\ |\psi_{\mathsf{dk}_1,x^*}\rangle^{\otimes n}, & \text{if } m = 1 \end{cases}. \tag{4.4}$$

To decrypt a ciphertext $(\hat{x}, \hat{\rho})$, we split $\hat{\rho}$ into n subsystems and check whether each subsystem is the PRFS with input $\hat{x}$ and key dk using the PRFS Test procedure from [AQY22, Section 3.3]. If the test succeeds for all subsystems, we output 0 and otherwise, we output 1. For a large enough n, our scheme achieves statistical correctness.

We prove that this construction guarantees IND-CCA1 security by a hybrid argument in section 4.3.2. We emphasize that as the ciphertexts of the scheme are quantum states it is challenging to define adaptive CCA2 security.

QPKE from PRFSPDs. Our third scheme is based on pseudo-random function-like states with proof of destruction (PRFSPDs), which was recently defined in [BBSS23]. The authors extended the notion of PRFS to pseudo-random function-like states with proof of destruction, where we have two algorithms $\mathcal{D}estruct$ and $\mathcal{V}er$, which allows us to verify if a copy of the PRFS was destructed.

We will discuss now how to provide non-reusable CPA security security[3] of the encryption of a one-bit message and we discuss later how to use it to achieve reusable security, i.e., CPA-EO security. The quantum public-key in this simplified case is

$$\frac{1}{\sqrt{2^\lambda}} \sum_{x\in\{0,1\}^\lambda} |x\rangle \left|\psi_{\mathsf{dk},x}\right\rangle. \tag{4.5}$$

The encryptor will then measure the first register of $\left|qp\mathcal{k}\right\rangle$ and the post-measurement state is $|x^*\rangle \left|\psi_{\mathsf{dk},x^*}\right\rangle$. The encryptor will then generate a (classical) proof of destruction for this state $\pi = \mathcal{D}estruct(\left|\psi_{\mathsf{dk},x^*}\right\rangle)$. The encryption procedure also picks dk_1 uniformly at random, generated $\left|\psi_{\mathsf{dk}_1,x^*}\right\rangle$ and generates the proof of destruction $\pi' = \mathcal{D}estruct(\left|\psi_{\mathsf{dk}_1,x^*}\right\rangle)$. The corresponding ciphertext for a bit b is given by $c = (x^*, y)$, where

$$y = \begin{cases} \pi', & \text{if } b = 0 \\ \pi, & \text{if } b = 1 \end{cases}.$$

The decryptor will receive some value $(\hat{x}, \hat{y})$ and decrypt the message $\hat{b} = \mathcal{V}er(\mathsf{dk}, \hat{x}, \hat{y})$. The proof of the security of the aforementioned construction follows from a hybrid argument reminiscent of the security proof of the previous schemes (see section 4.4). Notice that repeating such a process in parallel trivially gives a one-shot security of the encryption of a string m and moreover, such an encryption is classical. Therefore, in order to achieve IND-CPA-EO secure qPKE scheme, we can actually encrypt a secret key sk that is chosen by the encryptor, and send the message encrypted under sk. We leave the details of such a construction and its proof of security to section 4.4.

Impossibility of Information-Theoretically Secure qPKE

So far, we have established that qPKE can be built from assumptions weaker than the ones required for the classical counterpart, and potentially even weaker than those needed to build secret-key encryption classically. This naturally leads to the question of whether it is possible to build an information-theoretically secure qPKE. In the following, we present an impossiblity proof of this fact, using techniques from the literature on shadow tomography. Although proving the impossibility of classical PKE is immediate, there are a few challenges when trying to prove a result of a similar flavour for qPKE. Even when considering security against a computationally unbounded adversary, there is a limitation that such adversary has,

[3] Meaning that one can only encrypt once using a $\left|qp\mathcal{k}\right\rangle$.

namely, they are only provided with polynomially many copies of the public-key.

The first step of the proof is reducing winning the IND-CPA game to finding a secret-key/public-key pair $(\mathsf{dk}, |qpk_{\mathsf{dk}}\rangle)$ such that

$$\langle qpk^* | qpk_{\mathsf{dk}} \rangle \approx 1.$$

In other words, we show that if $|qpk_{\mathsf{dk}}\rangle$ is relatively close to $|qpk^*\rangle$, there is a good chance that dk can decrypt ciphertexts encrypted by $|qpk^*\rangle$ correctly. A formal statement and the proof of this argument can be found in lemma 4.

Given this lemma, the second part of the proof consists in constructing an adversary that takes polynomially many copies of $|qpk^*\rangle$ as input and outputs $(\mathsf{dk}, |qpk_{\mathsf{dk}}\rangle)$ such that $|qpk_{\mathsf{dk}}\rangle$ is relatively close to $|qpk^*\rangle$. The technique to realize this adversary is *shadow tomography*, which shows procedures to estimate the values $\langle qpk_{\mathsf{dk}} | qpk^* \rangle$ for all $(|qpk_{\mathsf{dk}}\rangle, \mathsf{dk})$ pairs. It is apparent that doing this naively, i.e. by SWAP-testing multiple copies of $|qpk^*\rangle$ with each $|qpk_{\mathsf{dk}}\rangle$, would require exponentially many copies of the public-key $|qpk^*\rangle$. The way we circumvent this problem is by using a recent result by Huang, Kueng, and Preskill [HKP20]. Informally, this theorem states that for M rank 1 projective measurements $O_1, \ldots, O_M$ and an unknown n-qubit state ρ, it is possible to estimate $\mathrm{Tr}(O_i \rho)$ for all i, up to precision ϵ, by only performing $T = O(\log(M)/\epsilon^2)$ single-copy random Clifford measurements on ρ.

Employing this theorem, we show that a computationally unbounded adversary can estimate all the values $\langle qpk_{\mathsf{dk}} | qpk^* \rangle$ from random Clifford measurements on polynomially many copies of $|qpk^*\rangle$. Having the estimated values of $\langle qpk_{\mathsf{dk}} | qpk^* \rangle$ the adversary picks a dk such that the estimated value is relatively large and uses this key to decrypt the challenge ciphertext. Now invoking Lemma 4 we conclude that the probability of this adversary winning the IND-CPA game is significantly more than 1/2.

4.1.3 Related works

The notion of qPKE was already considered in the literature, although without introducing formal security definitions. For instance, Gottesman [Got05] proposed a candidate construction in an oral presentation, without a formal security analysis. The scheme has quantum public-keys and quantum ciphers, which consumes the public-key for encryption. Kawachi et al. [KKNY05] proposed a construction of qPKE (with quantum keys and ciphertexts) from a newly introduced hardness assumption, related to the graph automorphism problem. [OTU00] defines and constructs a public-key encryption where the keys, plaintexts and ciphers are classical, but the algorithms are quantum the (key-generation uses Shor's algorithm). One of the contributions of this work, is to provide a unifying framework for these results, as well as improve in terms of computational assumptions and security guarantees.

In [NI09], the authors define and provide impossibility results regarding encryption with quantum public-keys. Classically, it is easy to show that a (public) encryption scheme cannot have deterministic ciphers; in other words, encryption must use randomness. They show

that this is also true for a quantum encryption scheme with quantum public-keys. In [Dol20], a secure encryption scheme with quantum public keys based on the LWE assumption is introduced. That work shows (passive) indistinguishable security, and is not IND-CPA secure.

In [MY22b, MY22a], the authors study digital signatures with quantum signatures, and more importantly in the context of this work, quantum public-keys.

4.1.4 Concurrent and subsequent work

In a concurrent and independent work, Coladangelo [Col23] proposes a qPKE scheme with a construction that is very different from ours, and uses a quantum trapdoor function, which is a new notion first introduced in their work. Their construction is based on the existence of quantum-secure OWF. However, in their construction, each quantum public-key can be used to encrypt a single message (compared to our construction from OWF, where the public-key can be used to encrypt multiple messages), and the ciphertexts are quantum (whereas our construction from OWF has classical ciphertexts). They do not consider the stronger notion of IND-CCA security.

Building on the results presented in this chapter, two follow-up works [KMNY23, MW23] consider a *stronger* notion of qPKE where the public-key consists of a classical and a quantum part, and the the adversary is allowed to tamper arbitrarily with the quantum part (but not with the classical component).[4] The authors provide constructions assuming quantum-secure OWF. While their security definition is stronger, we remark that our approach is more general, as exemplified by the fact that we propose constructions from potentially weaker computational assumptions. In [BS23], the authors give another solution for the quantum public-key distribution problem using time-dependent signatures, which can be constructed from quantum-secure OWF, but the (classical) verification key needs to be continually updated.

4.2 Definitions of qPKE

In this section, we introduce the new notion of encryption with quantum public keys (definition 10). The indistinguishability security notions are defined in section 4.2.1 and section 4.2.2.

Definition 10 (Encryption with quantum public keys)**.** *Encryption with quantum public keys (qPKE) consists of 4 algorithms with the following syntax:*

1. $\mathsf{dk} \leftarrow \mathcal{G}en(1^\lambda)$*: a QPT algorithm, which receives the security parameter and outputs a classical decryption key.*
2. $|qpk\rangle \leftarrow \mathcal{QPKG}en(\mathsf{dk})$*: a QPT algorithm, which receives a classical decryption key* dk*, and outputs a quantum public key* $|qpk\rangle$*. In this work, we require that the output is a pure*

[4] Because of this stronger security definition, here the notion of public-keys with mixed states is meaningful since there is an alternative procedure to ensure that the key is well-formed (e.g., signing the classical component).

state, and that t calls to $\mathcal{QPKGen}(\mathsf{dk})$ should yield the same state, that is, $|qpk\rangle^{\otimes t}$.

3. $(qc, |qpk'\rangle) \leftarrow \mathcal{Enc}(|qpk\rangle, m)$*: a QPT algorithm, which receives a quantum public key* $|qpk\rangle$ *and a plaintext* m*, and outputs a (possibly classical) ciphertext* qc *and a recycled public key* $|qpk'\rangle$*.*

4. $m \leftarrow \mathcal{Dec}(\mathsf{dk}, qc)$*: a QPT algorithm, which uses a decryption key* dk *and a ciphertext* qc*, and outputs a classical plaintext* m*.*

We say that a qPKE scheme is *correct* if for every message $m \in \{0,1\}^*$ and any security parameter $\lambda \in \mathbb{N}$, the following holds:

$$\Pr\left[\mathcal{Dec}(\mathsf{dk}, qc) = m \;\middle|\; \begin{array}{c} \mathsf{dk} \leftarrow \mathcal{Gen}(1^\lambda) \\ |qpk\rangle \leftarrow \mathcal{QPKGen}(\mathsf{dk}) \\ (qc, |qpk'\rangle) \leftarrow \mathcal{Enc}(|qpk\rangle, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda),$$

where the probability is taken over the randomness of $\mathcal{Gen}$, $\mathcal{QPKGen}$, $\mathcal{Enc}$ and $\mathcal{Dec}$. We say that the scheme is reusable if completeness holds to polynomially many messages using a single quantum public key. More precisely, we say that a qPKE scheme is *reusable* if for every security parameter $\lambda \in \mathbb{N}$, polynomial number of messages $m_1, \ldots, m_{n(\lambda)} \in \{0,1\}^*$, the following holds:

$$\Pr\left[\forall i \in [n(\lambda)],\, \mathcal{Dec}(\mathsf{dk}, qc_i) = m_i \;\middle|\; \begin{array}{c} \mathsf{dk} \leftarrow \mathcal{Gen}(1^\lambda) \\ |qpk_1\rangle \leftarrow \mathcal{QPKGen}(\mathsf{dk}) \\ (qc_2, |qpk_2\rangle) \leftarrow \mathcal{Enc}(|qpk_1\rangle, m_1) \\ \vdots \\ (qc_n, |qpk_{n+1}\rangle) \leftarrow \mathcal{Enc}(|qpk_i\rangle, m_{n(\lambda)}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

4.2.1 Security Definitions for qPKE with Classical Ciphertexts

In this section, we present a quantum analogue of classical indistinguishability security for qPKE with classical ciphertexts. There are a few subtleties in defining this notion that require attention. Firstly, since in general the public keys are quantum states and unclonable, in the security games, we allow the adversary to receive polynomially many copies of $|qpk\rangle$, by making several calls to the $\mathcal{QPKGen}(\mathsf{dk})$ oracle. Secondly, in the classical setting, there is no need to provide access to an encryption oracle since the adversary can use the public key to apply the encryption themself. In the quantum setting, this is not the case: as we will see, the quantum public key might be measured, and the ciphertexts might depend on the measurement outcome. Furthermore, the quantum public key can be reused to encrypt multiple different messages. This motivates a stronger definition of indistinguishability with encryption oracle, in which the adversary gets oracle access to the encryption, denoted as IND-ATK-EO security, where ATK can be either chosen-plaintext attacks (CPA), (adaptive or non-adaptive) chosen-ciphertext attacks (CCA1 and CCA2).

We define the oracles $\mathcal{O}_1, \mathcal{O}_2$ depending on the level of security as follows.

Game 3 Indistinguishability security with an encryption oracle (IND-ATK-EO) for encryption with quantum public key and classical ciphertext schemes.

1: The challenger generates $\mathsf{dk} \leftarrow \mathcal{G}en(1^\lambda)$.
2: The adversary gets 1^λ as an input, and oracle access to $\mathcal{QPKG}en(\mathsf{dk})$.
3: The challenger generates $|qpk\rangle \leftarrow \mathcal{QPKG}en(\mathsf{dk})$. Let $|qpk_1\rangle := |qpk\rangle$.
4: For $i = 1,\ldots,\ell$, the adversary creates a classical message m_i and send it to the challenger.
5: The challenger computes $(qc_i, |qpk_{i+1}\rangle) \leftarrow \mathcal{E}nc(|qpk_i\rangle, m_i)$ and send qc_i to the adversary.
6: During step (2) to step (5), the adversary also gets classical oracle access to an oracle $\mathcal{O}_1$.
7: The adversary sends two messages m'_0, m'_1 of the same length to the challenger.
8: The challenger samples $b \in_R \{0,1\}$, computes $(qc^*, |qpk_{l+2}\rangle) \leftarrow \mathcal{E}nc(|qpk_{\ell+1}\rangle, m'_b)$ and sends qc^* to the adversary.
9: For $i = \ell+2,\ldots,\ell'$, the adversary creates a classical message m_i and send it to the challenger.
10: The challenger computes $(qc_i, |qpk_{i+1}\rangle) \leftarrow \mathcal{E}nc(|qpk_i\rangle, m_i)$ and send qc_i to the adversary.
11: During step (9) to step (10), the adversary also gets classical oracle access to an oracle $\mathcal{O}_2$. One can observe that after step (7), the adversary no longer gets access to oracle $\mathcal{O}_1$.
12: The adversary outputs a bit b'.

We say that the adversary wins the game (or alternatively, that the outcome of the game is 1) iff $b = b'$.

ATK	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
CPA	$\varnothing$	$\varnothing$
CCA1	$\mathcal{D}ec(\mathsf{dk},\cdot)$	$\varnothing$
CCA2	$\mathcal{D}ec(\mathsf{dk},\cdot)$	$\mathcal{D}ec^*(\mathsf{dk},\cdot)$

Here $\mathcal{D}ec^*(\mathsf{dk},\cdot)$ is defined as $\mathcal{D}ec(\mathsf{dk},\cdot)$, except that it return $\perp$ on input the challenge ciphertext qc^*.

Definition 11. *A qPKE scheme is IND-ATK-EO secure if for every QPT adversary, there exists a negligible function ϵ such that the probability of winning the IND-ATK-EO security game (Game 3) is at most $\frac{1}{2} + \epsilon(\lambda)$.*

Remark 1. *The definition presented in definition 11 is stated for the single challenge query setting. Using the standard hybrid argument, it is straightforward to show that single-challenge definitions also imply many-challenge definitions where the adversary can make many challenge queries.*

Remark 2. *We emphasize that the IND-CCA2-EO definition is only well-defined for schemes with classical ciphertexts. The other two notions are well-defined even for quantum ciphertexts, though we do not use those.*

4.2.2 Security Definitions for qPKE with Quantum Ciphertexts

We now give a definition for qPKE with quantum ciphertexts. In the case of adaptive chosen ciphertext security, the definition is non-trivial due to the no-cloning and the destructiveness

of quantum measurements. We note there are indeed several works considering the notions of chosen-ciphertext security in the quantum setting: [AGM18] defines chosen-ciphertext security for quantum symmetric-key encryption (when the message is a quantum state), and [BZ13b, CEV22] defines chosen-ciphertext security for classical encryption under superposition attacks. However, extending the technique from [AGM18] to the public-key setting is non-trivial, and we leave this open problem for future work. In this section, we only consider security notions under chosen-plaintext attacks and non-adaptive chosen-ciphertext attacks.

Even though one can similarly define security notions with encryption oracle for schemes with quantum ciphertexts as in section 4.2.1, bear in mind in all constructions of qPKE with quantum ciphertexts present in this work are not reusable, and thus we do not present the definition in which the adversary has oracle access to the encryption oracle for the sake of simplicity. We denote these notions as IND-ATK, where ATK is either chosen-plaintext attacks (CPA) or non-adaptive chosen-ciphertext attacks (CCA1).

Game 4 IND-ATK security game for encryption with quantum public key and quantum ciphertexts schemes.

1: The challenger generates $\mathsf{dk} \leftarrow \mathcal{G}en(1^\lambda)$.
2: The adversary $\mathcal{A}_1$ gets 1^λ as an input, and oracle access to $\mathcal{QPKG}en(\mathsf{dk})$, $\mathcal{E}nc(|qpk\rangle, \cdot)$ and $\mathcal{O}_1$, and sends m_0, m_1 of the same length to the challenger. $\mathcal{A}_1$ also output a state $|\mathsf{st}\rangle$ and sends it to $\mathcal{A}_2$.
3: The challenger samples $b \in_R \{0,1\}$, generates $|qpk\rangle \leftarrow \mathcal{QPKG}en(\mathsf{dk})$ and sends c^* such that $(c^*, |qpk'\rangle) \leftarrow \mathcal{E}nc(|qpk\rangle, m_b)$ to the adversary $\mathcal{A}_2$.
4: $\mathcal{A}_2$ gets oracle access to $\mathcal{QPKG}en(\mathsf{dk})$, $\mathcal{E}nc(|qpk\rangle, \cdot)$.
5: The adversary $\mathcal{A}_2$ outputs a bit b'.

We say that the adversary wins the game (or alternatively, that the outcome of the game is 1) iff $b = b'$.

The oracles $\mathcal{O}_1$ is defined depending on the level of security as follows.

ATK	Oracle $\mathcal{O}_1$
CPA	$\varnothing$
CCA1	$\mathcal{D}ec(\mathsf{dk}, \cdot)$

Definition 12. *A qPKE scheme with quantum ciphertexts is IND-ATK secure if for every QPT adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function ϵ such that the probability of winning the IND-ATK security game (Game 4) is at most $\frac{1}{2} + \epsilon(\lambda)$.*

4.3 Constructions of CCA-Secure qPKE

In this section, we present our qPKE constructions from OWF and PRFS and prove that their CCA security. The former (given in section 4.3.1) has classical ciphertexts, and allows to encrypt arbitrary long messages. The latter (given in section 4.3.2) has quantum ciphertexts, and only

allows to encrypt a single-bit message. However, the latter is based on a weaker assumption than the former. Finally, in section 4.3.3, we give a remark on the black-box construction of non-malleable qPKE from CPA-secure qPKE using the same classical approach.

4.3.1 CCA-Secure Many-Bit Encryption from OWF

We start by presenting a simple qPKE construction from OWF which prove that it provides our strongest notion of security, i.e. IND-CCA-EO security. The scheme is formally presented in construction 1. The ciphertexts produced by the scheme are classical, and the public-keys are reusable. The cryptographic components of our construction are a quantum secure PRF family $\{f_k\}$ and a post-quantum IND-CCA secure symmetric-key encryption scheme $(\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$, defined in section 1.4.2, which can both be built from a quantum-secure OWF [Zha12, BZ13a].

Construction 1 (IND-CCA-EO secure qPKE from OWF)**.**

- ***Assumptions:*** *A family of quantum-secure pseudorandom functions $\{f_k\}_k$, and post-quantum IND-CCA SKE* $(\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$.
- $\underline{\mathcal{G}en(1^\lambda)}$
  1. $\mathsf{dk} \xleftarrow{\$} \{0,1\}^\lambda$
  2. $\left|qpk\right\rangle \leftarrow \sum_{x\in\{0,1\}^\lambda} |x, f_{\mathsf{dk}}(x)\rangle$
- $\underline{\mathcal{E}nc(\left|qpk\right\rangle, m)}$
  1. *Measure* $\left|qpk\right\rangle$ *to obtain classical strings* x, y.
  2. *Let* $c_0 \leftarrow x$ *and* $c_1 \leftarrow \mathsf{SKE.Enc}(y, m)$.
  3. *Output* (c_0, c_1) *as the ciphertext and* $\left|x, y\right\rangle$ *as the residual public key*
- $\underline{\mathcal{D}ec(\mathsf{dk}, (c_0, c_1))}$
  1. *Compute* $y \leftarrow f_{\mathsf{dk}}(c_0)$.
  2. *Compute* $m \leftarrow \mathsf{SKE.Dec}(y, c_1)$ *and return* m.

It can be trivially shown that the scheme achieves perfect correctness if the underlying SKE provides the perfect correctness property.

Theorem 2. *Let $\{f_k\}_k$ be a quantum secure PRF and* $(\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ *be a post-quantum IND-CCA secure SKE. Then, the quantum qPKE given in construction 1 is IND-CCA-EO secure.*

Proof. We proceed with a sequence of hybrid games detailed in

- **Hybrid** H_0**:** This is the IND-CCA game with Π with the challenge ciphertext fixed to (x^*, c^*) to be the ciphertext portion of $\mathcal{E}nc(\left|qpk\right\rangle, m_0')$.

- **Hybrid** H_1**:** This is identical to H_0 except instead of measuring $\left|qp\mathcal{k}\right\rangle$ when the adversary queries the encryption oracle, the challenger measures a copy of $\left|qp\mathcal{k}\right\rangle$ in advance to obtain $(x^*, y^* = f_{\mathsf{dk}}(x^*))$ and answers queries to the encryption oracle using (x^*, y^*) instead. The decryption oracle still returns $\perp$ when queried (x^*, c^*). This change is only syntactical so the two hybrids are the same from the adversary's view.

The hybrids H_2 to H_5 have 2 main goals: (i) to decorrelate the encryption/decryption oracles $\mathcal{Dec}^*, \mathcal{Enc}$ from the public-keys handed to the adversary and (ii) to remove the oracles' dependency on dk.

- **Hybrid** H_2**:** This is identical to H_1, except (x^*, y^*) is removed from the copies of $\left|qp\mathcal{k}\right\rangle$ handed to the adversary. More precisely, the adversary is handed $\left|qp\mathcal{k}'\right\rangle$ of the following form:

$$\left|qp\mathcal{k}'\right\rangle = \frac{1}{\sqrt{2^{|x^*|}-1}} \sum_{x:x\neq x^*} |x\rangle \left|f_{\mathsf{dk}}(x)\right\rangle \tag{4.6}$$

  Given oracle access to f_{dk}, this state can be efficiently generated. For instance a way to do this would be preparing a uniform superposition, and computing the indicator function $I_{x^*}(x) = 1_{x\neq x^*}$ in a separate register and measuring it. The output of this measurement is 1 with $1-\mathsf{negl}(\lambda)$ probability and the residual state is of the desired form.

  The decryption oracle still returns $\perp$ when queried on the challenge ciphertext. One can observe that $\left|qp\mathcal{k}\right\rangle$ and $\left|qp\mathcal{k}'\right\rangle$ have $\mathsf{negl}(\lambda)$ trace distance so the advantage of distinguishing H_1 and H_2 is $\mathsf{negl}(\lambda)$.

- **Hybrid** H_3**:** This (inefficient) hybrid is identical to H_2 other than f_{dk} being replaced with a truly random function f, i.e. the public-keys are changed to:

$$\left|qp\mathcal{k}'\right\rangle = \frac{1}{\sqrt{2^{|x^*|}-1}} \sum_{x:x\neq x^*} |x\rangle \left|f(x)\right\rangle \tag{4.7}$$

  The encryption and decryption oracle can be simulated by oracle access to f. The decryption oracle returns $\perp$ when queried (x^*, c^*). The indistinugishability of H_3 and H_2 follows directly from pseudorandomness property of $\{f_k\}_k$.

- **Hybrid** H_4**:** This hybrid is identical to H_3 other than y^* being sampled uniformly at random. Upon quering (c_0, c_1) to the decryption oracle if $c_0 \neq x^*$, the oracle computes $y = f(c_0)$ and returns $m = \mathsf{SKE.Dec}(y, c_1)$. In case $c_0 = x^*$ and $c_1 \neq c^*$, the decryption oracle returns $m = \mathsf{SKE.Dec}(y^*, c_1)$. On (x^*, c^*) the oracle returns $\perp$. The encryption oracle returns $(x^*, \mathsf{SKE.Enc}(y^*, m))$ when queried on m. As x^* does not appear in any of the public-keys this change is only syntactical.

- **Hybrid** H_5**:** This hybrid reverts the changes of H_3, i.e. dk' is sampled uniformly at

random and the public-keys are changed as follows:

$$\left|qpk'\right\rangle = \frac{1}{\sqrt{2^{|x^*|}-1}} \sum_{x:x\neq x^*} |x\rangle \left|f_{\mathsf{dk}'}(x)\right\rangle \tag{4.8}$$

With this change, on query (c_0, c_1) if $c_0 \neq x^*$, the decryption oracle computes $y = f_{\mathsf{dk}'}(c_0)$ and returns $m = \mathsf{SKE.Dec}(y, c_1)$. In case $c_0 = x^*$, the decryption oracle simply returns $m = \mathsf{SKE.Dec}(y^*, c_1)$ when $c_1 \neq c^*$ and $\perp$ otherwise. The encryption oracle is unchanged from H_4. The indistinguishability of H_4 and H_5 follows from the pseudorandomness of f and the fact that $\left|qpk'\right\rangle$ and (x^*, y^*) are decorrelated. The hybrid is efficient again.

The next step is to remove the dependency of the encryption and decryption oracles on y^*. This is done by querying the encryption and decryption oracles of the SKE.

- **Hybrid H_6:** Let $\mathsf{SKE.OEnc}$ and $\mathsf{SKE.ODec}^*$ be two oracles implementing the encryption and decryption procedures of SKE with the key y^*. $\mathsf{SKE.ODec}^*$ returns $\perp$ when queried y^*. In this hybrid, we syntactically change the encryption and decryption oracle using these two oracles. To implement the encryption oracle, on query m we simply query $\mathsf{SKE.OEnc}$ on message m and return $(x^*, \mathsf{SKE.OEnc}(m))$. To simulate the decryption oracle, on query (c_0, c_1) we act the same as in H_5 when $c_0 \neq x^*$, but on queries of form (x^*, c) we query $\mathsf{SKE.ODec}^*$ on c and return $\mathsf{SKE.ODec}^*(c)$. Due to the definition of OEnc and ODec^* these changes are also just syntactical. Note that although $\mathsf{SKE.ODec}^*$ always returns $\perp$ on y^*, this event only happens when $c_0 = x^*$, i.e. to cause this event the decryption oracle should be queried on the challenge ciphertext (x^*, c^*).

- **Hybrid H_7:** We provide the adversary with $x^*, \mathsf{SKE.OEnc}, \mathsf{SKE.ODec}^*$, instead of access to the encryption and decryption oracle. The important observation is that the adversary can implement the encryption and decryption oracles themselves by having access to $x^*, \mathsf{SKE.OEnc}, \mathsf{SKE.ODec}^*$ and sampling a uniform dk' themselves and vice versa ($\mathsf{SKE.ODec}^*$ can be queried on c by querying the decryption oracle (x^*, c) and $\mathsf{SKE.OEnc}$ can be queried on m by querying the encryption oracle on m). This demonstrates that the hybrids are only syntactically different and hence are indistinguishable.

- **Hybrid H_8:** This hybrid is identical to H_7 with the only difference that the challenge ciphertext is swapped with $(x^*, \mathsf{SKE.OEnc}(0))$. Now notice that any adversary that can distinguish H_8 from H_7 can effectively break the IND-CCA security of SKE. Hence, the indistinguishability of the two hybrids follows directly from the IND-CCA security of SKE.

  Following the same exact hybrids for challenge ciphertext $\mathcal{E}nc(\left|qpk\right\rangle, m_1')$ we can deduce that the scheme is IND-CCA-EO secure.

□

4.3.2 CCA1-Secure Many-Bit Encryption from PRFS

We continue by presenting a CCA1-secure bit-encryption from PRFS, defined in section 1.4.3. Extending this scheme to polynomially many bits is discussed at the end of this section, see remark 3. The description of the scheme is given below in construction 2.

Construction 2 (IND-CCA1 secure qPKE from PRFS)**.**

- ***Assumptions:*** *A PRFS family* $\{|\psi_{\mathsf{dk},x}\rangle\}_{\mathsf{dk},x}$ *with super-logarithmic input-size. Let* $n := n(\lambda)$.
- $\underline{\mathcal{G}en(1^\lambda)}$
  1. *Output* $\mathsf{dk} \leftarrow_R \{0,1\}^\lambda$.
- $\underline{\mathcal{QPKG}en(\mathsf{dk})}$
  1. *Output* $|qpk\rangle \leftarrow \sum_x |x\rangle_R |\psi_{\mathsf{dk},x}\rangle_S^{\otimes n}$, *where* $x \in \{0,1\}^{\omega(\log\lambda)}$.
- $\underline{\mathcal{E}nc(|qpk\rangle, m)}$ *for* $m \in \{0,1\}$
  1. *Measure the R registers of* $|qpk\rangle$ *to obtain a classical string* x. *Let* $|x\rangle|\phi\rangle := |x\rangle|\psi_{\mathsf{dk},x}\rangle^{\otimes n}$ *denote the residual state.*
  2. *If* $m = 0$, *output the ciphertext as* $(x, |\phi\rangle)$.
  3. *Else, sample a uniformly random key* dk_1, *and output the ciphertext as* $(x, |\psi_{\mathsf{dk}_1,x}\rangle^{\otimes n})$.
- $\underline{\mathcal{D}ec(\mathsf{dk},(x,\Psi))}$
  1. *Split* Ψ *into n equally sized registers and call them* ρ_i *for* $i \in [n]$.
  2. *Run* $\mathsf{Test}(\mathsf{dk},x,\cdot)$ *from [AQY22, corollary 3.9] on each* ρ_i.
  3. *If the outcome of all the tests is* 1, *output* 0, *otherwise output* 1.

We start by providing a proof sketch for the correctness of the scheme.

sketch. First, we can write the correctness term as:

$$\begin{aligned}&\Pr[\mathcal{D}ec(\mathsf{dk},(x,\Psi)) = m | (x,\Psi) \leftarrow \mathcal{E}nc(|qpk\rangle, m)]\\ =&\frac{1}{2}[\Pr[\mathcal{D}ec(\mathsf{dk},\mathcal{E}nc(|qpk\rangle,0)) = 0] + \Pr[\mathcal{D}ec(\mathsf{dk},\mathcal{E}nc(|qpk\rangle,1)) = 1]]\end{aligned}$$

Note that the first term is $1-\mathsf{negl}(\lambda)$ following [AQY22, lemma 3.10]. The only case that requires attention is when the plaintext is 1 but it decrypts to 0, i.e. $\Pr[\mathcal{D}ec(\mathsf{dk},\mathcal{E}nc(|qpk\rangle,1)) = 0]$.

To give a bound on the second term, we use the guarantees of [AQY22, corollary 3.9], which guarantees that

$$\Pr_{\mathsf{dk}_1}[\mathsf{Test}(\mathsf{dk}_1,x,|\psi_{\mathsf{dk},x}\rangle) = 1] \leq \mathbb{E}_{\mathsf{dk}_1}[\mathrm{Tr}(|\psi_{\mathsf{dk},x}\rangle\langle\psi_{\mathsf{dk},x}|\,|\psi_{\mathsf{dk}_1,x}\rangle\langle\psi_{\mathsf{dk}_1,x}|)] \tag{4.9}$$

Using the indistinguishability property of the PRFS, we can deduce that the L.H.S of the equation 4.9 is less than $2^{-\#\text{qubits}(\psi_{\mathsf{dk},x})} + \mathsf{negl}(\lambda)$ via lemma 3.6 from [AQY22]. By doing many runs of this test, we can use Chernoff-type concentration arguments to prove the PKE provides at least constant correctness error, which can be boosted to negligible correctness error using i.i.d repetitions of the encryption.

□

Theorem 3. *The construction in construction 2 is IND-CCA1 secure (see definition 12), assuming* $\{|\psi_{\mathsf{dk},x}\rangle\}_{\mathsf{dk},x}$ *is a PRFS with super-logarithmic input-size.*

Proof. We prove the theorem via a series of hybrids.

- **Hybrid** H_0**.** The original security game as defined in definition 12.
- **Hybrid** H_1**.** This is identical to hybrid H_0, except that the challenger, instead of measuring $|qpk\rangle$ when the adversary queries the encryption oracle for the first time, the challenger measures (the R registers of) this state before providing the copies of $|qpk\rangle$ to the adversary. Note that by measuring $|qpk\rangle$ in the computational basis, the challenger would obtain a classical uniformly random string x^*, let the residual state be $|\phi^*\rangle := |\psi_{\mathsf{dk},x^*}\rangle^{\otimes n}$.

  As the two operations corresponding to the challenger's measurement of $|qpk\rangle$ and the creation of the copies of $|qpk\rangle$ given to the adversary commute, the distributions of the two hybrids are identical and no adversary can distinguish H_0 from H_1 with non-zero advantage.
- **Hybrid** H_2**.** This is identical to hybrid H_1, except that the challenger samples x^* as in the previous hybrid, and instead of providing $|qpk\rangle$ to the adversary, it provides

$$|qpk'\rangle := \frac{1}{\sqrt{2^{|x^*|}-1}} \sum_{x:x\neq x^*} |x\rangle |\psi_{\mathsf{dk},x}\rangle^{\otimes n}.$$

  Moreover, in the challenge query, the challenger uses $(x^*, |\phi^*\rangle)$ for the encryption of the chosen message m, without measuring a fresh copy of $|qpk\rangle$ (that is, it skips the first step of the encryption algorithm). As we elaborated before, this state $|qpk'\rangle$ can be efficiently prepared.

  The distinguishing probability of the two hybrids H_1 and H_2 implies that we can distinguish the following quantum states $|qpk\rangle^{\otimes p} \otimes |x^*\rangle$ and $|qpk'\rangle^{\otimes p} \otimes |x^*\rangle$ with the same probability, but these two quantum states have $\mathsf{negl}(\lambda)$ trace-distance for any polynomial p. Therefore, any adversary can only distinguish H_1 and H_2 with success probability at most $\mathsf{negl}(\lambda)$.
- **Hybrid** H_3**.** This (inefficient) hybrid is identical to H_2, except that the challenger uses a Haar oracle $\mathcal{O}_{\mathsf{Haar}}$ to generate $|qpk'\rangle$ in place of $|\psi_{\mathsf{dk},.}\rangle$. In particular, the quantum

public key in the hybrid H_3 is computed as:

$$\left|qp\mathcal{k}'\right\rangle \leftarrow \sum_{x:x\neq x^*} |x\rangle \otimes |\vartheta_x\rangle^{\otimes n},$$

where each $|\vartheta_x\rangle$ is an output of $\mathcal{O}_{\mathsf{Haar}}$ on input x. The decryption oracle is the same as the decryption algorithm with the difference that $\mathcal{O}_{\mathsf{PRFS}}$ (the algorithm generating the PRFS) is swapped with $\mathcal{O}_{\mathsf{Haar}}$. The crucial point here is that the decryption oracle only uses the PRFS in a black-box way (in particular, it only uses $\mathcal{O}_{\mathsf{PRFS}}$ and does not use $\mathcal{O}^{\dagger}_{\mathsf{PRFS}}$).

Although the decryption oracle can return $\perp$ on query $(x^*,\cdot)$, this can not be used to distinguish the two hybrids as the adversary has a negligible chance of querying x^* as x^* is picked uniformly at random. The adversary is only provided with the value of x^* when given the challenge ciphertext, at which point they do not have access to the decryption oracle anymore.

Notice that, the adversary does not have direct access to this $\mathcal{O}_{\mathsf{Haar}}$, but only via the decryption oracle. By pseudorandomness property of $\left|\psi_{\mathsf{dk},\cdot}\right\rangle$, we have that H_2 and H_3 are computationally indistinguishable.

- **Hybrid** H_4**.** In this hybrid, we revert the changes in H_3, except that the challenger samples a uniformly random key dk' to compute all states in $\left|qp\mathcal{k}'\right\rangle$, except for the one used to encrypt the challenge query. In particular, the public key $\left|qp\mathcal{k}'\right\rangle$ is now generated using the PRFS generator with the key dk', and the secret key dk and its public counterpart $(x^*,\left|\psi_{\mathsf{dk},x^*}\right\rangle^{\otimes n})$ are used for the challenge encryption. With the changes, the hybrid is now efficient again. Similar to the previous argument, H_3 and H_4 are also computationally indistinguishable due to pseudorandomness property of $\left|\psi_{\mathsf{dk}',\cdot}\right\rangle$.

- **Hybrid** H_5**.** This hybrid is identical to H_4, except that in the challenge query, instead of encrypting 0 as $(x^*,\left|\psi_{\mathsf{dk},x^*}\right\rangle^{\otimes n})$, the challenger encrypts 0 as $(x^*,|\vartheta_{x^*}\rangle^{\otimes n})$, where each $|\vartheta_x\rangle$ is an output of $\mathcal{O}_{\mathsf{Haar}}$ on input x.

  Notice that in this hybrid, the secret key dk and its public counterpart $(x^*,\left|\psi_{\mathsf{dk},x^*}\right\rangle)^{\otimes n})$ are not correlated with any of other variables in the hybrid. Furthermore, after receiving the challenge ciphertext, the adversary no longer gets access to the decryption oracle. By the pseudorandomness property of $\left|\psi_{\mathsf{dk},x^*}\right\rangle$, we have that H_4 and H_5 are computationally indistinguishable.

  Furthermore, in this final hybrid, the adversary needs to distinguish the output of PRFS with a uniformly random key dk_1 (for encryption of 1) and the output of a Haar random oracle (for encryption of 0). By the same argument as above, the winning advantage of the adversary is also negligible.

Overall, since all hybrids are negligibly close and the winning advantage of the adversary in the last hybrid in negligible, we conclude the proof. □

Remark 3. *We sketch here how to achieve many-bit encryption (i.e., non-restricted length encryption) from our scheme present above. We do this through several steps.*

- *The scheme stated in construction 2 can easily be extended to a length-restricted scheme, by applying bit-by-bit encryption.*
- *Given a qPKE length-restricted CCA1 encryption, and a (non-restricted length) symmetric key encryption, we can define a hybrid encryption scheme, where the qPKE scheme is used first to encrypt a random (fixed length) secret key, which is later used to encrypt an arbitrarily long message. The entire scheme is CPA- (respectively, CCA1-) secure if the symmetric key encryption has CPA- (respectively, CCA1-) security.*
- *Finally, the following many-bit symmetric key encryption scheme can be proven to be CCA1 secure, using the same proof strategy as in theorem 3, based on the existence of PRFS alone. Given a secret key* dk*, to encrypt a message* $m \in \{0,1\}^\ell$*, we sample* ℓ *distinct uniformly random strings* x_i*, and compute* $\left|\psi_{\mathsf{dk},x_i}\right\rangle^{\otimes n}$*. Then each bit* m_i *will be encrypted using as* $(x_i, \left|\psi_{\mathsf{dk},x_i}\right\rangle^{\otimes n})$ *if* $m_i = 0$*, or* $(x_i, \left|\psi_{\mathsf{dk}',x_i}\right\rangle^{\otimes n})$ *if* $m_i = 1$ *for a fresh key* dk'*.*

4.3.3 Generic Construction of Non-Malleable qPKE

We remark that known implications from the literature can be used to show that IND-CPA secure qPKE *with classical ciphertexts* implies non-malleable qPKE: The work of [CDMW18] shows a black-box compiler from IND-CPA encryption to non-malleable encryption, which also applies to the settings of quantum public-keys. The only subtlety is that the compiler assumes the existence of a one-time signature scheme to sign the ciphertext. In [MY22b, MY22a] it is shown that one-time signatures (with quantum verification keys) exist assuming one-way state generators, which in turn are implied by qPKE. Combining the implications of these two works, we obtain a generic construction of non-malleable qPKE from any IND-CPA secure one.

4.4 IND-CPA-EO secure qPKE from PRFSPD

In this section, we propose a construction for qPKE from pseudo-random function-like states with proof of destruction. The construction is reusable, has classical ciphers, and is CPA-EO secure.

We first import the following result that builds *symmetric*-key encryption from PRFSPD, defined in section 1.4.4.

Proposition 1 ([BBSS23])**.** *If quantum-secure PRFSPD exists, then there exists a quantum CPA symmetric-key encryption with classical ciphertexts.*

We give the formal construction for many-bit reusable encryption scheme from PRFSPD in construction 3.

Construction 3 (IND-CPA-EO secure qPKE from PRFSPD)**.**

- ***Assumptions:*** *A PRFSPD family* $\{|\psi_{\mathsf{dk},x}\rangle\}_{\mathsf{dk},x}$ *and a quantum symmetric encryption scheme with classical ciphers* $\{\mathsf{Enc},\mathsf{Dec}\}$.
- $\underline{\mathcal{G}en(1^\lambda)}$
  1. *Let* $\mathsf{dk}_{0,i} \leftarrow_R \{0,1\}^\lambda$ *for all* $i \in [1,\lambda]$.
  2. *Output* $\mathsf{dk} \leftarrow \{\mathsf{dk}_{0,i}\}_{i\in[1,\lambda]}$.
- $\underline{\mathcal{QPKG}en(\mathsf{dk})}$
  1. *Output* $|qpk\rangle = \bigotimes_{i\in[\lambda]} \frac{1}{\sqrt{2^\lambda}} \sum_{x_i\in\{0,1\}^\lambda} |x_i\rangle |\psi_{\mathsf{dk}_{0,i},x_i}\rangle$.
- $\underline{\mathcal{E}nc(|qpk\rangle, m)}$ *for* $m \in \{0,1\}^*$
  1. *Let* $|qpk_i\rangle := \frac{1}{\sqrt{2^\lambda}} \sum_{x_i\in\{0,1\}^\lambda} |x_i\rangle |\psi_{\mathsf{dk}_{0,i},x_i}\rangle$, *and write* $|qpk\rangle$ *as* $|qpk\rangle = \bigotimes_{i\in[\lambda]} |qpk_i\rangle$.
  2. *Measure the left registers of* $|qpk_i\rangle$ *to obtain classical strings* x_i. *Denote the post-measurement states as* $|\psi'_i\rangle$.
  3. *Set* $y_i \leftarrow \mathcal{D}estruct(|\psi'_i\rangle)$.
  4. *For* $i \in [1,\lambda]$, *pick* $\mathsf{dk}_{1,i} \leftarrow \{0,1\}^\lambda$ *and compute* $|\psi_{\mathsf{dk}_{1,i},x_i}\rangle$.
  5. *Set* $y'_i \leftarrow \mathcal{D}estruct(|\psi_{\mathsf{dk}_{1,i},x_i}\rangle)$ *for all* $i \in [\lambda]$.
  6. *Pick a uniformly random key* $k \leftarrow \{0,1\}^\lambda$.
  7. *Set* $\tilde{y}_i = \begin{cases} y'_i & ,\text{if } k_i = 0 \\ y_i & ,\text{if } k_i = 1 \end{cases}$.
  8. *Output* $\big(\mathsf{Enc}(k,m), \big((x_i,\tilde{y}_i)\big)_i\big)$ *as ciphertext and* $\big(k, \big((x_i,\tilde{y}_i)\big)_i\big)$ *as the recycled public-key.*
- $\underline{\mathcal{D}ec(\mathsf{dk},c)}$
  1. *Interpret* c *as* $\big(c', \big((x_i,\tilde{y}_i)\big)_i\big)$.
  2. *Let* $k'_i = \mathcal{V}er(\mathsf{dk}_{0,i}, x_i, \tilde{y}_i)$ *and let* $k' = k'_0 \ldots k'_\lambda$.
  3. *Output* $\mathsf{Dec}(k', c')$.

The correctness of our scheme relies on the existence of PRFSPD with pseudorandomness and unclonability of proof properties. To give a high-level view, the unclonability property ensures that the probability of the symmetric key computed by the parties not being equal is negligible. This allows us to bound the correctness error of the PKE construction the correctness error of the symmetric-key encryption used in it, plus a negligible loss. Next, we show that this construction achieves IND-CPA-EO security in theorem 4.

Theorem 4. *If quantum-secure PRFSPD with super-logarithmic input size exists, then there exists public-key encryption with classical ciphertexts which is IND-CPA-EO secure.*

Proof. Our construction is given in construction 3. It uses a PRFSPD family $\{|\psi_{\mathsf{dk},x}\rangle\}_{\mathsf{dk},x}$ and a quantum symmetric encryption scheme with classical ciphers $\{\mathsf{Enc},\mathsf{Dec}\}$. We prove the security of our scheme through a series of hybrids.

- **Hybrid H_0.** The original security game as defined in definition 11.
- **Hybrid H_1.** This is identical to hybrid H_0, except that the challenger, instead of measuring $|qpk_i\rangle$ (for all $i \in [\lambda]$) when the adversary queries the encryption oracle for the first time, the challenger measures the left register of each $|qpk_i\rangle$ before providing the copies of $|qpk\rangle$ to the adversary. Note that by measuring $|qpk_i\rangle$ in the computational basis, the challenger would obtain a classical uniformly random string x_i^*.

  As the two operations corresponding to the challenger's measurement of $|qpk\rangle$ and the creation of the copies of $|qpk\rangle$ given to the adversary commute, the distributions of the two hybrids are identical and no adversary can distinguish H_0 from H_1 with non-zero advantage.
- **Hybrid H_2.** This is identical to hybrid H_1, except that the challenger samples x_i^* as in the previous hybrid, and instead of providing $|qpk\rangle$ to the adversary, it provides

$$|qpk'\rangle := \bigotimes_{i\in[\lambda]} \frac{1}{\sqrt{2^{|x_i^*|}-1}} \sum_{x_i : x_i \neq x_i^*} |x_i\rangle |\psi_{\mathsf{dk}_{0,i},x_i}\rangle .$$

  Moreover, in the challenge query, the challenger uses $(x_i^*, |\psi_{\mathsf{dk}_{0,i},x_i^*}\rangle)$ for all $i \in [\lambda]$ for the encryption of the chosen message m, without measuring a fresh copy of $|qpk\rangle$ (that is, it skips the first step of the encryption algorithm).

  The distinguishing probability of the two hybrids H_1 and H_2 implies that we can distinguish the following quantum states $|qpk\rangle^{\otimes p} \otimes \bigotimes_{i\in[\lambda]} |x_i^*\rangle$ and $|qpk'\rangle^{\otimes p} \otimes \bigotimes_{i\in[\lambda]} |x_i^*\rangle$ with the same probability, but these two quantum states have $\mathsf{negl}(\lambda)$ trace-distance for any polynomial p. Therefore, any adversary can only distinguish H_1 and H_2 with success probability at most $\mathsf{negl}(\lambda)$.
- **Hybrid $H_{2,i}$ for $i \in [0,\lambda]$.** We define a series of (inefficient) hybrids $H_{2,i}$, in which $H_{2,0} := H_2$, and we denote $H_{2,\lambda} := H_3$. Each $H_{2,i+1}$ is identical as $H_{2,i}$, except that the challenger uses a Haar oracle $\mathcal{O}_{\mathsf{Haar}_i}$ in place of $|\psi_{\mathsf{dk}_{0,i},\cdot}\rangle$. In particular, the quantum public key in the hybrid $H_{2,i}$ is computed as:

$$|qpk'\rangle \leftarrow \bigotimes_{j=1}^{i} \sum_{x_j : x_j \neq x_j^*} |x_j\rangle \otimes |\vartheta_{x_j}\rangle \otimes \bigotimes_{j=i+1}^{\lambda} \sum_{x_j : x_j \neq x_j^*} |x_j\rangle |\psi_{\mathsf{dk}_{0,j},x_j}\rangle ,$$

  where each $|\vartheta_{x_j}\rangle$ is an output of $\mathcal{O}_{\mathsf{Haar}_j}$ on input x_j. For the challenge encryption query, the challenger uses $(x_j^*, |\vartheta_{x_j^*}\rangle)$ for all $j \in [1,i]$, and $(x_j^*, |\psi_{\mathsf{dk}_{0,j},x_j^*}\rangle)$ for all $j \in [i+1,\lambda]$.

  By pseudorandomness property of $|\psi_{\mathsf{dk}_{0,i},\cdot}\rangle$, we have that $H_{2,i}$ and $H_{2,i+1}$ are computationally indistinguishable.

- **Hybrid $H_{3,i}$ for $i \in [0,\lambda]$.** We define a series of (inefficient) hybrids $H_{3,i}$, in which $H_{3,0} := H_3$, and we denote $H_{3,\lambda} := H_4$. In each $H_{3,i+1}$, we revert the changes in $H_{3,i}$, except that the challenger samples uniformly random keys dk'_i to compute the i-the component in $\left|qpk'\right\rangle$, except for the one used to encrypt the challenge query.

  Similar to the previous argument, $H_{3,i+1}$ and $H_{3,i}$ are also computationally indistinguishable due to pseudorandomness property of $\left|\psi_{\mathsf{dk}'_i,\cdot}\right\rangle$.

- **Hybrid $H_{4,i}$ for $i \in [0,\lambda]$.** We define a series of (inefficient) hybrids $H_{4,i}$, in which $H_{4,0} := H_4$, and we denote $H_{4,\lambda} := H_5$.

  Each hybrid $H_{4,i}$ is identical to $H_{4,i+1}$, except that for the challenge encryption, the challenger does not sample $\mathsf{dk}_{1,i}$ and compute $\left|\psi_{\mathsf{dk}_{1,i},x_i^*}\right\rangle$. Instead, the challenger generates $\left|\vartheta_{x_i^*}\right\rangle$ using a Haar random oracle $\mathcal{O}_{\mathsf{Haar}_i}$ and uses this state to compute y'_i (by applying $\mathcal{D}estruct$ to $\left|\vartheta_{x_i^*}\right\rangle$).

  By the pseudorandomness of $\left|\psi_{\mathsf{dk}_{1,i},\cdot}\right\rangle$, $H_{4,i}$ and $H_{4,i+1}$ are computationally indistinguishable.

- **Hybrid H_6.** This hybrid is identical to H_5, except that now the challenger sets $\tilde{y}_i = y_i$ for all i for the challenge encryption query.

  In this hybrid, both y_i and y'_i are computed by applying $\mathcal{D}estruct$ to a Haar random state, thus they are outputs of the same distribution. Therefore, H_5 and H_6 are identical.

- **Hybrid $H_{6,i}$ for $i \in [0,\lambda]$.** We define a series of hybrids $H_{6,i}$, in which $H_{6,0} := H_6$, and we denote $H_{6,\lambda} := H_7$.

  Each hybrid $H_{6,i+1}$ is identical to $H_{6,i}$, except now instead of using a Haar random oracle in encryption of the challenge query, the challenger samples a fresh key dk_i and uses this key to compute $\tilde{y}_i$ which is a proof of destruction of the state $\left|\psi_{\mathsf{dk}_i,x_i^*}\right\rangle$.

  By pseudorandomness of $\left|\psi_{\mathsf{dk}_i,\cdot}\right\rangle$, $H_{6,i+1}$ and $H_{6,i}$ are computationally indistinguishable.

  After the changes the hybrid H_7 is now efficient again. In this final hybrid, the secret key k of the symmetric key encryption scheme is uniformly random and independent from all the other variables in the hybrid. Thus, we can easily reduce the winning probability of the adversary in this hybrid to the security of the symmetric key encryption scheme, which is negligible.

Overall, we obtain the winning probability of the adversary in the first hybrid H_0 is negligible, and conclude the proof.

□

4.5 Impossibility of Unconditionally Secure qPKE

In the following, we investigate the question on whether qPKE is possible to construct with information-theoretic security, and we give definitive proof against this. First, let us mention that an independent work by Morimae et al. [MY22a] shows that an object called quantum pseudo one-time pad (QPOTP) implies the existence of efficiently samplable, statistically far but computationally indistinguishable pairs of (mixed) quantum states (EFI pairs). QPOTP is a one-time symmetric encryption with quantum ciphertexts and classical keys, whose key length is shorter than the message length. qPKE immediately implies the existence of QPOTP, by increasing the message length, using bit-by-bit encryption. Since EFI pairs cannot exist information-theoretically, this chain of implications rules out the existence of unconditionally secure qPKE.[5]

We provide an independent and direct proof of the impossibility statement using a shadow tomography argument.

A Proof from Shadow Tomography. In order to prove our impossibility result, we first show that if two public-keys $|qpk\rangle$ and $|qpk^*\rangle$ are close, if we encrypt a random bit using $|qpk^*\rangle$, the probability of decrypting correctly with dk is high, where dk is the corresponding secret-key of $|qpk\rangle$.

Lemma 4. *Let λ be the security parameter and $\Gamma = (\mathcal{Gen}, \mathcal{QPKGen}, \mathcal{Enc}, \mathcal{Dec})$ be a qPKE. Let $\mathsf{dk}^*, |qpk\rangle^*$ be a fixed pair of honestly generated keys and for all decryption keys dk define p_{dk} to be:*

$$p_{\mathsf{dk}} = \Pr\left[\mathcal{Dec}(\mathsf{dk}, qc) = pt \;\middle|\; \begin{array}{r} pt \xleftarrow{\$} \{0,1\} \\ (qc, \cdot) \leftarrow \mathcal{Enc}(qpk^*, pt) \end{array}\right]$$

and let $|qpk_{\mathsf{dk}}\rangle \leftarrow \mathcal{QPKGen}(\mathsf{dk})$. For all dk, if $|\langle qpk^*|qpk_{\mathsf{dk}}\rangle| \geq 1-\epsilon$, then $p_{\mathsf{dk}} \geq 1-\sqrt{3\epsilon}$.

Proof. Let $U_{\mathcal{Enc}}$ be the purified implementation of the encryption procedures, i.e. given the state $|qpk^*\rangle|b\rangle|0\rangle$, $U_{\mathcal{Enc}}$ computes the state computed by $\mathcal{Enc}$ prior to the measurement. We argue that for any $|qpk_{\mathsf{dk}}\rangle$ which is close to $|qpk^*\rangle$, the purified ciphertexts generated by the two keys are also close. For any bit b, the purified ciphertext are defined as $q\tilde{c}_b = U_{\mathcal{Enc}}|qpk^*\rangle|b\rangle|0\rangle\langle 0|\langle b|\langle qpk^*|U_{\mathcal{Enc}}^\dagger$ and $q\tilde{c}_b{}' = U_{\mathcal{Enc}}|qpk_{\mathsf{dk}}\rangle|b\rangle|0\rangle\langle 0|\langle b|\langle qpk_{\mathsf{dk}}|U_{\mathcal{Enc}}^\dagger$. We refer to these as purified ciphertexts. Now we can show,

$$\begin{aligned} \mathrm{Tr}(q\tilde{c}_b q\tilde{c}_b{}'^\dagger) &= \mathrm{Tr}(U_{\mathcal{Enc}}\langle qpk^*|qpk_{\mathsf{dk}}\rangle|qpk^*\rangle\langle qpk_{\mathsf{dk}}|U_{\mathcal{Enc}}^\dagger) && (4.10)\\ &= |\langle qpk^*|qpk_{\mathsf{dk}}\rangle|^2 \geq (1-\epsilon)^2 && (4.11) \end{aligned}$$

[5] This observation was pointed out to us by Takashi Yamakawa.

The transition from Equation (4.10) to Equation (4.11) follows from the trace-preserving property of unitaries. Let $\{\Pi^b_{\mathsf{dk}}\}_{\mathsf{dk}}$ be the POVM corresponding to decrypting a purified ciphertext with key dk, i.e. the probability of a purified ciphertext qc being decrypted to b by dk is given by $\mathrm{Tr}(\Pi^b_{\mathsf{dk}} qc)$. Now the term p_{dk} can be rewritten as follows:

$$p_{\mathsf{dk}} = \frac{1}{2}[\mathrm{Tr}(\Pi^0_{\mathsf{dk}} \tilde{qc}_0) + \mathrm{Tr}(\Pi^1_{\mathsf{dk}} \tilde{qc}_1)] \tag{4.12}$$

Now observe that $\mathrm{Tr}(\Pi^0_{\mathsf{dk}} qc'_0) = \mathrm{Tr}(\Pi^1_{\mathsf{dk}} qc'_1) = 1 - \mathsf{negl}(\lambda)$ as we assumed Γ has negligible correctness error. Now we can bound p_{dk} as follows,

$$\begin{aligned}
p_{\mathsf{dk}} &= \frac{1}{2}[\mathrm{Tr}(\Pi^0_{\mathsf{dk}} \tilde{qc}_0) + \mathrm{Tr}(\Pi^1_{\mathsf{dk}} \tilde{qc}_1)] && (4.13)\\
&\geq 1 - \mathsf{negl}(\lambda) - \frac{1}{2}[\mathrm{Tr}(|\Pi^0_{\mathsf{dk}}(\tilde{qc}_0 - \tilde{qc}'_0)|) + \mathrm{Tr}(|\Pi^1_{\mathsf{dk}}(\tilde{qc}_1 - \tilde{qc}'_1)|)] && (4.14)\\
&\geq 1 - \mathsf{negl}(\lambda) - \frac{1}{2}[\mathrm{Tr}(|\tilde{qc}_0 - \tilde{qc}'_0|) + \mathrm{Tr}(|\tilde{qc}_1 - \tilde{qc}'_1|)] && (4.15)\\
&= 1 - \mathsf{negl}(\lambda) - \frac{1}{2}[\sqrt{1 - \mathrm{Tr}(\tilde{qc}_0 \tilde{qc_0}'^{\dagger})} + \sqrt{1 - \mathrm{Tr}(\tilde{qc}_1 \tilde{qc_1}'^{\dagger})}] && (4.16)\\
&\geq 1 - \mathsf{negl}(\lambda) - \sqrt{2\epsilon} \geq 1 - \sqrt{3\epsilon} && (4.17)
\end{aligned}$$

The transition from Equation (4.15) to Equation (4.16) is due to $\tilde{qc}_b$ and $\tilde{qc}'_b$ being pure states. This concludes the proof of the lemma.

□

Given lemma 4 one can reduce the adversary's task in the IND-CPA game to finding a decryption key dk such that the state $|\tilde{qpk}_{\mathsf{dk}}\rangle \leftarrow \mathcal{QPKG}en(\mathsf{dk})$ is close to $|\tilde{qpk}^*\rangle$ in inner product distance. The main technique we use to realize this subroutine of the adversary is shadow tomography introduced by Aaronson et al. [Aar18]. At the core of our proof is the following theorem by Huang, Kueng, and Preskill [HKP20].

Theorem 5 ([HKP20], Theorem 1)**.** *Let $O_1, \ldots, O_M$ be M fixed observables and let ρ be an unknown n-qubit state. There exists a quantum algorithm, only performing $T = O(\log(M/\delta)/\epsilon^2 \times \max_i \mathrm{Tr}(O_i^2))$ single-copy measurements in random Clifford basis of ρ, outputs $\tilde{p}_1, \ldots, \tilde{p}_M$ based on the outcome of the random measurements such that, with probability at least $1-\delta$*

$$\forall i, |\tilde{p}_i - \mathrm{Tr}(O_i \rho)| \leq \epsilon$$

At a high level, the theorem states that outcomes of polynomially many random Clifford measurements on a state, i.e. a classical shadow, are enough to reconstruct an estimate of the statistics obtained by measuring an exponential number of observables. Bare in mind that the post-processing required to reconstruct $\tilde{p}_i$ values is often inefficient, however for our purpose, i.e. proving the impossibility of an information-theoretically secure quantum PKE

the efficiency of the procedure is not of concern. Using theorem 5 we are able to prove the impossibility statement.

Theorem 6. *For any security parameter λ and qPKE $\Gamma = (\mathcal{Gen}, \mathcal{QPKGen}, \mathcal{Enc}, \mathcal{Dec})$ there exists a polynomial m and a computationally unbounded adversary $\mathcal{A}$ who can win the IND-CPA game with significant advantage only given $m(\lambda)$ copies of the public-key.*

Remark 4. *Actually our attack allows us to recover the secret key with high probability, and thus the attack also breaks the one-wayness security of qPKE (which is a weaker security notion than IND-CPA). Thus, our theorem indeed shows a generic impossibility of unconditionally secure qPKE.*

Proof. Let us describe the adversary given m copies of the public-key $\left|qpk^*\right\rangle$ alongside a challenge ciphertext qc. We set the value of m later in the proof. For a value N, we define the following rank 1 projection ensemble $\{\Pi^1_{\mathsf{dk}} = \left|qpk_{\mathsf{dk}}\right\rangle\left\langle qpk_{\mathsf{dk}}\right|^{\otimes N}\}_{\mathsf{dk}\leftarrow\mathcal{Gen}(1^\lambda)}$. The adversary tries to find a decryption key dk such that $\mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N})$ is relatively large. In order to do so the adversary computes $\mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N})$ for all decryption keys dk.

By theorem 5, the adversary performs $T = O(\log(\frac{\#\{\mathsf{dk}|\mathsf{dk}\leftarrow\mathcal{Gen}(1^\lambda)\}}{\delta})\frac{1}{\epsilon^2}\mathrm{Tr}(\Pi^1_{\mathsf{dk}}{}^2))$ random Clifford measurements to compute values $\tilde{p}_{\mathsf{dk}}$ such that with probability $1-\delta$, for all dk we have that, $\left|\tilde{p}_{\mathsf{dk}} - \mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N})\right| \leq \epsilon$. Let us set $\epsilon < 1/6$ and δ to be a small constant, e.g. $1/100$. We claim that if the adversary picks any key such that $\tilde{p}_{\mathsf{dk}} > 1/2$, they have found a key that has a high chance of decrypting the challenge ciphertext correctly. Let us elaborate.

First of all, the adversary finds at least one such dk with probability at least $1-\frac{1}{100}$, as for the correct decryption key dk^*, $\mathrm{Tr}(\Pi^1_{\mathsf{dk}^*}|qpk^*\rangle\langle qpk^*|^{\otimes N}) = 1$ hence $\tilde{p}_{\mathsf{dk}^*} > 1-1/6$ with probability at least $1-\frac{1}{100}$.

The next thing to show is that any dk such that $\tilde{p}_{\mathsf{dk}} > 1/2$ is a *good* decryption key. We have,

$$\mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N}) = \left|\langle qpk_{\mathsf{dk}}|qpk^*\rangle\right|^{2N} \tag{4.18}$$

For all dk such that $p_{\mathsf{dk}} \leq 1-\sqrt{\frac{3}{\log(N)}}$ we have:

$$p_{\mathsf{dk}} \leq 1-\sqrt{\frac{3}{\log(N)}} \Rightarrow \langle qpk_{\mathsf{dk}}|qpk^*\rangle \leq 1-\frac{1}{\log(N)} \tag{4.19}$$

$$\Rightarrow \mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N}) \leq (1-\frac{1}{\log(N)})^{2N} \tag{4.20}$$

$$\leq e^{-2N/\log(N)} \ll 1/3, \text{ for a large enough } N \tag{4.21}$$

This ensures that if the adversary picks any dk such that $\tilde{p}_{\mathsf{dk}} > 1/2$, with probability at least $1-\frac{1}{100}$ we have that $\left|\tilde{p}_{\mathsf{dk}} - \mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N})\right| \leq 1/6$, $\mathrm{Tr}(\Pi^1_{\mathsf{dk}}|qpk^*\rangle\langle qpk^*|^{\otimes N}) > 1/3$ hence, $p_{\mathsf{dk}} > 1-\sqrt{\frac{3}{\log(N)}}$.

As the last step, the adversary uses the dk they obtain from the previous procedure to decrypt the challenge ciphertext qc^*. Using the guarantees of the shadow tomography procedure, with probability at most $\frac{1}{100}$, p_{dk} is larger than $1-\sqrt{\frac{3}{\log(N)}}$ which for a large enough N is larger than $1-\frac{1}{10}$ per-se. By union bound on the fail events, the adversary wins the IND-CPA game with a probability larger than $1-\frac{1}{5}$ which is significantly larger than $1/2$.

The last thing to prove is that T is $\mathsf{poly}(\lambda)$. We set both $1/\epsilon$ and $1/\delta$ to be constants. $\mathrm{Tr}(\Pi^{1}_{\mathsf{dk}}{}^{2})$ is 1 as Π^{1}_{dk} is a rank one projector. Lastly $\#\{\mathsf{dk}|\mathsf{dk}\leftarrow \mathcal{G}en(1^\lambda)\}=2^n$ where $n=\mathsf{poly}(\lambda)$, hence T is polynomial in λ. Hence, the adversary only requires $m=TN$ copies of the public-key which is polynomial in λ. □

4.6 Conclusion

In this chapter we studied the notion of public-key encryption (PKE) through the lens of quantum computation. We showed that if the public-keys of the scheme are allowed to be quantum states, public-key encryption can be built from assumptions weaker than ones classically required. We showed the existance of one-way functions is adequate to build an IND-CCA secure public-key encryption scheme with quantum public-keys and classical ciphertext. Moreover, we showed that by allowing the ciphertexts to also be quantum states, this primitive can be built from assumptions potentially weaker than existence one-way functions. On a final note, we showed that computational assumptions are in fact necessary to realize the notion of public-key encryption, even when the keys and the ciphertexts are quantum states, i.e. quantum public-key encryption can not be information-theoretically secure.

5 Cryptographically Robust Classifiers Against Arbitrary Test Examples in a Quantum Learning Model

In this chapter, we delve into the topic of classification tasks when the test-time examples are provided from an arbitrary distribution, e.g. the test-time examples are adversarially chosen. We show that, in some regimes, assuming the quantum hardness of learning with errors problem (LWE) [Reg05], the classical rejection-rate lower-bounds proven by Goldwasser et al. [GKKM20] for this problem can be circumvented in a quantum learning model. The personal contribution of this chapter is mostly taken from a joint work with Grzegorz Głuch, and Rüdiger Urbanke, published in International Conference on Artificial Intelligence and Statistics (AISTATS) 2023 [GBU23].

Structure of the Chapter: We start by introducing the problem in section 5.1. We then describe the classification model and the main result presented in this chapter in section 5.2 and how it compares to the results from [GKKM20]. In Section 5.3, we introduce the main technical component of our result, i.e. an interactive protocol that allows a verifier to assure examples they have received for classification adhere to the intended distribution. We provide 3 variants of this protocol. In sections 5.3.1 we present the simplest protocol, where the verifier is quantum and the prover and verifier communicate over a quantum channel. We later provide the proof for the main theorem of this work, after stating the guarantees of the classical verifier variant of the sampling protocol in section 5.4. The rest of the chapter is dedicated to building and proving the properties of the sampling protocol where the communication and the verifier are classical. We do this in two steps: in 5.6 we provide a protocol with a quantum verifier with a constant qubit quantum memory and finally in section 5.7 we show how one can make the verifier classical, assuming the hardness of LWE. From a technical perspective, this is done by tweaking the delegation protocol from Mahadev [Mah18] for sampling tasks with arbitrary inputs.

5.1 Introduction

We are interested in the task of classifying[1] test examples that are arbitrary, by which we mean any set of examples from the input space. More formally, assume that a classifier $f : \mathscr{X} \to \{-1,1\}$ was trained using iid samples from the training distribution $\mathscr{D}$. Then, at test time, a set of *arbitrary* examples is given to the classifier. In particular, this models the adversarial robustness setup, where the test time examples are provided by an adversary who applies imperceptible (think of perturbations small in ℓ_2 norm) perturbations to iid samples from $\mathscr{D}$ in order to fool f [SZS+14, NYC15]. This setup also covers a situation in which an adversary is *not* limited to small perturbations. For an example of such a situation consider the case of explicit content detection [YTL+19], where an adversary produces endless variations of an image to pass the detection test.

Perhaps unsurprisingly, the task of classifying arbitrary test examples is impossible to solve in the usual settings. If f has accuracy strictly smaller than 100% and if all the test examples are chosen to correspond to inputs where f makes an error then all of them will be misclassified by f. To resolve issues of this nature a new model was recently introduced in [GKKM20]. The authors argue that one should consider *selective classifiers* and *transductive learning*. A selective classifier is allowed to abstain from prediction on certain examples, while transductive learning refers to a situation, where the (unlabeled) test examples are presented together with (labelled) training examples. In [GKKM20] it is argued that selective classifiers are necessary to obtain meaningful guarantees in the arbitrary test examples case.

The guarantees obtained in [GKKM20] give bounds on the interplay of two quantities: the risk on arbitrary test examples and the rejection rate on iid samples from $\mathscr{D}$ (training distribution). It is natural that there is a trade-off, because one could easily maximize both of this metrics separately by either: rejecting almost all inputs or just applying f without rejecting anything. One of the results in [GKKM20] is a lower bound on the possible trade-offs of these two quantities. The lower bound provides a minimum number of training samples and test examples needed for the risk on arbitrary examples + the rejection rate on $\mathscr{D}$ to be smaller than ϵ. The bound is expressed in terms of the VC-dimension and ϵ. We break this lower bound by considering a quantum model. Instead of the standard samples $x \sim \mathscr{D}$ we assume access to the many qubit quantum states $\sum_{x\in\{0,1\}^n} \sqrt{\mathscr{D}(x)}\,|x\rangle$ – similar to the quantum PAC-learning model by [BJ95].

On the technical side, we borrow heavily from a series of results on the delegation of quantum computation [Mah18]. These techniques allow us to "restrict the actions of the adversary." Using ideas from this line of work, we are able to design a key tool for our result. Namely a protocol between a classical verifier and a quantum prover that guarantees that the samples collected by the verifier at the end of interaction come from a distribution close to $\mathscr{D}$ - we call it a certifiable sampling protocol. This is done under the assumption that the prover cannot solve the LWE problem - an assumption also present in previous works. Our protocol builds

[1] we refer the reader to section 1.6 for the definition of the classification tasks and learning phases.

upon ideas from [Mah18] but it is not just plug and play: in our setting, we need to collect samples from some distribution - whereas, the previous results only provide guarantees for delegating decision problems. Due to these differences, a new protocol(s) is required, together with a careful analysis to verify correctness in this extended setting. For readers who are familiar with the proof in [Mah18], you can see how our extra requirement manifests itself by comparing for instance Theorem 8 and Theorem 15.

5.2 Model and Main Result

Our model differentiates from the standard learning model in the test phase, i.e. after a hypothesis function $f : \mathscr{X} \to \{-1,1\}$ is learned from iid samples from $\mathscr{D}$. The most common modeling approach for the classification of arbitrary test time examples is that instead of receiving examples from nature directly, the classifier is queried on examples that are provided by a potentially dishonest party that gets examples from nature as input. One differentiation that we make is that we make this procedure interactive, i.e. we consider a verifier **V** handed the hypothesis function f, and a prover **P** provided with examples from nature which interact over several rounds, and at the end, **V** classifies an example which was deduced from the interaction. The two quantities of interest in this model are the rejection rate, and the risk on arbitrary examples. The rejection rate is the probability of the verifier rejecting the interaction, or not classifying a sample when interacting with an honest **P**. The risk on arbitrary examples is the maximum probability of **V** misclassifying examples collected from an accepting interaction. The quantities can be seen as the completeness and soundness errors of an interactive proof.

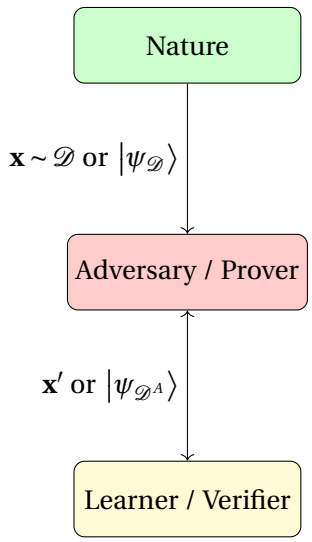


Figure 5.1: Model

The second major difference between our model from the standard one is that the samples from nature are provided as quantum states, similar to the quantum PAC-learning model [BJ95]. Fixing the input space to be $\mathscr{X} = \{0,1\}^n$, instead of being provided samples $\mathbf{x} \sim \mathscr{D}$, the prover receives a quantum state $|\psi_{\mathscr{D}}\rangle = \sum_{x\in\{0,1\}^n} \sqrt{\mathscr{D}(x)}\,|x\rangle$.[2]

5.2.1 Notation and Quantities of Interest

For $\mathscr{D} \in \mathfrak{D}(n)$ we define $\mathsf{O}(\mathscr{D})$ as an oracle giving access to $|\psi_{\mathscr{D}}\rangle := \sum_{x\in\{0,1\}^n} \sqrt{\mathscr{D}(x)}|x\rangle$.In our protocols, we will be interested in an interaction between **V** (Verifier) and **P** (Prover). We will write $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ to denote that **P** has access to $\mathsf{O}(\mathscr{D})$. For a quantum circuit C acting on n-qubits via the unitary transform U_C, we define $\mathscr{D}_C \in \mathfrak{D}(n)$ as the distribution arising from measuring all n qubits of $U_C|0^{\otimes n}\rangle$ in the

[2] In [BJ95] quantum samples are states of the form $\sum_{x\in\{0,1\}^n} \sqrt{\mathscr{D}(x)}\,|x, g(x)\rangle$. Here, g is the ground truth.

computational (which we will also denote as Z) basis. For $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ we say that $\mathscr{D}_\psi \in \mathfrak{D}(n)$ defined as $\mathscr{D}_\psi(x) = |\langle x|\psi\rangle|^2$ for every $x \in \{0,1\}^n$ is the distribution associated with $|\psi\rangle$.

As described before the two main quantities of interest are: the rejection rate when **P** acts honestly and the risk when **P** acts maliciously. We define these quantities more formally now. In our protocols, we perform various consistency checks and collect some statistics. Because of that the protocols need to be repeated some number of times to obtain meaningful guarantees. This is why rejection rates and risk on arbitrary examples are defined as values in expectation.

Definition 13 (**Rejection Rate**)**.** *We define the rejection rate as* 1 *minus the expectation of the ratio of the number of samples obtained by* **V** *in the protocol (when an honest* **P** *interacts with* **V***) to the number of states* $|\psi_{\mathscr{D}}\rangle$ *that* **P** *used in the protocol. We denote it by*

$$\perp_{\mathscr{D}} := 1 - \mathbb{E}\left[\frac{\#\textit{examples obtained by}\,\mathbf{V}}{\#\textit{number of}\,|\psi_{\mathscr{D}}\rangle\ \textit{used by}\,\mathbf{P}}\right],$$

where the expectation is over the randomness of **V** *and* **P** *(that also includes the randomness stemming from quantum mechanics).*

Definition 14 (**Risk on Arbitrary Examples**)**.** *We define the risk on arbitrary examples as the supremum over malicious provers accepted with probability* 1 *of the expected risk of* f *on examples accepted by* **V**. *We denote it by*

$$AR^g(\mathbf{V}^f) := \sup_{\mathbf{P}} \mathbb{P}[f(x) \neq g(x) | Out_{\mathbf{V}}(\mathbf{V} \leftrightarrow \mathbf{P}) = (x, Accept)],$$

where the probability is over the randomness of **V** *and* **P** *conditioned on accepted interaction and* x *is sampled at random from all obtained examples. AR stands for Arbitrary Risk but can be also thought of as Adversarial Risk in a sense that it is a risk in the presence of an adversary.*

5.2.2 Main Result

As discussed above, our result is applicable to the test phase. We assume that the training phase is completed and **V** has access to two objects obtained during the training phase: a classifier f and a description of a generative quantum circuit C with the following properties.

The circuit C captures the true distribution well, i.e. $d_H(\mathscr{D}_C, \mathscr{D}) = \eta \ll 1$, where d_H is the Hellinger distance defined in section 1.6. The classifier f is robust with respect to small changes in the distribution (i.e., it is robust to distributional shifts). This means that for all $\mathscr{D}^A \in \mathfrak{D}(n)$ such that $d_H(\mathscr{D}^A, \mathscr{D}) \leq O(\eta)$ we would have $R_{\mathscr{D}^A}(f) \approx R_{\mathscr{D}}(f)$, where $R_{\mathscr{D}}$ is the standard risk, defined in section 1.6.

We claim that if such a $\mathbf{V}^{C,f}$ (**V** having access to f and the description of C) interacts with **P** using our protocol (defined in Section 5.3) then this will yield a framework robust under all (computationally bounded) adversaries. Indeed there are two scenarios of interest: (1) **P** is

honest, (2) **P** is malicious. In (1) **P** acts *equivalently* to just measuring $|\psi_{\mathscr{D}}\rangle$ and sending the result to **V**. Our protocol guarantees that a big fraction of these samples will be accepted (small rejection rate) as they came from $\mathscr{D}$ itself. Classifier f is robust wrt distributional shifts around $\mathscr{D}$, which in particular implies that it has a low risk on $\mathscr{D}$ itself. In (2) the certifiable sampling protocol (defined in Section 5.3) guarantees that the interaction will only be accepted if the distribution *from which* **P** *samples* is close to $\mathscr{D}$. Then again we know that f has low risk on samples from such a distribution, which guarantees low risk on arbitrary examples. Thus we arrive at the main theorem of this work.

Theorem 7. *There exists a universal constant $K \in \mathbb{N}$ such that for every $n \in \mathbb{N}$, any small enough $\eta \in (0,1)$, for every binary, separable classification task with a distribution $\mathscr{D} \in \mathfrak{D}(n)$ and a ground truth $g : \{0,1\}^n \to \{-1,1\}$, every classifier $f : \{0,1\}^n \to \{-1,1\}$ and every quantum circuit C with T gates the following conditions hold. If*

- *($\mathscr{D}_C$ **is a good approximation of** $\mathscr{D}$) $\|\sqrt{\mathscr{D}_C} - \sqrt{\mathscr{D}}\|_2 \leq \eta$ and*
- *(f **is robust wrt distributional shifts**) for all $\mathscr{D}^A$ such that $\|\sqrt{\mathscr{D}^A} - \sqrt{\mathscr{D}}\|_2 \leq K \cdot \eta^{1/4}$ we have $R^g_{\mathscr{D}^A}(f) \leq O(R^g_{\mathscr{D}}(f))$*

then there exists an efficient interactive protocol with the following properties.

- ***(Completeness / Low Rejection Rate)** There exists an honest quantum prover $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ such that*
$$\perp_{\mathscr{D}} = 1 - \Omega\left(\frac{1}{poly(n, T, 1/\eta)}\right).$$
- ***(Soundness / Low Risk)** For every Quantum Polynomial Time (QPT) prover **P** that is accepted by the interaction with probability 1 we have that with high probability*
$$AR^g(\mathbf{V}) = O(R^g_{\mathscr{D}}(f)).$$

For a proof sketch, we refer the reader to Section 5.4.

5.2.3 Comparison to [GKKM20]

In this section, we compare Theorem 7 to the results from [GKKM20] and in particular to the lower bound presented there.

First, let us discuss the similarities and differences between the model from [GKKM20] and our model. In [GKKM20] learner **V** receives as input two sets of samples: the iid samples from $\mathscr{D}$, $x_1, \ldots, x_N$ and a set of arbitrary test examples $\tilde{x}_1, \ldots, \tilde{x}_M$. Having access to both sets **V** rejects some of $\tilde{x}_i$'s and classifies the rest. In our language we think of $\tilde{x}_1, \ldots, \tilde{x}_M$ as being generated by **P**. In our model, during the training phase, **V** has access to iid samples from $\mathscr{D}$

$(x_1,\dots,x_N)$[3]. During the test phase **V** interacts with **P** over many rounds. An honest **P** in the model from [GKKM20] receives samples $x \sim \mathscr{D}$ and forwards them to **V**. For us an honest **P** receives quantum states $|\psi_{\mathscr{D}}\rangle$ and starts interacting with **V** according to our protocol. For both models, we measure two quantities: the risk on accepted samples and the rejection rate when **P** acts honestly.

The models are obviously different as only ours uses quantum states. We will however proceed with a comparison as if they were the same. That is we will treat iid samples from $\mathscr{D}$ and quantum states $|\psi_{\mathscr{D}}\rangle$ as an equivalent resource and compare the number of samples/states needed for meaningful guarantees. The equivalence is justified because, in an idealized setting, an honest **P** generates one sample from $\mathscr{D}$ from one $|\psi_{\mathscr{D}}\rangle$. Apart from this difference, our protocol requires interaction between **V** and **P** while the on in [GKKM20] does not.

Now we are ready to compare Theorem 7 to the lower bound from [GKKM20, Theorem 5.5]. In [GKKM20, Theorem 5.5], in order to have a non-vacuous bound on the rejection rate plus the risk on *accepted* arbitrary examples, one requires the number of examples to be $M = \Omega(d)$, where d is the VC-dimension of the hypothesis[4]. More concretely, the theorem states that to achieve ϵ risk plus rejection rate, $\Omega(d/\epsilon^2)$ training and test examples are required.

Theorem 7 guarantees a non-vacuous bound on the rejection rate plus the risk on accepted samples when $M = \Omega(\text{poly}(n, T, 1/\eta))$, where we think of M as the number of states $|\psi_{\mathscr{D}}\rangle$ that was used by **P** in the protocol. The two quantities, i.e. d and $\text{poly}(n, T, 1/\eta)$, are not comparable in general but there is a crucial difference. Our bound of $\text{poly}(n, T, 1/\eta)$ depends only on the distribution $\mathscr{D}$, because n is the dimension of the input space and T is the number of gates in C. On the other hand, the lower bound of d depends only on the hypothesis class. Thus there exist tasks for which $d \gg n, T$, for some circuits C with T gates for which $\|\sqrt{\mathscr{D}_C} - \sqrt{\mathscr{D}}\|_2 \ll 1$. This implies that Theorem 7 breaks the lower bound from [GKKM20] in some regimes!

For an example of a task for which a separation holds one can take a distribution and a hypothesis class constructed in [GKKM20] that certifies their lower-bound. For $d \in \mathbb{N}$ the distribution used is the uniform distribution over $\{1,\dots,O(d)\}$ and the hypothesis class are all functions of exactly d 1's. By construction, the VC-dimension is equal to d. Moreover, this distribution can be generated exactly ($\|\sqrt{\mathscr{D}_C} - \sqrt{\mathscr{D}}\|_2 = 0$), using quantum Fourier transform, by quantum circuits acting on $n = O(\log(d))$ qubits with $T = O(\log(d))$ gates. We compare: our guarantee gives a non-vacuous bound for $M = \text{poly}(n, T, 1/\eta) = \text{polylog}(d)$ while the lower-bound requires $M = \Omega(d)$. We see an **exponential separation**. This task has an additional property. The lower-bound holds also when the classical algorithm knows $\mathscr{D}$ exactly. This shows that access to a generator C, which is required by our construction, is not a hidden source of separation.

[3] **V** can also have access to states $|\psi_{\mathscr{D}}\rangle$ during the training phase. Our result is about the test phase and the exact mechanics of the training phase are not important as long as **V** has access to f and C.

[4] For a hypothesis class C on an input space $\mathscr{X}^n$, the VC-dimension is an integer d, such that there exist d points in $\mathscr{X}^n$ that can be shattered by functions in C, but there are no $n+1$ points that can be shattered by functions in C.

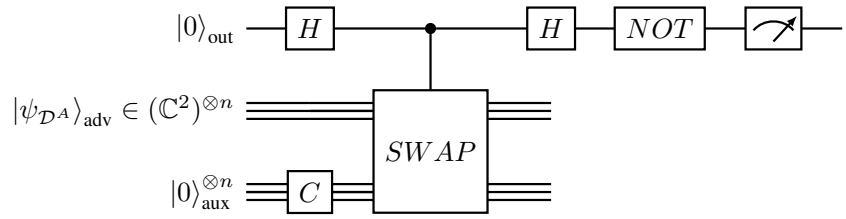


Figure 5.2: Comparison Circuit

5.3 Certifiable Sampling Protocols

Now we move to proving Theorem 7. To do that we show the existence of a protocol, which we name a certifiable sampling protocol. The name comes from the fact that this protocol guarantees that the samples collected by **V** came from a distribution close to the requested one.

We define the protocol in three settings (i) where the **V** has quantum capabilities (ii) where the **V** has access to a constant quantum memory (iii) where the **V** is fully classical. For the sake of readability, we present setting (i) and state the main result from setting (iii) first. We then spend the rest of the chapter proving the guarantees of the other two models (Sections 5.6 and 5.7).

5.3.1 Quantum Verifier

In this section we present a protocol in a setting where **V** has quantum capabilities. We start with an overview and then move to a formal result.

The key component of all our protocols is a quantum circuit G, acting on three registers: out (1 qubit), adv (n qubits) and aux (n qubits), depicted in Figure 5.2. G is parametrized by a quantum circuit C with the associated distribution $\mathscr{D}_C$. Recall that the result of applying U_C to $0^{\otimes n}$ is the state $|\psi_{\mathscr{D}_C}\rangle$. The circuit is designed so that it measures the similarity between $\mathscr{D}^A$ and $\mathscr{D}_C$, where $\mathscr{D}^A$ is the distribution corresponding to the state $|\psi_{\mathscr{D}^A}\rangle_{\text{adv}}$. More precisely, the closer $\mathscr{D}^A$ and $\mathscr{D}_C$ are in terms of the Hellinger distance the higher the probability that G outputs 1 in the out register. Circuits of this form, often referred to as the SWAP test, is a key component of many quantum algorithms [MCEM97].

Equipped with such a comparison circuit we are ready to design a protocol in a model where **V** has quantum capabilities. For now we assume that **P** acts i.i.d. in every round of the protocol (a generalization is discussed in the appendix). In the i-th round of the interaction **P** sends an n-qubit quantum state $|\psi_{\mathscr{D}^A}\rangle$ to **V**, **V** samples a bit $b_i \in \{0,1\}$ uniformly at random. If $b_i = 0$ then **V** inserts $|\psi_{\mathscr{D}^A}\rangle$ as an input to G, computes G, measures the output bit in the Z basis and records the result as γ_i. If $b = 1$ then **V** measures $|\psi_{\mathscr{D}^A}\rangle$ in the Z basis and records the

outcome as $\mathbf{x}_i \in \{0,1\}^n$. After a certain number of rounds (dependent on the desired accuracy and probability of success) **V** computes an average γ_{avg} of the set $\{\gamma_i : b_i = 0\}$. If γ_{avg} is bigger than a certain (to be determined) threshold **V** accepts the interaction and returns the set $\{\mathbf{x}_i : b_i = 1\}$.

Let us now consider the properties of this protocol. Completeness of the protocol is straightforward. An honest **P** can forward the state $|\psi_{\mathscr{D}}\rangle$ he receives to **V**. For soundness of the protocol note the following facts: i) γ_{avg} is a good approximation for the probability that G outputs 1 on $|\psi_{\mathscr{D}^A}\rangle$, ii) this probability is monotonically related to $d_H(\mathscr{D}^A,\mathscr{D}_C)$ by the properties of G, iii) the samples $\{\mathbf{x}_i : b_i = 1\}$ are i.i.d. from $\mathscr{D}^A$, iv) we assumed that $d_H(\mathscr{D},\mathscr{D}_C)$ is small. Moreover we assume that $d_H(\mathscr{D},\mathscr{D}_C) \approx \eta$ and that η is known to **V**. Combining these facts we arrive at the following conclusion. If **V** accepts the interaction then the samples it returns are i.i.d. from a distribution $\mathscr{D}^A$ such that

$$d_H(\mathscr{D}^A,\mathscr{D}) < O(d_H(\mathscr{D}_C,\mathscr{D})). \tag{5.1}$$

The reason the above holds is because we can set the threshold in the protocol over which **V** accepts γ_{avg} to be such that the interaction is accepted when $d_H(\mathscr{D}^A,\mathscr{D}_C) \lesssim \eta$. Then using a triangle-like inequality we arrive at (5.1).

Note 1. *For the most part of this chapter we assume that the* **P** *acts in an i.i.d. fashion. For the fully-quantum verifier, we provide a proof for the general setting where we drop the i.i.d. assumption in Appendix A.3.1.*

Protocol and a Proof

In this section we define the protocol formally and prove its correctness.

The protocol is defined in Figure 5.3. We start by assuming that **P** acts in an i.i.d. fashion and that the states **P** sends are pure. We discuss how to remove these assumptions in Appendix A.3.1 and A.3.2.

Let us prove the correctness of this protocol. The following lemma shows how the distribution of measuring the out register of G relates to the Hellinger distance of $\mathscr{D}_C$ and $\mathscr{D}^A$. The proof is almost identical to the proof of output distribution of a swap-test and is deferred to Appendix A.1.

Lemma 5. *The probability of obtaining outcome* $|1\rangle$ *when measuring the out register of* G *executed on* $|\psi_{\mathscr{D}^A}\rangle$*, i.e.* $\langle 0^{\otimes n}|_{aux}\langle\psi_{\mathscr{D}^A}|_{adv}\langle 0|_{out} G^\dagger \Pi^{(1)}_{out} G|0\rangle_{out}|\psi_{\mathscr{D}^A}\rangle_{adv}|0^{\otimes n}\rangle_{aux}$*, is equal to* $\frac{1}{2}\left(1+(1-d_H^2(\mathscr{D}^A,\mathscr{D}_C))^2\right)$.

Next we show that the number of times each of the types (0 and 1) occurs is at least $N/4$ with high probability. This is a simple application of the Chernoff bound.

Lemma 6. *Let* n_0, n_1 *be the number of times each type occurs in the protocol from Figure 5.3. If* $N = \Omega(\log(1/\delta))$ *then* $\mathbb{P}[n_0, n_1 > \frac{N}{4}] \geq 1-\delta$.

Verifier V **Prover P**

$p := 0, S := \emptyset$

for $i \in \{1, \dots N\}$ where $N = O\left(\frac{K}{\eta^2}\log(1/\delta)\right)$

do :

type $\xleftarrow{\$} \{0,1\}$

$\left|\psi_{\mathscr{D}^A}\right\rangle \in (\mathbb{C}^2)^{\otimes n}$

$\left|\psi_{\mathscr{D}^A}\right\rangle$ ←

if type $= 0$:

Set $b_1, b_2, \dots, b_n$ to be the measurement of

$\left|\psi_{\mathscr{D}^A}\right\rangle$ in the Z basis

$S := S \cup \{(b_j)_{j\in\{1,\dots,n\}}\}$

if type $= 1$:

Set p_i to be the measurement of the out register of

$G|0\rangle_{\text{out}} \left|\psi_{\mathscr{D}^A}\right\rangle_{\text{adv}} \left|0^{\otimes n}\right\rangle_{\text{aux}}$ in the Z basis

$p := p + \frac{p_i}{N}$

done :

if $p < 1 - 2\eta^2$ **abort**

return S

Figure 5.3: The interactive protocol for the model where the verifier has access to a quantum computer.

We are now ready to combine all the pieces and prove that the protocol from Figure 5.3 guarantees that if **V** accepts the interaction then with high probability the samples he collected are i.i.d. from a distribution close to $\mathscr{D}$.

Theorem 8 (Quantum Verifier)**.** *For every circuit C acting on n qubits, for every $\delta \in (0, \frac{1}{3})$, $K \in \mathbb{N}$ and all $\eta > 0$ sufficiently small there exists an interactive protocol between a quantum verifier* **V** *and a quantum prover* **P** *with the following properties. The protocol runs in $N = O\left(\frac{K}{\eta^2}\log(1/\delta)\right)$ rounds and in each round* **P** *sends a pure quantum state on n qubits to* **V**. *At the end of the protocol* **V** *outputs $\perp$ when it rejects the interaction or it outputs $S = \{x_1, \ldots, x_{|S|}\}$, $x_i \in \{0,1\}^n$, when it accepts.*

- ***(Completeness)*** *There exists $\mathbf{P}^{\mathsf{O}(*)}$ such that for every $\mathscr{D} \in \mathfrak{D}(n)$ satisfying $d_H(\mathscr{D}, \mathscr{D}_C) \leq \eta$ the following holds. With probability $1-\delta$ over the randomness in the protocol $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ succeeds, $S \sim_{i.i.d.} \mathscr{D}^{|S|}$, and $|S| \geq \Omega(K)$.*

- ***(Soundness)*** *For every* **P** *that succeeds with probability at least $\frac{2}{3}$ we have $S \sim_{i.i.d.} (\mathscr{D}^A)^{|S|}$ and $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta)$.*[5]

Note 2. *How can we check the success probability of the prover? Assuming that the prover behaves in an i.i.d. fashion, it suffices to run the protocol $(2/\epsilon)\log(1/\delta)$ times. If the fraction of successes is bigger than $1-\epsilon/2$ then we know with confidence $1-\delta$ that the success probability is at least $1-\epsilon$.*

Remark 5. *For certifiable sampling protocols we use the number of repetitions instead of the rejection rates and risk on arbitrary examples. This phrasing is better suited for these protocols. We stated Theorem 7 differently to easily compare it to [GKKM20]. For instance, in Theorem 8 we state that by performing $O(\frac{K}{\eta^2}\log(1/\delta))$ repetitions of the protocol when the prover acts honestly, we collect at least K samples with high probability. However, note that this theorem could also be stated as the probability of collecting a sample at each repetition being η^2, i.e. the rejection rate $\perp_{\mathscr{D}} = 1 - \Omega(\eta^2)$.*

Proof. We start with the completeness property and then move to soundness.

Completness. An honest $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ obtains $|\psi_{\mathscr{D}}\rangle$ from $\mathsf{O}(\mathscr{D})$ and forwards it to **V**. Lemma 6 guarantees that with probability $1-\frac{\delta}{2}$, $n_0, n_1 = \Omega(\frac{K}{\eta^2}\log(1/\delta))$. This automatically guarantees that $|S| \geq \Omega(K)$. Moreover by Fact 2 we have that with probability $1-\frac{\delta}{2}$

$$\left| p - \langle 0^n|_{\text{aux}} \langle \psi_{\mathscr{D}}|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi^{(1)}_{\text{out}} G |0\rangle_{\text{out}} |\psi_{\mathscr{D}}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}} \right| \leq \eta^2. \tag{5.2}$$

[5] $\mathscr{D}^A$ is the implicit distribution from which we collect the samples, which is the distribution corresponding to $|\psi_{\mathscr{D}_A}\rangle$

By Corollary 5 we thus get that $\left|p - \frac{1}{2}\left(1 + (1 - d_H^2(\mathscr{D}, \mathscr{D}_C))^2\right)\right| \leq \eta^2$ holds with probability $1 - \frac{\delta}{2}$. By assumption $d_H(\mathscr{D}, \mathscr{D}_C) \leq \eta$ so we get that $p \geq 1 - 2\eta^2$ (as a function $\frac{1}{2}(1 + (1 - x^2)^2)$ is decreasing). This means that $\mathbf{P}^{0(\mathscr{D})}$ succeeds with probability $1 - \delta/2$.

By the union bound over the error events with probability $1 - \delta/2 - \delta/2 = 1 - \delta$ we have that $|S| \geq \Omega(K)$ and $\mathbf{P}^{0(\mathscr{D})}$ succeeds. The property $S \sim_{\text{i.i.d.}} \mathscr{D}^{|S|}$ holds because the state sent by $\mathbf{P}$ to $\mathbf{V}$ is equal to $|\psi_{\mathscr{D}}\rangle$.

Soundness. By Corollary 5 we get that $\left|p - \frac{1}{2}\left(1 + (1 - d_H^2(\mathscr{D}^A, \mathscr{D}_C))^2\right)\right| \leq \eta^2$ with probability $1 - \frac{\delta}{2}$. $\mathbf{P}$ succeeds with probability $\frac{2}{3}$ so by the union bound and the fact that $\frac{1}{3} + \frac{\delta}{2} < 1$ we get that $h(d_H(\mathscr{D}^A, \mathscr{D}_C)) \geq p - \eta^2 \geq 1 - 3\eta^2$, where we used h to denote the function $\frac{1}{2}(1 + (1 - x^2)^2)$. As h is a decreasing function we get that that $d_H(\mathscr{D}^A, \mathscr{D}_C) \leq \sqrt{1 - \sqrt{2(1 - 3\eta^2) - 1}} \leq 10\eta$, for sufficiently small η. □

5.4 Proof of Theorem 7

The goal of this section is to sketch the proof of theorem 7. The main ingredient of the proof is the classical verifier variant of the protocol described in section 5.3. We provide an informal statement of the theorem regarding the guarantees of such protocol here and move on to the proof of theorem 7. The rest of this chapter is dedicated to proving theorem 9.

Theorem 9 (Classical Verifier)**.** *For a security parameter λ, every generative circuit C acting on n qubits, for every $K \in \mathbb{N}$ and all $\delta, \eta > 0$ sufficiently small there exists an interactive protocol $(\mathbf{V}, *)$ between a classical verifier $\mathbf{V}$ and a quantum prover $\mathbf{P}$ with the following properties. The protocol runs in $N = O\left(\frac{K}{\eta^4} poly(n, T) \log(1/\delta)\right)$ rounds and in each round $\mathbf{P}$ and $\mathbf{V}$ exchange $poly(n, T, \lambda)$ bits. At the end of the protocol $\mathbf{V}$ outputs $\perp$ when it rejects the interaction or it outputs $S = \{x_1, \ldots, x_{|S|}\}$, $x_i \in \{0,1\}^n$, when it accepts.*

- ***(Completeness)*** *There exists a QPT prover $\mathbf{P}^{0(*)}$ such that for every $\mathscr{D} \in \mathfrak{D}(n)$ satisfying $d_H(\mathscr{D}, \mathscr{D}_C) \leq \eta$ the following holds. With probability $1 - \delta$ over the randomness in the protocol $\mathbf{P}^{0(\mathscr{D})}$ succeeds, $S \sim_{i.i.d.} \mathscr{D}^{|S|}$, and $|S| \geq \Omega(K)$.*
- ***(Soundness)*** *For every QPT bounded $\mathbf{P}$ that succeeds with probability 1 we have that with probability $1 - \delta - \mu(\lambda)$ the following conditions hold: $S \sim_{i.i.d.} (\mathscr{D}^A)^{|S|}$ and $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta^{1/4})$, where μ is a negligible function.*

Remark 6. *Just as mentioned in remark 5, the completeness can be stated as $\perp_{\mathscr{D}} = 1 - \Omega(\eta^4 \frac{1}{poly(n,T)})$.*

Having Theorem 9 it is quite straightforward to prove Theorem 7. We provide a short proof sketch.

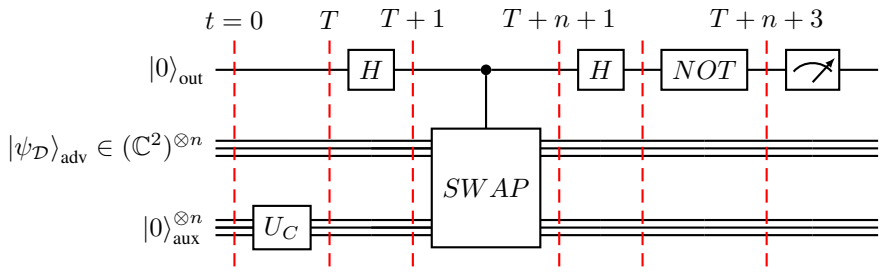


Figure 5.4: omparison Circuit with Time Slices

Theorem 7(sketch). Let us assume that there exists an efficient verifier **V** having access to classifier f and a description of circuit C, satisfying the conditions in the theorem 7, i.e. $\mathscr{D}_C$ is a good approximation of $\mathscr{D}$ and f is robust wrt distributional shifts of maximum distance $M \cdot \eta^{1/4}$ in Hellinger distance. Now according to Theorem 9, for this C, η, and a security parameter λ, there exists an efficient classical verifier **V** interacting with a QPT prover **P** that satisfies the soundness and completeness properties of theorem 9.

Due to the completeness statement of Theorem 9, there exists an honest prover **P** that given access to the distribution $\mathscr{D}$, and partaking in $O(\frac{K}{\eta^4}\text{poly}(n,T)\log(1/\delta))$ repetitions of the protocol, returns samples from the same distribution and is accepted with probability $1-\delta$ for small δ, and collects K examples from $\mathscr{D}$. Now as noted in remark 6 we have,

$$\begin{aligned}\perp_{\mathscr{D}} &= 1 - \mathbb{E}\left[\frac{\#\text{numer of samples collected by } \mathbf{V}}{\#\text{number of states used by } \mathbf{P}}\right] \\ &= 1 - \Omega(\eta^4 \frac{1}{\text{poly}(n,T)})\end{aligned}$$

so the completeness statement of Theorem 7 holds.

For the soundness, due to the soundness statement of Theorem 9, for any QPT bounded (in n and λ) prover **P**, if **P** is accepted with probability 1, with confidence $1-\delta$ we know that if the samples given by the adversary follow a distribution $\mathscr{D}^A$, $d_H(\mathscr{D}^A,\mathscr{D}) \leq O(\eta^{1/4})$. Now using the second assumption in the statement of theorem 7, i.e. (f is robust wrt distributional shifts), as $d_H(\mathscr{D}^A,\mathscr{D}) \leq O(\eta^{1/4})$, we have that $AR^g(\mathbf{V}) = R_{\mathscr{D}^A}(f) \leq O(R_{\mathscr{D}}(f))$, which concludes the soundness proof.

□

5.5 Overview of Making the Verifier Classical

In this section we give an overview of how to generalize the protocol from Figure 5.3 to, first, the setting where **V** has access to a constant memory quantum computer and then to a setting where **V** is fully classical. We present it this way, as the protocols in the consecutive settings

build on top of each other.

Constant Memory Quantum Verifier. In this model the messages in the protocol can still be quantum. (But we will see that in our protocol only **P** will send quantum states and **V** will send only classical messages.) But **V** now only has access to a constant-size quantum computer and can store only a constant number of qubits at each point in time. The only operation that will be required from **V** is measuring the qubits sent by **P** in either the Z or the X basis. Protocols of this form are called receive-and-measure protocols and were already previously considered in the literature, see e.g. [FHcvM18].

Our goal is to emulate the protocol that we designed in the previous step in this more restrictive constant-quantum memory model. The idea is the following. We let **P** choose an n-qubit state $|\psi_{\mathscr{D}^A}\rangle$ and then force them to create a state $|\phi\rangle$ that depends on $|\psi_{\mathscr{D}^A}\rangle$ and to send this state to **V**.[6] The goal is that the constant-sized local subsystems of the state $|\phi\rangle$ contain all the required information that **V** needs to be assured that **P** ran the circuit correctly.

Ideally, the state $|\phi\rangle$ should satisfy the following properties. When **V** measures $|\phi\rangle$ in the Z basis then (i) with probability $\Omega(1/T)$ the distribution of outcomes of measuring one of the qubits is close to the distribution of measuring the output qubit of $U_G|0\rangle|\psi_{\mathscr{D}^A}\rangle|0^{\otimes n}\rangle$ in the Z basis (ii) with probability $\Omega(1/T)$ **V** can obtain $\mathbf{x}' \sim \mathscr{D}^A$.[7] These two operations emulate the steps **V** performed in the previous protocol for $b=0$ and $b=1$, respectively. These operations succeed only with probability $\Omega(1/T)$ but this suffices for our purpose. The main question is how to force **P** to create $|\phi\rangle$ with these properties.

To solve this problem we use the well-known circuit-to-Hamiltonian reduction introduced in [KSV02]. This reduction was originally used to show that a local Hamiltonian problem is QMA-complete. Later on it was a crucial component in the the delegation of quantum computation in the constant quantum memory model [FHcvM18] and in the delegation of quantum computation with a classical verifier in [Mah18]. Unfortunately, we can not use the reduction in a black-box manner. The main issue is, the reduction is designed for decision problems, and our problem of interest is a sampling problem. Hence, in order to use the Hamiltonian model, one would need to modify the reduction to adapt sampling problems.

What is the purpose of this reduction in our context? The circuit-to-Hamiltonian reduction allows to reduce the computation of a quantum circuit G to estimating an energy of a state $|\rho\rangle$ with respect to a local Hamiltonian H_G. In particular, it allows us to build a protocol that forces **P** to prepare a so-called *history state* $|\phi\rangle$ of G. Assume that **P** chooses to evaluate G on a state

[6] By sending the state to **V** we mean sending the state one qubit at a time. Whenever a qubit arrives to **V** they have a choice whether to keep it or discard it. At all times the number of qubits **V** stores cannot exceed the constant predefined number.

[7] Although $\mathbf{x}' \in \{0,1\}^n$ and **V** has only a constant quantum memory it is possible to realize a protocol with these properties. Imagine that while the qubits come to **V** one by one he measures a qubit, records the result, and discards the qubit making room for the next ones. In total, he collects many measurement outcomes out of which he can create $\mathbf{x}'$.

$|\psi_{\mathscr{D}^A}\rangle$. Assume further that the circuit C has T gates and denote $T+n+3$ by T'. Then denote by $|\xi_0\rangle, |\xi_1\rangle, \ldots, |\xi_{T+n+3}\rangle$ the $(2n+1)$-qubit states, where $|\xi_i\rangle$ is the state after the first i gates of G are performed on $|\psi_{\mathscr{D}^A}\rangle$. We refer the reader to Figure 5.4, where the ξ_i's are depicted as time slices in G. With this notation the history state is defined as:

$$|\phi\rangle := \frac{1}{\sqrt{T'+1}}\left(|0\rangle_{\text{clock}}|\xi_0\rangle_{\text{comp}} + |1\rangle_{\text{clock}}|\xi_1\rangle_{\text{comp}} + |2\rangle_{\text{clock}}|\xi_2\rangle_{\text{comp}} + \cdots + |T'\rangle_{\text{clock}}|\xi_{T'}\rangle_{\text{comp}}\right).$$

Hence, $|\phi\rangle$ represents a history of the evaluation of G. It is a superposition of states of the circuit after applying $0,1,2\ldots,T'$ gates of the circuit tensored with a state representing a *clock*. We denoted by comp the concatenation of the three registers out, adv, aux. For instance $|\xi_0\rangle_{\text{comp}} = |0\rangle_{\text{out}}|\psi_{\mathscr{D}^A}\rangle_{\text{adv}}|0^{\otimes n}\rangle_{\text{aux}}$.

Assume for now that **P** sends $|\phi\rangle$ to **V**. We will show that with such a state it is possible to realize the two properties we were hoping for. **V** measures $|\eta\rangle$ in the Z basis and depending on the outcome of measuring the clock register performs further actions.

If the outcome of measuring the clock register is equal to T', which by definition of $|\phi\rangle$ happens with probability $\frac{1}{T'+1}$, then the distribution of measuring the out register is exactly equal to the desired distribution. This is because $|\xi_{T'}\rangle_{\text{comp}}$ represents the last slice of the computation of G (see Figure 5.4).

If the outcome of measuring the clock register is equal to 0 then the distribution of measuring the adv register is exactly equal to $\mathscr{D}^A$. This is because $|\xi_0\rangle_{\text{comp}} = |0\rangle|\psi_{\mathscr{D}^A}\rangle|0^{\otimes n}\rangle$ represents the first slice of the computation of G (see Figure 5.4). One might notice that in the final protocol, we also check if the outcomes of measuring the out and aux registers are all 0. This is done for technical reasons to simplify the proof of soundness.

We realized the two properties we were looking for. Now we can emulate the protocol described in the first step (Quantum Verifier). Thus we will obtain a result similar to Theorem 8 also in this setting. Notice that in each of the cases we were succeeding only with probability $\approx 1/T'$. This will influence the guarantee of Theorem 8 in this model. In particular, this will imply that we will recover 1 sample from $\mathscr{D}$ for every T' states $|\psi_{\mathscr{D}}\rangle$ provided to an honest **P**.

In Section 5.6 we will explain in more detail what it formally means that we can force **P** to produce the history state. In short, the circuit-to-Hamiltonian reduction allows **V** to perform local (which means involving only few qubits) checks on the state obtained from **P** to check that it is in fact a history state. These local checks and the whole reduction have a flavor similar to the famous Cook-Levin proof that shows that 3-SAT is NP-complete.

Classical Verifier. In the last model we consider **V** that is classical and all exchanged messages are also classical. To make our protocol work we need to impose a computational restriction on **P**, namely we assume that **P** is in QPT- Quantum Polynomial Time.

The goal now is to adopt the protocol from the previous step to this model. The protocol can be understood as forcing **P** to construct a history state by performing checks (measurements in the X or the Z basis) that involve only a constant number of qubits. In the model where the communication is only classical, we need to somehow force **P** to perform the measurements chosen by **V** and report the result of these measurements back to **V**.

To achieve this we use an idea that was a crucial component in the delegation of quantum computation with a classical verifier in [Mah18]. A similar idea was used in [BCM+21] to generate certified randomness with a classical verifier. On a high level, we design a protocol that forces **P** to commit to an n-qubit state $|\phi\rangle$, then receive instructions for measurements from **V**, measure $|\phi\rangle$ accordingly and report the results back to **V**. The commitment stage is done using a cryptographic primitive called a claw-free family with adaptive hardcore bit property, defined in section 1.4.5.

5.6 Constant Memory Quantum Verifier

In section 5.3.1 we described a protocol in which a verifier **V** can certify that the distribution of the samples they get from the prover **P** is η-close to the distribution of the samples given by nature. However, this protocol required **V** to perform computation on $2n+1$-qubit states, whereas here we assume quantum memory of **V** is constant.

We proceed by describing a protocol, achieving the same goal, in which **V** can perform operations only on a constant number of qubits.[8] On a high level **V** wants to outsource the execution of the comparison circuit G to **P**. Intuitively we want **P** to send to **V** a state that certifies execution of G. This is possible by modifying a well-known result called circuit-to-Hamiltonian reduction.

Circuit-to-Hamiltonian reduction. This reduction was introduced by Kitaev in the late 1990's, see [KSV02]. This reduction allows one to reduce the computation of a quantum circuit to estimate the ground energy of a local Hamiltonian. With such a tool in hand **V** can first perform the reduction to create H_G, send a classical description of H_G to **P**, then **P** is supposed to send a low energy state $|\psi\rangle$ of H_G back to **V**, and finally **V** estimates the energy of $|\psi\rangle$ with respect to H_G to verify that it is indeed of low energy.

For our purposes, we need a slight modification of the standard reduction. Due to this fact, here we give an overview of this classical result and point to the differences needed for our setup. The main difference is, the output of the circuits we are concerned with are not single bit, and also a portion of the input $\psi_{\mathscr{D}}$, is plugged directly by the prover and **V** does not know what this input is, hence the hamiltonian can not have penalization terms based on a portion of the input and the output of the circuit, otherwise **V** would not be able to compute this

[8] This protocol is based on a circuit-to-Hamiltonian reduction. The size of this constant depends on which reduction we use

Verifier V | **Prover P**

H_G ⟶

$\gamma = 0, p = 0$
$T' = n + T + 3$
$n' = 2n + T' + 1$
$L =$ number of terms of H_G
for $i \in \{1, \dots N\}$ where $N = O\left(\frac{K(n^5 + n^2 T^3 + T^5)}{\eta^4} \log(1/\delta)\right)$
do:
  $t \xleftarrow{\$} \mathsf{Terms}(H_C)$
  $\mathsf{type} \xleftarrow{\$} \{1,2,3\}$
  $n_1, n_2, n_3 = 0$

$|\phi\rangle_{AB} \in (\mathbb{C}^2)_A^{\otimes n'} \otimes \mathscr{H}_B$

⟵ $|\phi\rangle_A$

  if $\mathsf{type} = 0$:
    # measure energy of the state with respect to the Hamiltonian
    $b_i =$ measurement of $|\phi\rangle_i$ in basis σ_i^B, $\quad \forall \sigma_i^B \in t$
    $\gamma = \gamma - J_t(\Pi_{i \in t} b_i)$
    $n_1 = n_1 + 1$
  if $\mathsf{type} = 1$:
    # obtain a sample
    $b =$ measurement of the second qubit of clock in the Z basis
    $a_1, \dots, a_n =$ measurement of aux in the Z basis
    $b' =$ measurement of out in the Z basis
    if $b = 0, b' = 0, a_1, \dots, a_n = 0$:
      $b_1, \dots, b_n =$ measurement of adv in the Z basis
      $S = S \cup \{(b_i)_{i \in [n]}\}$
    $n_2 = n_2 + 1$
  if $\mathsf{type} = 2$:
    # estimate the output probability
    $b_1 =$ measurement of the T'-th qubit of clock in the Z basis
    if $b_1 = 1$:
      $r =$ measurement of out in the Z basis
      $p = p + r$
      $n_3 = n_3 + 1$
Done:
if $\frac{\gamma \cdot L}{n_1} > \frac{\eta^2}{2T'^2} \vee \frac{p}{n_3} < 1 - 2\eta^2$:
  Abort

Figure 5.5: The interactive protocol, in which the verifier collects samples from a distribution close to the desired one. The verifier only requires a single qubit, as they measure one qubit at a time. H_G is the Hamiltonian corresponding to the comparison circuit, described in Figure 5.2. We emphasize that we send $|\phi\rangle$ one qubit a time. When we measure the clock register we use the unary representation of the clock. By writing $|\phi\rangle_{AB} \in (\mathbb{C}^2)_A^{\otimes n'} \otimes \mathscr{H}_B$ and then sending $|\phi\rangle_A$ to **V** we mean that **P** might be sending a mixed state.

hamiltonian. We follow the approach from [KSV02] and we refer the reader to this book for more details.

The starting point of the reduction is the comparison circuit G[9]. Recall that G acts on three registers: out (1 qubit), adv (n qubits), aux (n qubits) and the output of the circuit is obtained by measuring the out register in the Z basis. We want to find an object called a local Hamiltonian H_G.

Definition 15. *We say that an operator $H : (\mathbb{C}^2)^{\otimes N} \to (\mathbb{C}^2)^{\otimes N}$ on N qubits is a k-local Hamiltonian if H is expressible as $H = \sum_{r=1}^{j} H_j$, where each H_j is a Hermitian operator acting on k qubits.*

Our goal will be to define a Hamiltonian that is 5-local. As mentioned before H_G acts on a bigger number of qubits than G does. More precisely it acts on four registers *clock*, comp = (out, adv, aux) - that is there is an additional register called clock in comparison to registers of G. The standard reduction defines

$$H_G = H_{\text{in}} + H_{\text{out}} + H_{\text{prop}} + H_{\text{clock}}.$$

The high level idea is to define the terms $H_{\text{in}}, H_{\text{out}}, H_{\text{prop}}, H_{\text{clock}}$ such that G outputs 1 with high probability if and only if H_G has a small eigenvalue. In this case the minimizing vector $|\phi_{\text{hist}}\rangle$ is the so-called *history state*

$$\frac{1}{\sqrt{T'+1}} \sum_{j=0}^{T'} |j\rangle_{\text{clock}} \otimes G_j \dots G_1 |0\rangle_{\text{out}} |\psi\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}, \tag{5.3}$$

where, for every j, G_j is the unitary transformation corresponding to the j-th gate in G and $|j\rangle_{\text{clock}}$ is a state in the clock state space that we will define in detail later. The terms are defined so that they impose penalties to $\langle\phi| H_G |\phi\rangle$ whenever $|\phi\rangle$ is far from the history state.

For our purposes we change the reduction by removing the H_{out} term. By doing that we will be able to say that for every $|\phi\rangle$ such that $\langle\phi| H_G |\phi\rangle$ is small there exists $|\psi_{\mathscr{D}^A}\rangle_{\text{adv}}$ such that $|\phi\rangle$ is close to the history state for $|\psi_{\mathscr{D}^A}\rangle_{\text{adv}}$. With that property in hand we can then say that if we measure $|\phi\rangle$ in the Z basis then (i) with probability $\Omega(1/T')$ the clock register is equal to $|0\rangle_{\text{clock}}$, the out is equal to $|0\rangle_{\text{out}}$, the aux register is equal to $|0^n\rangle_{\text{aux}}$ and the adv register contains a sample from $\mathscr{D}^A$ (ii) with probability $\Omega(1/T')$ the clock register is equal to $|T'\rangle_{\text{clock}}$ and the out register contains a sample from a Bernoulli variable with parameter p such that p is close to the probability of G outputting 1 on $|0\rangle_{\text{out}} |\psi_{\mathscr{D}^A}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}$. In fact, we can also write this probability as $\langle 0^n| \langle\psi_{\mathscr{D}^A}|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi_1^{(1)} G |0\rangle_{\text{out}} |\psi_{\mathscr{D}^A}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}$, where $\Pi_s^{(\alpha)}$ is the projection onto the subspace of vectors for which the s-th qubit equals α. This notation will be useful later.

[9] The reduction can be applied to any circuit but we focus only on the comparison circuit for simplicity.

Overview of the Protocol. Assuming that the above properties hold we give a high level idea of the protocol defined in Figure 5.5. In each round of the protocol we perform one of the three types of operations, where the type is chosen uniformly at random (i) we estimate the energy $\langle\phi| H_G |\phi\rangle$ (ii) we measure $|\phi\rangle$ in the Z basis and if the clock register is equal to $|0\rangle_{\text{clock}}$, the out register is equal to $|0\rangle_{\text{out}}$ and the aux register is equal to $|0^n\rangle_{\text{aux}}$ then we collect a sample (iii) we measure $|\phi\rangle$ in the Z basis and if the clock register is equal to $|T'\rangle_{\text{clock}}$ we update the estimate for p. We run the protocol $\Theta(T)$ rounds thus each of the types will occur $\Omega(T)$ times with high probability and our reduction guarantees that for (ii) we successfully $\Omega(1)$ samples and for (iii) we update the estimate $\Omega(1)$ times. Overall this guarantees that the estimate for $\langle\phi| H_G |\phi\rangle$ and p will be accurate and the number of samples collected will be in $\Omega(1)$. Our reduction guarantees moreover that if $|\phi\rangle$ is in fact a low energy state of H_G then p is close to the probability of G outputting 1 on $|0\rangle_{\text{out}} |\psi_{\mathscr{D}^A}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}$ and the samples we collect come i.i.d. from distribution $\mathscr{D}^A$ that corresponds to $|\psi_{\mathscr{D}^A}\rangle$. Moreover using Lemma 11 from p we can estimate $|\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle|$, recall that $\mathscr{D}_C$ is the distribution generated by C on $|0^n\rangle$ of which we think as being close to $\mathscr{D}$. As explained in Section 5.3.1 estimating $|\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle|$ is enough to guarantee that the distribution from which we collected the samples is close to $\mathscr{D}$.

For the remainder of this section we first explain the details of the circuit-to-Hamiltonian reduction and then formalize the correctness and soundness requirements and prove the desired properties.

Building the Hamiltonian

The next step is to adapt Kitaev's reduction for our scenario, i.e. for circuits sampling from a distribution rather than solving a decision problem. As most of the steps in the reduction have a significant amount of overlap with Kitaev's original reduction, we only state the final result here and defer the proof to Appendix A.2.

Note 3. *In [CLLW22], Chung et al. proved how the delegation protocol from Mahadev [Mah18] can be extended to sampling problems in BQP. However, the results presented in this chapter were independently found although they share a fair amount of similarities in the techniques used. Another difference between the results is that in our setting, the verifier does not have control over a portion of the input, i.e. the prover plugs in a portion of the input.*

Lemma 7 (Circuit-to-Hamiltonian Reduction)**.** *For every comparison circuit G, for all, sufficiently small, $\epsilon > 0$ there exists an efficiently computable description of a 5-local Hamiltonian H_G with $L = O(n + T')$ many terms such that the following conditions hold. Let $\mathscr{D}^A$ be the distribution of the content of the adv register when measuring $|\phi\rangle$ in the Z basis conditioned on the clock, out and aux registers being all 0 after measurement. For every $|\phi\rangle$ such that $\langle\phi| H_G |\phi\rangle \leq \frac{\epsilon}{T'}$ if we measure $|\phi\rangle$ in the Z basis then*

- *with probability $\in \left[\frac{1-5\epsilon}{T'+1}, \frac{1+5\epsilon}{T'+1}\right]$ the clock register is equal to $|0\rangle_{clock}$, the out register is equal to $|0\rangle_{out}$, the aux register is equal to $|0^n\rangle_{aux}$,*

- *with probability* $\in \left[\frac{1-5\epsilon}{T'+1}, \frac{1+5\epsilon}{T'+1}\right]$ *the clock register is equal to* $\left|T'\right\rangle_{clock}$ *and conditioned on this event the distribution of the out register is a Bernoulli variable with parameter* p *such that* $|p - \langle 0^n|_{aux}\left\langle\psi_{\mathscr{D}^A}\right|_{adv}\langle 0|_{out} G^\dagger \Pi^{(1)}_{out} G|0\rangle_{out}\left|\psi_{\mathscr{D}^A}\right\rangle_{adv}|0^n\rangle_{aux}| \leq 5\epsilon T'$.

Correctness of the Protocol

Recall that protocol from Figure 5.5 builds upon the protocol from Figure 5.3. Now **V**, instead of running G itself, outsources its execution to **P**. On a high level correctness of this new protocol is a consequence of correctness of the quantum verifier protocol (Theorem 8) and circuit-to-Hamiltonian reduction (Lemma 7). One, however, needs to be careful as the guarantees about the protocol will change slightly and some details in the proof need to be verified.

Lemma 8. *Let* n_1, n_2, n_3 *be the number of times each type occurs in protocol from Figure 5.5. If* $N = \Omega(\log(1/\delta))$ *then* $\mathbb{P}[n_1, n_2, n_3 > \frac{N}{6}] \geq 1-\delta$.

Proof. For $b \in \{1,2,3\}$, n_b can be seen as sum of random Bernoulli variables $\{x_i\}_{i\in[N]}$ with parameter $1/3$. Then by Fact 2 we get that $\mathbb{P}[|\frac{n_b}{N} - \frac{1}{3}| > \frac{1}{6}] \leq 2e^{-\frac{N}{72}} \leq \frac{\delta}{3}$. We finish by applying the union bound to the error events.

□

Lemma 9. *Let* ρ_A *be the reduced density of the first* n' *qubits of* $\left|\phi\right\rangle_{AB}$, γ, p, n_1, n_2, n_3, S *be as in the protocol defined in Figure 5.5. Let* p^*, q^* *and* λ *be defined as,*

$$\begin{aligned}
\lambda &= Tr(H_G \rho_A),\\
q^* &= Tr(|0\rangle\langle 0|_{clock} \otimes |0\rangle\langle 0|_{out} \otimes \left|0^n\right\rangle\left\langle 0^n\right|_{aux} \rho_A),\\
p^* &= \frac{Tr(\left|T'\right\rangle\left\langle T'\right|_{clock} \otimes |1\rangle\langle 1|_{out} \rho_A)}{Tr(|1\rangle\langle 1|_{out}\rho_A)}.
\end{aligned}$$

We define the event $\mathscr{F}$ *to be* $\left|\frac{\gamma\cdot L}{n_1} - \lambda\right| \leq \epsilon, \left|\frac{|S|}{n_2} - q^*\right| \leq \epsilon \left|\frac{p}{n_3} - p^*\right| \leq \epsilon$. *If* $N = \Omega(\frac{n^2+T'^2}{\epsilon^2}\log(1/\delta))$ *then* $\mathbb{P}[\mathscr{F}] \geq 1-\delta$.

Note 4. *For the sake of convenience, we often write* $|0\rangle\langle 0|_{clock} \otimes |0\rangle\langle 0|_{out} \otimes |0^n\rangle\langle 0^n|_{aux}$, *when we actually mean* $|0\rangle\langle 0|_{clock} \otimes |0\rangle\langle 0|_{out} \otimes I_{adv} \otimes |0^n\rangle\langle 0^n|_{aux}$.

Proof. Notice that for every term $t \in H_G$ we have $|J_t| \leq 1$. Then if $n_1 = \Omega(\frac{L^2}{\epsilon^2}\log\frac{1}{\delta})$ then Fact 2 guarantees that $\mathbb{P}[|\frac{\gamma\cdot L}{n_1} - \lambda| > \epsilon] \leq \delta$.

Next we define Bernoulli variables $\{s_i\}_{i\in[n_2]}$ to indicate whether $|S|$ increases in a given round, i.e. $|S| = \sum_{i=1}^{n_2} s_i$. By definition $\mu = \mathbb{E}[s_i] = \mathrm{Tr}(|0\rangle\langle 0|_{\mathrm{clock}} \otimes |0\rangle\langle 0|_{\mathrm{out}} \otimes |0^n\rangle\langle 0^n|_{\mathrm{aux}} \rho_A)$. Using Fact 2 we get that if $n_2 = \Omega(\frac{1}{\epsilon^2}\log(1/\delta))$ then $\mathbb{P}\left[\left|\frac{|S|}{n_2} - q^*\right| > \epsilon\right] \leq \delta$. The exact same argument can be used for $\frac{p}{n_3}$.

To conclude, by the union bound, if $n_1 = \Omega(\frac{L^2}{\epsilon^2}\log(1/\delta))$ and $n_2, n_3 = \Omega(\frac{1}{\epsilon^2}\log(1/\delta))$ then $\mathbb{P}[\mathscr{F}] \geq 1-\delta$. By Lemma 8 and the union bound we get that if $N = \Omega(\frac{L^2}{\epsilon^2}\log(1/\delta))$ then $\mathbb{P}[\mathscr{F}] \geq 1-\delta$. As Lemma 7 guarantees that $L = O(n+T')$ we can also set $N = \Omega(\frac{n^2+T'^2}{\epsilon^2}\log(1/\delta))$.

□

Intuitively Lemma 9 guaranties that with a high probability, the estimates $\frac{\gamma \cdot L}{n_1}, \frac{|S|}{n_2}, \frac{p}{n_3}$ are accurate enough. With that fact in hand we proceed by stating the main theorem of this section.

Theorem 10 (Constant Memory Quantum Verifier)**.** *For every circuit C acting on n qubits, with T gates, for every $\delta \in (0, \frac{1}{3}), K \in \mathbb{N}$ and all $\eta > 0$ small enough there exists an interactive protocol between a verifier with constant quantum memory* **V** *and a quantum prover* **P** *with the following properties. The protocol runs in $N = O\left(\frac{K\cdot(n^5+n^2T^3+T^5)}{\eta^4}\log(1/\delta)\right)$ rounds, in each round* **P** *sends a (potentially mixed) quantum state on $O(n+T)$ qubits to* **V**. *At the end of the protocol* **V** *outputs $\perp$ when it rejects the interaction or $S = \{x_1, \ldots, x_{|S|}\}$, where $x_i \in \{0,1\}^n$, when it accepts.*

- *(Completeness) There exists $\mathbf{P}^{\mathbb{O}(*)}$ such that for every $\mathscr{D} \in \mathfrak{D}(n)$ satisfying $d_H(\mathscr{D}, \mathscr{D}_C) \leq \eta$ the following holds. With probability $1-\delta$ over the randomness in the protocol $\mathbf{P}^{\mathbb{O}(\mathscr{D})}$ succeeds, $S \sim_{i.i.d.} \mathscr{D}^{|S|}$ and $|S| \geq \Omega(K)$.*
- *(Soundness) For every* **P** *that succeeds with probability at least $\frac{2}{3}$ we have $S \sim_{i.i.d.} (\mathscr{D}^A)^{|S|}$ and $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta^{1/4})$.*

Proof. We first address completeness of the protocol and then move to soundness.

Completeness. Recall that the **P** that was guaranteed to exist in Theorem 8 was just sending state $\left|\psi_{\mathscr{D}}\right\rangle_{\text{adv}}$ to **V**. Recall that we denote by $T' = n+T+3$ the number of gates in G and by n' the number of qubits that are sent by **P** in each round. As we discussed the natural extension of this strategy to the constant memory model is for **P** to prepare the history state $\left|\phi_{\mathscr{D}}\right\rangle_{\text{comp}}$ of $\left|\psi_{\mathscr{D}}\right\rangle_{\text{adv}}$ and send it to **V**. As $N = O\left(\frac{K\cdot(n^2T'^3+T'^5)}{\eta^4}\log(1/\delta)\right) = O\left(\frac{K\cdot(n^5+n^2T^3+T^5)}{\eta^4}\log(1/\delta)\right)$ we get by Lemma 9 that with probability $1-\delta$

- the estimate of the energy $\frac{\gamma\cdot L}{n_1} \leq \frac{\eta^2}{4T'^3}$ as $\left\langle\phi_{\mathscr{D}}\right| H_G \left|\phi_{\mathscr{D}}\right\rangle = 0$,
- $|S| = \Omega(K)$ as in this case $\text{Tr}(|0\rangle\langle 0|_{\text{clock}} \otimes |0\rangle\langle 0|_{\text{out}} \otimes |0^n\rangle\langle 0^n|_{\text{aux}}\rho_A)$, which is the probability of getting a sample if the type is 1 is equal to $\left\langle\phi_{\mathscr{D}}\right| \Pi^{(T')}_{\text{clock}} \left|\phi_{\mathscr{D}}\right\rangle = \frac{1}{T'+1}$,
- $p \geq \frac{\langle\phi_{\mathscr{D}}|\Pi^{(0)}_{\text{clock}}\Pi^{(1)}_{\text{out}}|\phi_{\mathscr{D}}\rangle}{\langle\phi_{\mathscr{D}}|\Pi^{(0)}_{\text{clock}}|\phi_{\mathscr{D}}\rangle} - \frac{\eta^2}{4} \geq f(d_H(\mathscr{D}_C, \mathscr{D})) - \frac{\eta^2}{4} \geq 1-2\eta^2$, thus the two checks are verified and the interaction is accepted. By definition $S \sim_{\text{i.i.d.}} (\mathscr{D})^{|S|}$. Thus completeness is verified.

Soundness. We follow the structure of the proof of Theorem 16, which is the analog of this theorem for a fully quantum verifier. Let ρ_A be the density matrix representing the state sent by **P**. By Lemma 9 we know that with probability $1-\delta/2$ the energy estimate is within an additive error of $\frac{\eta^2}{4T^3}$ and p is estimated within an additive error of $\frac{\eta^2}{4}$. So as **P** succeeds with probability $\frac{2}{3}$ then by the union bound and the fact that $\frac{1}{3}+\frac{\delta}{2}<1$ we get that $\mathrm{Tr}(H_G\rho_A)\leq \frac{\eta^2}{2T^3}+\frac{\eta^2}{4T^3}=\frac{\eta^2}{T^3}$ and $p\geq 1-2\eta^2-\frac{\eta^2}{4}\geq 1-3\eta^2$. With that we can apply Corollary 1 and conclude that $d_H(\mathscr{D}^A,\mathscr{D}_C)\leq O(\eta^{1/4})$. □

5.7 Classical Verifier

Now we are ready to move to the last model we consider in this work, namely the one where **V** is fully classical and the communication is also classical. Recall that in Section 5.6 we designed the protocol by forcing **P** to send to **V** a history state $\left|\phi_{\mathscr{D}^A}\right\rangle_{\mathrm{comp}}$ corresponding to a distribution satisfying $d_H(\mathscr{D}_C,\mathscr{D}^A)\leq O(\eta^{1/4})$. To extend this protocol to the classical model we first force **P** to commit to a state ρ, a state that will in some sense correspond to $\left|\phi_{\mathscr{D}^A}\right\rangle_{\mathrm{comp}}$ and then force **P** to measure this state in the basis chosen by **V**. By making the prover to measure his qubits honestly we get a version of constant quantum memory Protocol (Figure 5.5) in which all the quantum computation is done on the prover side and the verifier and the communication is completely classical.

To achieve our goal we will use cryptographic tools. As the protocol will rely on the hardness of computational problems, our soundness results will only address provers that are computationally bounded, namely only provers in the QPT class.

Next, we give a high-level overview of the protocol. An honest prover **P** is given a local Hamiltonian corresponding to G and computes the ground state of the Hamiltonian, i.e. the history state $\left|\phi_{\mathrm{hist}}\right\rangle$. Later the prover is asked to commit to this state before the protocol proceeds with the interactive stage, in which the prover is asked to measure qubits of the state he has committed to either in computational or the Hadamard basis, and send the outcomes to the verifier. At each iteration, the verifier decides to do one of the following 3,

- estimate the energy of the state the prover has committed to,
- estimate the probability of the output of the circuit being 1,
- collect a sample from the distribution corresponding to the prover's state.

The description of this protocol is given in Figure 5.6. We note that the results presented in this section heavily rely on [Mah18]. Some of the technical lemmas are not proven here. We refer the reader to [Mah18] for said proofs.

Note 5. *We stress that with this protocol one can only retrieve samples from measurements done in the Z basis. The distribution of samples collected in the protocol when* **V** *asks for the X basis*

*measurements are **not** in general equal to the distribution of measuring the state of* **P** *in the X basis. This means that if our protocol required samples from the distribution corresponding to the X measurements it is not clear if it could be realized in the fully classical model.*

Similar to [Mah18] we require a more refined version of the circuit-to-Hamiltonian reduction, namely we require our Hamiltonians to be 2-local and of the form $\sum_{i,j} -\frac{J_{i,j}}{2}(\sigma_{X,i}\sigma_{X,j} + \sigma_{Z,i}\sigma_{Z,j})$.

Theorem 11 ([BL08])**.** *For any integer $n \geq 1$ there exists $n' = poly(n)$, $a(n)$ and $\delta \geq 1/poly(n)$ such that given a T-gate quantum circuit G, there exists an efficiently computable real-weighted Hamiltonian H_G in $XX - ZZ$ form, such that,*

- *(completeness) If G accepts x with probability at least $2/3$, then $\lambda_0(H_G) \leq a$.*
- *(soundness) If G accepts x with probability at most $1/3$, then $\lambda_0(H_G) > a + \delta$.*

As proved in Section 5.6, by modifying the standard circuit-to-Hamiltonian reduction, we can show that "for any $|\phi\rangle$ such that $\langle\phi| H_G |\phi\rangle < \epsilon$ the distribution of the measurement outcome of the first qubit of $|\phi\rangle$ (conditioned on the clock register being T') is ϵ close to the distribution of what G would output". We refer the reader to [CLLW22] to see how the same guarantees can be derived with a 2-local Hamiltonian.

We proceed by stating the completeness and soundness properties of this protocol and providing a proof sketch.

Prover's Observables

In order to prove γ is an accurate estimate of the energy using a similar argument to Lemma 9, we have to prove the $\mathbb{E}[\gamma] = \mathrm{Tr}(H_G\rho)$, where in a sense ρ is the prover's state. Letting (X_i, Z_i) be the observables of **P** which determine the value of the i^{th} response, we require an isometry which *teleports* these observables to $\sigma_{X,i}, \sigma_{Z,i}$ as the Hamiltonian is penalizing the bad configurations of the state with respect to $\sigma_{X,i}, \sigma_{Z,i}$. Let us assume the prover's state is in a Hilbert space $\mathscr{H} \otimes \mathscr{H}_{\text{env}}$, where he might share some entanglement with the environment.

Based on how the estimates are updated in the protocol the natural way to define the observables that **P** measures, would be,[10]

$$Z(a) = \sum_{x_1,\ldots,x_{n'} \in \{0,1\}^m} (-1)^{b(x)\cdot a} |x_1\rangle\langle x_1| \otimes \ldots |x_{n'}\rangle\langle x_{n'}| \otimes \mathbb{I}_P$$
$$X(a) = \sum_{d_1,\ldots,d_{n'}} (-1)^{\sum a_i(d_i(x_{i,0}+x_{i,1}))} U^\dagger(|d_1\rangle\langle d_1| \otimes \ldots |d_{n'}\rangle\langle d_{n'}| \otimes \mathbb{I}_P)U$$

Basically in our modeling of actions of **P**, if the challenge bit is $b = 0$, **P** measures his state in the computational basis in order to get the preimages, and if $b = 1$, he applies an arbitrary unitary U, followed by a Hadamard measurement to retrieve the d values.

[10] here we will use b by absuing the notation instead of $(b_i(x_i))_{i\in[n]}$

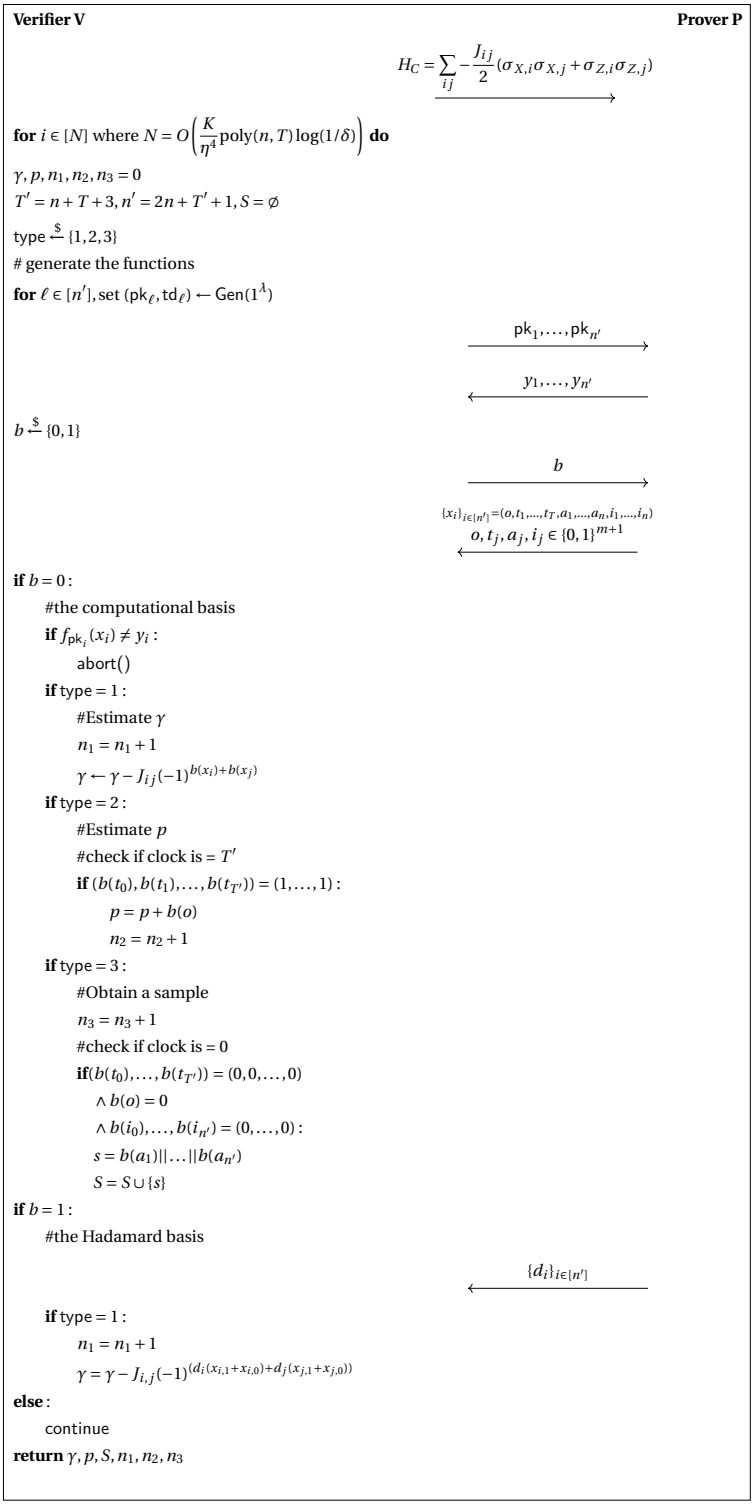


Figure 5.6: The description of the classical verifier protocol. Notice that x_i values are $m+1$ bits long each, e.g. o contains the measurement outcome of the output register, plus the remaining m bits of the input to f_{pk_1}.

Now we apply the canonical isometry from lemma 1 on the two observables, given as follows,

$$V\left|\psi\right\rangle = (\frac{1}{2^{n'}}\sum_{a,b}\mathbb{I}\otimes\sigma_X(a)\sigma_Z(b)\otimes X(a)Z(b))\left|\phi^+\right\rangle^{\otimes n'}\left|\psi\right\rangle$$

and prove that it interacts with the observables X and Z as intended, i.e. forms a commutative diagram. Note that for V to be an isometry X and Z do not need to anti-commute, but the diagram need not commute if X and Z do not anti-commute.

We start by defining a notion of an extracted qubit.

Definition 16 (Extracted Qubits)**.** *Let* **P** *be a prover playing in Protocol 5.6, X, Z defined and let V be the canonical isometry sending (X,Z) to (σ_X,σ_Z). Let $\left|\phi\right\rangle \in \mathcal{H}\otimes\mathcal{H}_{env}$ be the state of the prover after sending the y_i values. We call the reduced density of the first qubit of $V\left|\phi\right\rangle$ the extracted qubits of the prover, and we denote it by ρ.*

Now we restate lemma 7.4 from [Vid22] shows that the observables and the isometry form an almost commutative diagram.

Fact 3 ([Vid22])**.** *Let* **P** *be any QPT prover, ρ be their extracted qubit. We have,*[11]

- $\forall b\in\{0,1\}^{n'},\quad Tr(\sigma_Z(b)\rho) = \left\langle\psi\right| Z(b)\left|\psi\right\rangle$
- $\forall b\in\{0,1\}^{n'},\quad Tr(\sigma_X(b)\rho) = \frac{1}{2^{n'}}\sum_a(-1)^{a\cdot b}\left\langle\psi\right| Z(b)X(a)Z(b)\left|\psi\right\rangle$

Previously we mentioned that one can retrieve samples by asking the prover to measure their state in the computational basis, the first bullet exactly corresponds to this scenario. Intuitively what it tells us is that the distribution of the $b(x)$, for x values returned by **P**, is identical to the distribution of the measurement outcomes of the extracted qubit ρ in the computational basis. However, in the case of the Hadamard basis, the matter is more subtle as the distribution is "twirled". As long as we only care about collecting samples via Z measurements, the twirl operator does not cause us any issues, as we will show that it would not affect the energy estimate in the protocol.

In order to follow the proofs done in [Mah18] we require our function family $\mathscr{F}$ to have the adaptive hardcore property and moreover be collapsing, where both properties are defined in sections 1.4.5 and 1.5. Note that, the collapsing property holds naturally for the LWE-based construction given in [Mah18][12].

We proceed by stating and proving the completeness of the protocol.

[11] $\sigma_W(a) = \Pi_{i \text{ s.t. } a_i=1}\sigma_{W,i}$ the X or Z measurement of indices such that $a_i = 1$

[12] the authors consider two families of functions, a 2-to-1 family and a bijective family, and prove that based on the hardness of LWE, no adversary can distinguish between the two. This is another phrasing of the collapsing property.

Completeness

In this section we describe an honest prover strategy. We describe a prover $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ that wins in protocol 5.6 with probability 1, and provides us with samples from $\mathscr{D}$. Recall that in the constant quantum memory protocol, $\mathbf{P}$ first creates a history state $|\phi_{\mathscr{D}}\rangle$ for $|\psi_{\mathscr{D}}\rangle$ and then sends $|\phi_{\mathscr{D}}\rangle$ to $\mathbf{V}$. This prover satisfies the completeness property. For the classical model we will show that a prover who commits to the same history state also satisfies completeness.

Theorem 12 (Completeness)**.** *There exists a QPT prover $\mathbf{P}^{\mathsf{O}(*)}$, such that for any distribution $\mathscr{D} \in \mathfrak{D}(n)$, any λ-collapsing claw-free family $\mathscr{F}$, $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ wins in Protocol 5.6 with probability 1 and we have:*

- $S \sim \mathscr{D}^{|S|}$

The completeness of this protocol is in some sense easier to prove than the completeness of the protocols described in the previous sections. The reason for this is that the protocol does not abort when the estimates are not satisfying the desired bounds. We proceed by describing the strategy for the honest prover and the proof of completeness.

Proof. Let us denote $2n + T' + 1$ by n'. The honest prover will create a state such that each bit b_i would correspond to the measurement of a qubit from the history state. They extend the state with zeros in the following way.

$$|\phi_{\text{hist}}\rangle \left|0^{mn'}\right\rangle_X = \sum_{b_1,\ldots,b_{n'}} \alpha_{b_1,\ldots,b_{n'}} |b_1\rangle |0^m\rangle \ldots |b_{n'}\rangle |0^m\rangle \tag{5.4}$$

By applying QFT on the 0 registers we get the state,

$$|\phi'\rangle = \frac{1}{\sqrt{2^{mn'}}} \sum_{b_1,\ldots,b_{n'}} \alpha_b \Big(\sum_{z \in \{0,1\}^{mn'}} |b_1\rangle |z_1\rangle \ldots |b_{n'}\rangle |z_{n'}\rangle \Big) \tag{5.5}$$

We add a zero register to $|\psi\rangle$ and evaluate f_{pk_i} on the superpositions to get the state,

$$|\phi''\rangle = \frac{1}{\sqrt{2^{mn'}}} \sum_{b_1,\ldots,b_{n'}} \alpha_b \Big(\sum_{x \in \{0,1\}^{mn'}} |b_1\rangle |z_1\rangle \left|f_{\mathsf{pk}_1}(b_1, z_1)\right\rangle \ldots |b_{n'}\rangle |z_{n'}\rangle \left|f_{\mathsf{pk}_{n'}}(b_{n'}, z_{n'})\right\rangle \Big) \tag{5.6}$$

$\mathbf{P}$ proceeds by measuring the image registers to get values $y_1, \ldots, y_{n'}$. The state after obtaining this measurement outcome will be,

$$|\phi_{\mathbf{P}}\rangle = \sum_b \alpha_b |b_1\rangle \left|x_{1,b_1}\right\rangle \left|y_1\right\rangle \ldots |b_{n'}\rangle \left|x_{n',b_{n'}}\right\rangle \left|y_{n'}\right\rangle \tag{5.7}$$

where (b_i, x_{i,b_i}) is the b_i-labeled preimage of y_i under f_{pk_i}.

Upon receiving challenge 0, $\mathbf{P}$ measure the state $|\phi_{\mathbf{P}}\rangle$ in computational basis and sends $x_1, \ldots, x_n = (b_1, x_{1,b_1}), \ldots, (b_{n'}, x_{n',b_{n'}})$ values to $\mathbf{V}$. From equation 5.7 one can deduce that

$b_1(x_1),\ldots,b_{n'}(x_{n'})$ as in the protocol is distributed identically to the outcome of the measurement of the history state in computational basis. By construction $\mathbf{P}$ always succeeds in the preimage check, hence, wins with probability 1.

As the outcomes of measuring the b registers of state $|\phi_{\mathbf{P}}\rangle$ are distributed identically to the measurement outcomes of the history state $|\phi_{\text{hist}}\rangle$ we have $S \sim \mathscr{D}^{|S|}$, following the completeness proof from Theorem 10. □

In fact the completeness can be modified so that it captures the fact that the estimates computed in the protocol are close to the actual energy and outcome probability. We state the theorem here, but as similar statements are proven in the soundness we avoid repeating the proof here.

Theorem 13 (Completeness 2)**.** *There exists a QPT prover $\mathbf{P}^{\mathsf{O}(*)}$, such that for any distribution $\mathscr{D} \in \mathfrak{D}(n)$, any family of λ-collapsing claw-free family $\mathscr{F}$, $\mathbf{P}^{\mathsf{O}(\mathscr{D})}$ wins $N = O(\frac{K}{\eta^4}\text{poly}(n,T)\log(1/\delta))$ iterations of Protocol 5.6 with probability 1, we have that with probability at least $1-\delta$,*

- $|S| \geq \Omega(K)$,
- $p/n_2 \geq 1 - 2d_H^2(\mathscr{D},\mathscr{D}_C)$,[13]
- $\gamma/n_1\binom{n'}{2} \in [q - \frac{\eta^2}{2T^3}, q + \frac{\eta^2}{2T^3}]$, *where* $q = \langle\phi_{hist}| H_G |\phi_{hist}\rangle$,
- $S \sim (\mathscr{D})^{|S|}$.

Soundness

Now that we have established an honest prover strategy, the only thing left is to prove that for any prover who wins the game with a high probability, the verifier $\mathbf{V}$ would collect samples from a distribution close to the $\mathscr{D}_C$.

The key fact to prove the soundness of the protocol is that the values x_i and d_i are somewhat correlated with the measurement outcomes of the i^{th} qubit of the prover's extracted qubit.

Fact 4 ([Mah18])**.** *Let $\mathscr{F}$ be a collapsing claw-free family and let $\mathbf{P}$ be any QPT prover who wins in Protocol 5.6 with probability 1, let ρ be the prover's extracted qubits, B_i and D_i the outcome of measuring the i^{th} qubit of ρ in the computational and the Hadamard basis respectively. For any parity $\chi : \{0,1\}^{n'} \to \{-1,+1\}$ we have,*

- *(computational basis measurement) $\chi(B_1,\ldots,B_{n'})$ is identically distributed to*

$$\chi(b_1(x_1),\ldots,b_{n'}(x_{n'})).$$

[13] This is done similar to the protocol described in Figure 5.5

- *(the Hadamard basis measurement)* $\chi(D_1,\ldots,D_{n'})$ *is computationally indistinguishable from* $\chi(d_1\cdot(x_{1,0}+x_{1,1}),\ldots,d_{n'}\cdot(x_{n',0}+x_{n',1}))$.

Lemma 10. *For any T-gate quantum circuit C, and its corresponding T'-gate comparison circuit G, let $\mathscr{F}=\{f_{\mathsf{pk}}\}$ be a family of claw-free functions satisfying the collapsing property, and let H_G be an $n'=2n+T'+1$ qubit Hamiltonian corresponding to G. For any QPT prover $\mathbf{P}$ let ρ be the reduced density of the extracted qubits of $\mathbf{P}$, and let $\mathscr{D}^A$ be the distribution of outcomes of measuring the adv register of ρ conditioned on the measurement outcome of clock, out and aux register being 0 in the computational basis. If $\mathbf{P}$ wins in protocol 5.6 with probability 1 we have,*

- $\mathbb{E}[\gamma/n_1\binom{n'}{2}]\approx Tr(H_G\rho)$,
- $\mathbb{E}[p/n_2]=\frac{Tr(|T'\rangle\langle T'|_{clock}\otimes|1\rangle\langle 1|_{out}\rho)}{Tr(|T'\rangle\langle T'|_{clock}\rho)}$,
- $S\sim(\mathscr{D}^A)^{|S|}$.

Proof. To prove this theorem we will be using Fact 4. First we prove the properties only relying on the computational measurements, namely properties about p and S. Let us focus on distribution of S first.

A sample is collected if type = 1, the challenge bit b is equal to 0 and $b(o), b(\text{clock}), b(\text{aux})$ are all 0. Due to Fact 4 this is equivalent to when the outcome of measuring the aux, clock and out registers of ρ are 0. Conditioned on this happening the sample collected has the exact same distribution as measuring the adv register of ρ, which is equivalent to $\mathscr{D}^A$.

The estimate p is increased by $b(o)$, when type = 2, the challenge bit is 0 and

$$b(t_0),\ldots,b(t_{T'})=(1,\ldots,1).$$

Conditioned on the clock being T', The expectation of $b(o)$ is $\mathrm{Tr}(\left|T'\right\rangle\left\langle T'\right|_{\text{clock}}\otimes|1\rangle\langle 1|_{\text{out}}\rho)$ due to fact 4. Hence we have $\mathbb{E}[p]=n_2\frac{\mathrm{Tr}(|T'\rangle\langle T'|_{\text{clock}}\otimes|1\rangle\langle 1|_{\text{out}}\rho)}{\mathrm{Tr}(|T'\rangle\langle T'|_{\text{clock}}\rho)}$.

The next thing to prove is that the energy estimate has the desired expectation. If we consider the n_1 rounds in which we change the energy estimate, the expectation of the amount of change done to γ is equal to:

$$-\frac{1}{2\binom{n'}{2}}\sum_{i,j}J_{i,j}(-1^{b_i(x_i)+b_j(x_j)}+-1^{d_i(x_{i,0}+x_{i,1})+d_j(x_{j,0}+x_{j,1})})$$

For $b_i(x_i)$ and $b_j(x_j)$, we know that these random variables are distributed identically to measurement of ρ in computational basis. The only issue is that $d_i(x_{i,0}+x_{i,1})$ is not distributed identically to Hadamard measurement of ρ, but rather is computationally indistinguishable from it.

However for any parity χ if the distance between the expectations of $\chi(d_i(x_{i,0}+x_{i,1}))$ and χ applied on the measurement outcomes of ρ in the Hadamard basis is negligible in λ; as otherwise an adversary could distinguish between the two by random sampling using only $O(1/poly(\mu))$ samples. Hence we have,

$$\mathbb{E}[J_{i,j}(-1)^{b_i(x_i)+b_j x_j}] = J_{i,j}\mathrm{Tr}(\sigma_{Z,i}\sigma_{Z,j}\rho)$$
$$\mathbb{E}[J_{i,j}(-1)^{d_i(x_{i,0}+x_{i,1})+d_j(x_{j,0}+x_{j,1})}] = J_{i,j}(\mathrm{Tr}(\sigma_{X,i}\sigma_{X,j}\rho) \pm ngl(\lambda))$$

Hence we have that $\mathbb{E}[\gamma] \approx n_1 \frac{1}{\binom{n'}{2}}\mathrm{Tr}(H_G\rho)$ as desired. □

Theorem 14 (Perfect Prover Soundness)**.** *For any security parameter λ, any T-gate circuit C acting on n-qubits, the protocol defined in Figure 5.6 has the following properties. It is an interactive protocol $(\mathbf{V},*)$ between a classical verifier $\mathbf{V}$ and a quantum prover. For any QPT prover $\mathbf{P}$, let ρ be the reduced density of the extracted qubits of $\mathbf{P}$, and $\mathscr{D}^A$ be the distribution of outcomes of measuring the adv register of ρ in the computational basis conditioned on the measurement outcome of clock, out and aux registers being 0. If $\mathbf{P}$ wins $N = O(\frac{K}{\eta^4}poly(n,T)\log(1/\delta))$ iterations of the protocol with probability 1 and $\frac{\gamma}{n_1}\binom{2n+T+1}{2} \leq \frac{\eta^2}{2T^3}$ and $\frac{p}{n_2} \geq 1-2\eta^2$, then with probability $1-\delta-\mu(\lambda)$, we have,*

- $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta^{1/4})$,
- $S \sim (\mathscr{D}^A)^{|S|}$.

where μ is a negligible function.

Note 6. *The guarantee expressed in the last sentence of Theorem 14 might seem mysterious at first. Note however that the conditions contained there are equivalent to the checks performed at the last step in the constant quantum memory protocol from Figure 5.5. The fact that the checks are contained in the statement of the theorem and not in the protocol itself allows us to analyze perfect provers only and simplifies the presentation considerably.*

Proof of Theorem 14. Let G be the T' comparison circuit of C and let H_G be the corresponding 2-local Hamiltonian acting on $n' = 2n+T'+1$ qubits.

Applying Lemma 8 we have that with probability $1-e^{-\frac{N}{18}}$ we have $n_1 \geq \Omega(N)$. From Lemma 10 we have,

$$\mathbb{E}\left[\frac{\gamma}{n_1}\binom{n'}{2}\right] \approx \mathrm{Tr}(H_G\rho) \tag{5.8}$$
$$\mathbb{E}\left[\frac{p}{n_2}\right] = \frac{\mathrm{Tr}(|T'\rangle\langle T'|_{\mathrm{clock}} \otimes |1\rangle\langle 1|_{\mathrm{out}}\rho)}{\mathrm{Tr}(|T'\rangle\langle T'|_{\mathrm{clock}}\rho)} \tag{5.9}$$

Using Fact 2 we get,

$$\mathbb{P}\left[\left|\frac{\gamma}{n_1}\binom{n'}{2}-\mathrm{Tr}(H_G\rho)\right|\geq\frac{\eta^2}{2T'^3}\right]\leq 2e^{-\frac{\eta^4 n_1}{8\binom{n'}{2}^2 T'^6 J}} \tag{5.10}$$

$$\mathbb{P}\left[\left|\frac{p}{n_2}-\frac{\mathrm{Tr}(\left|T'\right\rangle\left\langle T'\right|_{\mathrm{clock}}\otimes|1\rangle\langle 1|_{\mathrm{out}}\rho)}{\mathrm{Tr}(|T'\rangle\langle T'|_{\mathrm{clock}}\rho)}\right|\geq\frac{3\eta^2}{2}\right]\leq 2e^{-\frac{9\eta^4 n_2}{8}}, \tag{5.11}$$

where $J=\sup_{i\neq j\in[n']}\{|J_{i,j}|\}$.

If we use the hypothesis of the theorem, (5.10) and (5.11) we get that with probability $1-\frac{\delta}{8}$,

$$\mathrm{Tr}(H_G\rho)\leq\frac{\eta^2}{T'^3} \tag{5.12}$$

$$\frac{\mathrm{Tr}(\left|T'\right\rangle\left\langle T'\right|_{\mathrm{clock}}\otimes|1\rangle\langle 1|_{\mathrm{out}}\rho)}{\mathrm{Tr}(|T'\rangle\langle T'|_{\mathrm{clock}}\rho)}\geq 1-2\eta^2-\frac{3\eta^2}{2}\geq 1-\frac{7\eta^2}{2} \tag{5.13}$$

By eq. (5.13), the probability of the measurement outcome of the out register being 1, when the clock is T' is at least $1-\frac{7\eta^2}{2}$. By employing Corollary 1 we have that with probability $\frac{1\pm 7\eta}{T'+1}$, $d_H(\mathscr{D}_C,\mathscr{D}^A)\leq O(\eta^{1/4})$.

By Fact 2 we have that $n_2=\Omega(\frac{N}{T'})$ with probability $1-\frac{\delta}{20}$ so it is enough to set the number of rounds to be $N=O(\frac{1}{\eta^4}\log(1/\delta)\binom{n'}{2}^2 T'^7 J)=O(\frac{1}{\eta^4}\log(\frac{1}{\delta})\mathrm{poly}(T,n))$ for (5.11) to hold with probability $\leq\delta/10$. If we apply the union bound over all failure events we get that all the conditions will be satisfied with probability at least $1-\delta$, hence with probability $1-\delta$ we get $d_H(\mathscr{D}_C,\mathscr{D}^A)\leq O(\eta^{1/4})$.

The second bullet follows directly from Lemma 10. □

Discussion We have proven the soundness of our protocol only in the perfect prover setting. The problem with this statement is that it can not be verified that the prover is winning with probability 1. Also, the soundness guarantee is different from the previous sections as the game does not abort when the estimates do not satisfy the bound. The reason we modified the game in this manner is that if the game aborted after checking the bounds, even the honest prover would not have won the game with probability 1, as there is a small probability that the estimates computed in the protocol are far from the expected value.

However, it is possible to achieve a stronger soundness guaranties, similar to Theorem 10. This requires more adjustments to the protocol which allows one to prove the soundness for a non-perfect prover by following a similar path as the one in Claim 7.1 of [Mah18], where a reduction from the non-perfect prover to a perfect prover is given.

6 Conclusion and Further Work

6.1 Conclusion

In this thesis, We focused on exploring the realm of cryptography when confronted with quantum computation. Our objective was to offer two distinct viewpoints on this matter. Initially, we examined classical cryptographic objects that ensure security even when facing quantum adversaries. Subsequently, we delved into cryptographic primitives specifically tailored for quantum entities.

The initial two contributions presented in this thesis (Chapter 2 and 3) revolved around an analysis of a *back to the roots* approach to designing quantum-secure digital signatures. Specifically, we explored the construction of signature schemes based on symmetric-key cryptographic components. Our analysis focused on the PICNIC signature scheme [CDG+17], which is built using the MPC-in-the-head paradigm [IKOS07]. We demonstrated that reducing the multiplication count of the underlying cipher used in PICNIC results in a significant loss of security. We summarize the findings from these two chapters with the following fundamental principle:

"The security of signature schemes derived from the MPC-in-the-head paradigm is directly influenced by the multiplication count of the underlying symmetric-key primitive."

The third outcome we present in this thesis (Chapter 4) focused on how quantum communication affects the relationship between *the five worlds of Impagliazzo*. We establish that public-key encryption, which serves as the central primitive in Cryptomania, can indeed be constructed using Minicrypt objects when quantum states are allowed as public keys. Furthermore, we demonstrate how this notion of public-key encryption can be built based on assumptions that might even belong to Algorithmica, a world where classical cryptography is no longer viable. Additionally, we show that while quantum public-key encryption (qPKE) can be constructed using weaker assumptions compared to classical cryptography, it still falls short of providing information-theoretic security. This observation underscores the blurred

boundaries between Impagliazzo's five worlds when quantum computation is considered and emphasizes the need for modifying the definitions of these worlds in a quantum context.

In Chapter 5, we explored the possibilities offered by quantum information in leveraging Cryptomania assumptions. Specifically, we investigate problems that are considered classically intractable even under strong Cryptomania assumptions. By building upon the concept of delegating quantum computation, we provide cryptographic guarantees for classification tasks, where the test time examples can be adversarially perturbed, without any restrictions on the perturbations. The result presented in this chapter established that in a quantum learning model, assuming the hardness of the LWE problem, classical lower bounds previously established in [GKKM20] can be surpassed in certain regimes.

6.2 Future Work

Regarding the outcomes presented in Chapters 2 and 3, a highly desirable objective is to establish a direct relationship between the circuit complexity of the verification procedure for a general NP-statement and the security of the MPC-in-the-head signature derived from it. Obtaining fine-grained bounds on this quantity would enable a more systematic selection of parameters for signatures constructed using this framework.

The findings discussed in Chapter 4 open up new avenues for constructing quantum cryptographic primitives based on weaker assumptions compared to classical requirements. One immediate open problem arising from these results is the exploration of building a public-key infrastructure (PKI) using similar principles. One significant challenge in achieving this goal is the authentication of public keys, as it has been shown that authenticating arbitrary quantum states is impossible [BCG+02b]. However, the specific structure of the keys might allow for this task to be achievable only using a classical authenticated channel [KMNY23].

The analysis presented in the first three chapters gives rise to an intriguing problem: the exploration of fine-grained circuit complexity in the construction of quantum cryptography primitives. One particularly intriguing question that remains unresolved is whether objects like pseudorandom state generators (PRSGs) can be implemented using constant-depth quantum circuits. Consequently, a captivating research direction involves investigating both lower and upper bounds for the circuit complexities required to realize various quantum cryptographic objects.

The final direction for future work that we propose is to investigate the question: *"Where are Cryptomania assumptions necessary in a quantum world?"* While quantum communication allows the construction of cryptographic primitives that would traditionally demand heavier classical machinery, it remains uncertain whether certain tasks still necessitate Cryptomania assumptions. A notable example is qubit test protocols and randomness certification from [BCM+18]. In a recent breakthrough, Zhandry and Yamakawa [YZ22] demonstrated that random oracles suffice to build a publicly verifiable test of quantumness protocol. An

intriguing open problem is to explore whether this approach can be adapted to construct a qubit test and classical verification of quantum computation.

A Appendicies for Chapter 5

A.1 Ommited Proofs

Lemma 11. *For every $|\psi_{\mathscr{D}^A}\rangle$ and C the probability of obtaining outcome $|1\rangle$ when measuring the out register of $G|0\rangle_{out}|\psi_{\mathscr{D}^A}\rangle_{adv}|0^{\otimes n}\rangle_{aux}$ is equal to*

$$\frac{1}{2}\left(1+\left|\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle\right|^2\right).$$

Proof. We analyze the evolution of the state

$$\begin{aligned}
&(\mathrm{NOT}\otimes\mathbb{I}\otimes\mathbb{I})(H\otimes\mathbb{I}\otimes\mathbb{I})(\mathrm{CSWAP})(H\otimes\mathbb{I}\otimes\mathbb{I})(\mathbb{I}\otimes\mathbb{I}\otimes U_C)|0\rangle|\psi_{\mathscr{D}^A}\rangle|0^{\otimes n}\rangle\\
&=(\mathrm{NOT}\otimes\mathbb{I}\otimes\mathbb{I})(H\otimes\mathbb{I}\otimes\mathbb{I})(\mathrm{CSWAP})(H\otimes\mathbb{I}\otimes\mathbb{I})|0\rangle|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle\\
&=(\mathrm{NOT}\otimes\mathbb{I}\otimes\mathbb{I})(H\otimes\mathbb{I}\otimes\mathbb{I})(\mathrm{CSWAP})\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle\\
&=(\mathrm{NOT}\otimes\mathbb{I}\otimes\mathbb{I})(H\otimes\mathbb{I}\otimes\mathbb{I})\frac{1}{\sqrt{2}}\left(|0\rangle|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle+|1\rangle|\psi_{\mathscr{D}_C}\rangle|\psi_{\mathscr{D}^A}\rangle\right)\\
&=(\mathrm{NOT}\otimes\mathbb{I}\otimes\mathbb{I})\frac{1}{2}\left((|0\rangle+|1\rangle)|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle+(|0\rangle-|1\rangle)|\psi_{\mathscr{D}_C}\rangle|\psi_{\mathscr{D}^A}\rangle\right)\\
&=\frac{1}{2}\left(|1\rangle\left[|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle+|\psi_{\mathscr{D}_C}\rangle|\psi_{\mathscr{D}^A}\rangle\right]+|0\rangle\left[|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle-|\psi_{\mathscr{D}_C}\rangle|\psi_{\mathscr{D}^A}\rangle\right]\right).
\end{aligned}$$

The probability of obtaining outcome $|1\rangle$ when measuring the out register in the Z basis is

then

$$\begin{aligned}
&\frac{1}{4}\left[\left(\langle\psi_{\mathscr{D}^A}|\langle\psi_{\mathscr{D}_C}| + \langle\psi_{\mathscr{D}_C}|\langle\psi_{\mathscr{D}^A}|\right)\left(|\psi_{\mathscr{D}^A}\rangle|\psi_{\mathscr{D}_C}\rangle + |\psi_{\mathscr{D}_C}\rangle|\psi_{\mathscr{D}^A}\rangle\right)\right] \\
&= \frac{1}{4}[\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}^A}\rangle\langle\psi_{\mathscr{D}_C}|\psi_{\mathscr{D}_C}\rangle + \langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle\langle\psi_{\mathscr{D}_C}|\psi_{\mathscr{D}^A}\rangle + \\
&+ \langle\psi_{\mathscr{D}_C}|\psi_{\mathscr{D}^A}\rangle\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle + \langle\psi_{\mathscr{D}_C}|\psi_{\mathscr{D}_C}\rangle\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}^A}\rangle] \\
&= \frac{1}{4}\left[2\|\psi_{\mathscr{D}^A}\|^2\|\psi_{\mathscr{D}_C}\|^2 + 2\langle\psi_{\mathscr{D}_C}|\psi_{\mathscr{D}^A}\rangle\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle\right] \\
&= \frac{1}{2}\left[1 + |\langle\psi_{\mathscr{D}^A}|\psi_{\mathscr{D}_C}\rangle|^2\right].
\end{aligned}$$

□

Lemma 5. Apply Lemma 11 and the $1 - d_H^2(\mathscr{P},\mathscr{Q}) = \sum_{x\in\{0,1\}^n}\sqrt{\mathscr{P}(x)\mathscr{Q}(x)}$ identity. □

Lemma 6. For $b \in \{0,1\}$, n_b can be seen as a sum of random Bernoulli variables $\{x_i\}_{i\in[N]}$ with parameter $1/2$. Then, by Fact 2, we get that $\mathbb{P}[|\frac{n_b}{N} - \frac{1}{2}| > \frac{1}{4}] \leq 2e^{-\frac{N}{32}} \leq \frac{\delta}{2}$. We finish by applying the union bound over the error events. □

A.2 Proof of Lemma 7

Proof. As we discussed we want to base our reduction on the standard circuit-to-Hamiltonian reduction but drop the H_{out} term. We define

$$H_G = H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}}. \tag{A.1}$$

The term H_{in} corresponds to the condition that, at step 0, the qubits are in the right state. Formally

$$H_{\text{in}} = |0\rangle\langle 0|_{\text{clock}} \otimes \left(\sum_{j\in\text{out,aux}} \Pi_j^{(1)}\right), \tag{A.2}$$

where by $j \in$ out, aux we mean iterating over all the qubits in these registers. Informally speaking, we add a penalty whenever a qubit in registers out or aux is in state $|1\rangle$ while the clock is in state $|0\rangle_{\text{clock}}$.

The term H_{prop} guarantees the propagation of quantum states through the circuit. Formally

$$H_{\text{prop}} = \sum_{j=1}^{T'} H_j, \tag{A.3}$$

$$H_j = -\frac{1}{2}|j\rangle\langle j-1|_{\text{clock}} \otimes G_j - \frac{1}{2}|j-1\rangle\langle j|_{\text{clock}} \otimes G_j^\dagger + \frac{1}{2}(|j\rangle\langle j|_{\text{clock}} + |j-1\rangle\langle j-1|_{\text{clock}}) \otimes I.$$

We will define H_{clock} later. We could realize it with $O(\log(T')$ qubits but then our Hamiltonian would be $O(\log(T'))$-local. But we aim for a 5-local Hamiltonian. We explain how to address

this issue towards the end of this section. Because of this we will assume for now that H_{clock} does not appear in (A.1).

For the analysis we follow [KSV02]. It will be useful to consider a change of basis given by

$$W = \sum_{j=0}^{T'} |j\rangle\langle j|_{\text{clock}} \otimes G_j \dots G_1.$$

What we mean is that we represent the vector $|\phi\rangle$ in the form $|\phi\rangle = W|\tilde{\phi}\rangle$. Under this change the Hamiltonian is transformed into its conjugate $\widetilde{H}_G = W^\dagger H_G W$. Simple calculation verifies that $\widetilde{H}_{\text{in}} = H_{\text{in}}$ and $\widetilde{H}_{\text{prop}} = E \otimes I$, where

$$E = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \\ 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & \\ 0 & 0 & 0 & -\frac{1}{2} & 1 & \\ & & & & & \ddots \end{pmatrix}$$

Let $|\tilde{\phi}\rangle$ be such that $\langle\tilde{\phi}|\widetilde{H}_G|\tilde{\phi}\rangle \leq \frac{\epsilon}{T'}$. We will show that it is close to a history state of $|\psi_{\mathscr{D}^A}\rangle_{\text{adv}}$. Let's write $\tilde{\phi} = \sum_{j=0}^{T'} \alpha_j |j\rangle_{\text{clock}} |\xi_j\rangle_{\text{comp}}$, for $\alpha_j \in \mathbb{R}_{\geq 0}$ and

$$|\xi_0\rangle_{\text{comp}} = \sum_{s\in\{0,1\}^{n+1}} \beta_s |s[1]\rangle_{\text{out}} |\psi_s\rangle_{\text{adv}} |s[2,n+1]\rangle_{\text{aux}}$$

where $s[i,j]$ denotes the substring of s from i to j. Then by the fact that $\widetilde{H}_{\text{prop}} = E \otimes I$ we have

$$\begin{aligned}\langle\tilde{\phi}|\widetilde{H}_{\text{prop}}|\tilde{\phi}\rangle &= \frac{1}{2}\sum_{j=1}^{T'} \| \alpha_{j-1}|\xi_{j-1}\rangle_{\text{comp}} - \alpha_j |\xi_j\rangle_{\text{comp}} \|^2 \\ &\geq \frac{1}{2}\sum_{j=1}^{T'} |\alpha_{j-1} - \alpha_j|^2, \frac{1}{2}\sum_{j=1}^{T'} \min(|\alpha_{j-1})|^2, |\alpha_j|^2) \cdot \| |\xi_{j-1}\rangle_{\text{comp}} - |\xi_j\rangle_{\text{comp}} \|^2. \quad \text{(A.4)}\end{aligned}$$

The bound above gives two inequalities. Thus we get that $\max_{j\in[T']} |\alpha_{j-1} - \alpha_j|^2 \leq \frac{2\epsilon}{T'}$, which combined with the fact that $\sum_{j=0}^{T'} |\alpha_j|^2 = 1$ gives us that

$$\max_{j\in\{0,\dots,T'\}} \left| |\alpha_j|^2 - \frac{1}{T'+1} \right| \leq \frac{2\epsilon}{T'}. \quad \text{(A.5)}$$

Using (A.5) and the bound for $\frac{1}{2}\sum_{j=1}^{T'} \min(|\alpha_{j-1})|^2, |\alpha_j|^2) \cdot \| |\xi_{j-1}\rangle_{\text{comp}} - |\xi_j\rangle_{\text{comp}} \|^2$ from (A.4) we get that for $\epsilon \leq 1$

$$\sum_{j=1}^{T'} \| |\xi_{j-1}\rangle_{\text{comp}} - |\xi_j\rangle_{\text{comp}} \|^2 \leq \frac{\frac{2\epsilon}{T'}}{\frac{1}{T'+1} - \frac{2\epsilon}{T'}} \leq 4\epsilon,$$

which implies that

$$\| |\xi_0\rangle_{\text{comp}} - |\xi_{T'}\rangle_{\text{comp}} \|^2 \leq 4\epsilon T'. \tag{A.6}$$

Using the second term from H_G we also have

$$\langle\tilde{\phi}| \tilde{H}_{\text{in}} |\tilde{\phi}\rangle = \sum_{j=1}^{n} \sum_{s\in\{0,1\}^n : s[j]=1} \beta_s^2 \leq \frac{\epsilon}{T'}. \tag{A.7}$$

Note that the distribution corresponding to $|\psi_{0^n}\rangle$ is $\mathscr{D}^A$. Observe moreover that (A.5) guarantees that for small enough ϵ if we measure $|\tilde{\phi}\rangle$ in the Z basis then with probability $\in \left[\frac{1-3\epsilon}{T'+1}, \frac{1+3\epsilon}{T'+1}\right]$ the clock register is equal to $|0\rangle_{\text{clock}}$ and with probability $\in \left[\frac{1-3\epsilon}{T'+1}, \frac{1+3\epsilon}{T'+1}\right]$ the clock register is equal to $|T'\rangle_{\text{clock}}$. Moreover conditioned on the clock register being $|0\rangle_{\text{clock}}$ probability of out and aux register being $|0\rangle_{\text{out}}, |0^n\rangle_{\text{aux}}$ respectively is, by (A.7), lower bounded by $1 - \frac{\epsilon}{T'}$. Thus we collect a sample from $\mathscr{D}^A$ with probability $\in \left[\frac{1-3\epsilon}{T'+1}(1-\frac{\epsilon}{T'}), \frac{1-3\epsilon}{T'+1}\right] \subseteq \left[\frac{1-5\epsilon}{T'+1}, \frac{1+5\epsilon}{T'+1}\right]$.

For the second condition observe that

$$\begin{aligned}
&|p - \langle 0^n|_{\text{aux}} \langle\psi_{\mathscr{D}^A}|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi^{(1)}_{\text{out}} G |0\rangle_{\text{out}} |\psi_{\mathscr{D}^A}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}| \\
&= |\langle\xi_{T'}|_{\text{comp}} W^\dagger \Pi^{(1)}_{\text{out}} W |\xi_{T'}\rangle_{\text{comp}} - \langle 0^n|_{\text{aux}} \langle\psi_{\mathscr{D}^A}|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi^{(1)}_{\text{out}} G |0\rangle_{\text{out}} |\psi_{\mathscr{D}^A}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}| \\
&\leq |\langle\xi_0|_{\text{comp}} W^\dagger \Pi^{(1)}_{\text{out}} W |\xi_0\rangle_{\text{comp}} - \langle 0^n|_{\text{aux}} \langle\psi_{\mathscr{D}^A}|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi^{(1)}_{\text{out}} G |0\rangle_{\text{out}} |\psi_{\mathscr{D}^A}\rangle_{\text{adv}} |0^n\rangle_{\text{aux}}| + 4\epsilon T' \\
&\leq \frac{\epsilon}{T'} + 4\epsilon T' \leq 5\epsilon T',
\end{aligned}$$

where in the first inequality we used (A.6) and the fact that the largest eigenvalue of $W^\dagger \Pi^{(1)}_{\text{out}} W$ is at most of norm 1 and in the second inequality we used (A.7) and again the fact that the largest eigenvalue of $W^\dagger \Pi^{(1)}_{\text{out}} W$ is at most of norm 1.

Realizing the clock. As we mentioned we also need to specify how to realize the clock register. The naive implementation would result in a $O(\log(T'))$-local Hamiltonian. To obtain a 5-local Hamiltonian we use a unary representation. That is we embed the counter space in a larger space in the following way

$$|j\rangle_{\text{clock}} \mapsto |\underbrace{1,\ldots,1}_{j},\underbrace{0,\ldots,0}_{T'-j}\rangle.$$

We need to now change H_{in} and H_{prop} to be consistent with this change. But more importantly we need to also penalize incorrect configurations in the clock register. This is what the H_{clock} term is responsible for. We refer the readeer to [KSV02] for details. The proof of Lemma 7 extends naturally to this case. □

We will need a slight extension of Lemma 7 to the case where **P** sends mixed states. For the standard use cases of the reduction this extension is trivial but our purposes require more careful treatment. The difference of our setup in comparison to the standard reduction is that we also collect samples that need to satisfy a specific requirement and this is the reason why

the analysis is more involved.

Corollary 1 (Circuit-to-Hamiltonian Reduction for Mixed States)**.** *For every comparison circuit G, if $d_H(\mathscr{D},\mathscr{D}_C)=\eta$ is sufficiently small then there exists an efficiently computable description of a 5-local Hamiltonian H_G with $L=O(n+T')$ many terms such that the following conditions hold. Let $\mathscr{D}^A$ be the distribution of the content of the adv register when measuring ρ_A in the Z basis conditioned on the clock, out and aux registers being all 0 after measurement. For every density matrix ρ_A such that $Tr(H_G\rho_A)\leq\frac{\eta^2}{T'^3}$ if we measure ρ_A in the Z basis then*

- *with probability $\in\left[\frac{1-7\eta}{T'+1},\frac{1+7\eta}{T'+1}\right]$ the clock register is equal to $|0\rangle_{clock}$, the out register is equal to $|0\rangle_{out}$, the aux register is equal to $|0^n\rangle_{aux}$,*
- *with probability $\in\left[\frac{1-7\eta}{T'+1},\frac{1+7\eta}{T'+1}\right]$ the clock register is equal to $\left|T'\right\rangle_{clock}$ and if conditioned on this event the distribution of the out register is a Bernoulli variable with parameter $p\geq 1-3\eta^2$ then $d_H(\mathscr{D}_C,\mathscr{D}^A)\leq O(\eta^{1/4})$.*

Proof. Let $\epsilon=\frac{\eta^2}{T'^2}$. By the ensemble interpretation of density matrices we can express

$$\rho_A=\sum_{i=1}^{k}q_i\left|\phi_i\right\rangle\left\langle\phi_i\right|_{\text{comp}}.$$

Thus we can write

$$\sum_{i=1}^{k}q_i\left\langle\phi_i\right|H_G\left|\phi_i\right\rangle\leq\frac{\epsilon}{T'}.$$

By Markov inequality we have

$$\sum_{i=1}^{k}q_i\mathbb{1}_{\left\{\langle\phi_i|H_G|\phi_i\rangle>\frac{\sqrt{\epsilon}}{T'}\right\}}\leq\frac{\sqrt{\epsilon}}{T'}. \tag{A.8}$$

For $i\in[k]$ let $\mathscr{D}_i^A$ be the distribution of contents of adv conditioned on clock, out, and aux registers being all 0 when measuring $\left|\phi_i\right\rangle$ in the Z basis. For all i such that $\left\langle\phi_i\right|H_G\left|\phi_i\right\rangle\leq\frac{\sqrt{\epsilon}}{T'}$ Lemma 7 guarantees that $\mathscr{D}_i^A$ satisfies the conditions of the reduction.

To see the the first condition, by (A.8), we get that the probability that the clock register is $|0\rangle_{\text{clock}}$ is $\in\left[\frac{1-5\epsilon-2\sqrt{\epsilon}}{T'+1},\frac{1+5\epsilon+2\sqrt{\epsilon}}{T'+1}\right]\subseteq\left[\frac{1-7\sqrt{\epsilon}}{T'+1},\frac{1+7\sqrt{\epsilon}}{T'+1}\right]\subseteq\left[\frac{1-7\eta}{T'+1},\frac{1+7\eta}{T'+1}\right]$. Same bound on probability holds also for the clock register being equal to $\left|T'\right\rangle_{\text{clock}}$.

For $i\in[k]$ let p_i be the probability of obtaining outcome 1 in the out register when measuring $\left|\phi_i\right\rangle$ in the Z basis conditioned on clock register being in state $\left|T'\right\rangle_{\text{clock}}$. Then for the second condition observe that

$$
\begin{aligned}
p &= \sum_{i=1}^{k} q_i p_i \\
&\leq \sum_{i=1}^{k} q_i p_i \mathbb{1}_{\left\{\langle\phi_i|H_G|\phi_i\rangle \leq \frac{\sqrt{\epsilon}}{T'}\right\}} + \frac{\sqrt{\epsilon}}{T'} \\
&\leq \sum_{i=1}^{k} q_i \langle 0^n|_{\text{aux}} \left\langle \psi_{\mathscr{D}_i^A} \right|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi^{(1)}_{\text{out}} G |0\rangle_{\text{out}} \left| \psi_{\mathscr{D}_i^A} \right\rangle_{\text{adv}} |0^n\rangle_{\text{aux}} \mathbb{1}_{\left\{\langle\phi_i|H_G|\phi_i\rangle \leq \frac{\sqrt{\epsilon}}{T'}\right\}} + 6\sqrt{\epsilon}T' \\
&\leq \sum_{i=1}^{k} q_i \langle 0^n|_{\text{aux}} \left\langle \psi_{\mathscr{D}_i^A} \right|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi^{(1)}_{\text{out}} G |0\rangle_{\text{out}} \left| \psi_{\mathscr{D}_i^A} \right\rangle_{\text{adv}} |0^n\rangle_{\text{aux}} + 6\sqrt{\epsilon}T' + \frac{\sqrt{\epsilon}}{T'} \\
&\leq \sum_{i=1}^{k} q_i f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)) + 7\sqrt{\epsilon}T' \qquad \text{(A.9)}
\end{aligned}
$$

where in the first inequality we used (A.8), in the second inequality we used properties of $\mathscr{D}_i^A$ guaranteed by Lemma 7, in the fourth we used Corollary 5.

By (A.9) and the assumption $p \geq 1 - 3\eta^2$ we get that

$$
\sum_{i=1}^{k} q_i f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)) \geq 1 - 3\eta^2 - 7\sqrt{\epsilon}T' \geq 1 - 10\eta^2,
$$

where in the last inequality we used that $\epsilon = \frac{\eta^2}{T'^2}$. We conclude by applying Lemma 12.

□

A.3 Generalized Setting For the Quantum Verifier Protocol

A.3.1 Non i.i.d. Quantum Verifier

Let us now relax the assumption that **P** acts i.i.d., i.e. that **P** sends the same $\left|\psi_{\mathscr{D}^A}\right\rangle$ in every round. We still assume at this point that the states sent by **P** are pure. For a discussion about mixed states see Section A.3.2. First we state a slightly changed theorem.

Theorem 15 (Quantum Verifier)**.** *For every circuit C acting on n qubits, for every $\delta \in (0, \frac{1}{3})$ and all $\eta > 0$ sufficiently small there exists an interactive protocol between a quantum verifier* **V** *and a quantum prover* **P** *with the following properties. The protocol runs in $N = O(\frac{1}{\eta^2}\log(1/\delta))$ rounds, in each round* **P** *sends a pure quantum state on n qubits to* **V**. *At the end of the protocol* **V** *outputs $\perp$ when it rejects the interaction or $x \in \{0,1\}^n$ when it accepts.*

- *(Completeness) There exists $\mathbf{P}^{\mathbf{0}(*)}$ such that for every $\mathscr{D} \in \mathfrak{D}(n)$ satisfying $d_H(\mathscr{D}, \mathscr{D}_C) \leq \eta$ the following holds.. With probability $1 - \delta$ over the randomness in the protocol $\mathbf{P}^{\mathbf{0}(\mathscr{D})}$ succeeds and $x \sim_{i.i.d.} \mathscr{D}$.*

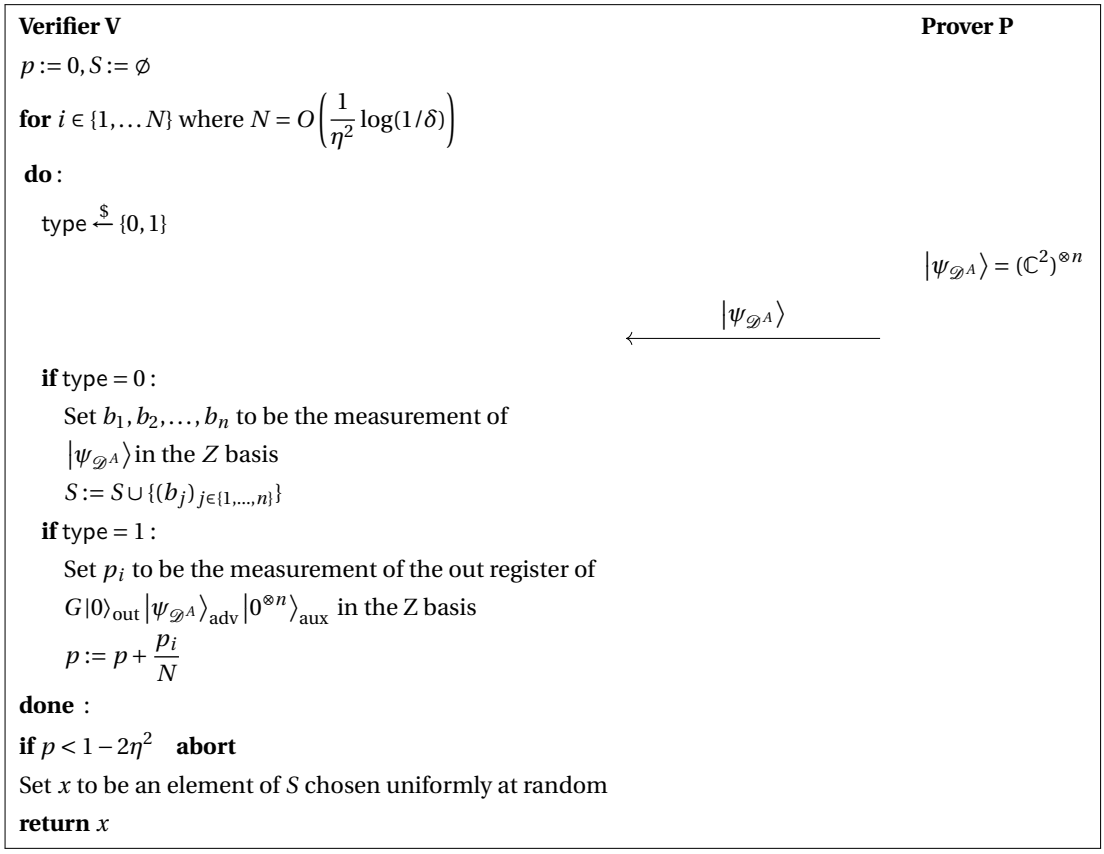


Figure A.1: The interactive protocol for the model where the verifier has access to a quantum computer and the prover doesn't need to act in an i.i.d. fashion.

- *(Soundness) For every* $\mathbf{P}$ *that succeeds with probability* $\geq 1-\frac{\delta}{2}$ *we have that with probability* $1-\delta$ *over the randomness in the protocol* $x \sim_{i.i.d.} \mathscr{D}^A$ *and* $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O\left(\eta^{1/4}\right)$.

Before going to the proof of the theorem we first state a technical lemma.

Lemma 12. *For every* $\eta > 0, k \in \mathbb{N}$*, every set of distributions* $\mathscr{D}, \mathscr{D}_C, \mathscr{D}_1^A, \ldots, \mathscr{D}_k^A \in \mathfrak{D}(n)$ *and every* $q_1, \ldots, q_k \in [0,1]$ *such that* $\sum_{i=1}^k q_1 = 1$ *the following holds. Let* $f(x) = \frac{1}{2}(1+(1-x^2)^2)$*. If* $\sum_{i=1}^k q_i f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)) \geq 1 - 50\eta^2$ *then*

$$d_H\left(\sum_{i=1}^k q_i \mathscr{D}_i^A, \mathscr{D}_C\right) \leq O\left(\eta^{1/4}\right).$$

Proof. We bound the quantity

$$\begin{aligned}
&\sum_{i=1}^k q_i d_H(\mathscr{D}_C, \mathscr{D}_i^A) \\
&\leq \sum_{i=1}^k q_i \left(d_H(\mathscr{D}_C, \mathscr{D}_i^A) \mathbb{1}_{\{d_H(\mathscr{D}_C, \mathscr{D}_i^A) \leq \sqrt{\eta}\}} + \mathbb{1}_{\{d_H(\mathscr{D}_C, \mathscr{D}_i^A) > \sqrt{\eta}\}} \right) \\
&\leq \sqrt{\eta} + \sum_{i=1}^k q_i \mathbb{1}_{\{d_H(\mathscr{D}_C, \mathscr{D}_i^A) > \sqrt{\eta}\}}
\end{aligned} \tag{A.10}$$

Let $l = \sum_{i=1}^k q_i \mathbb{1}_{\{d_H(\mathscr{D}_C, \mathscr{D}_i^A) > \sqrt{\eta}\}}$. By definition and the fact that $f(x) \leq 1 - x^2/2$ we have

$$\left(1 - \frac{\eta}{2}\right) l + (1-l) \geq \sum_{i=1}^k q_i f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)).$$

Using the assumption $\sum_{i=1}^k q_i f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)) \geq 1 - 50\eta^2$ we get $l \leq 100\eta$. Plugging it in (A.10) we get

$$\begin{aligned}
101\sqrt{\eta} &\geq \sqrt{\eta} + 100\eta \\
&\geq \sum_{i=1}^k q_i d_H(\mathscr{D}_C, \mathscr{D}_i^A) \\
&\geq \sqrt{2} \sum_{i=1}^k q_i \triangle(\mathscr{D}_C, \mathscr{D}_i^A) \qquad \text{As } d_H(\mathscr{P}, \mathscr{Q}) \geq \sqrt{2}\triangle(\mathscr{P}, \mathscr{Q}) \\
&\geq \sqrt{2} \triangle\left(\mathscr{D}_C, \sum_{i=1}^k q_i \mathscr{D}_i^A\right) \qquad \text{Triangle inequality and identity } \triangle(\mathscr{P}, \mathscr{Q}) = \frac{1}{2}\|\mathscr{P} - \mathscr{Q}\|_1.
\end{aligned} \tag{A.11}$$

Now we can bound the quantity of interest

$$\begin{aligned}
& d_H\left(\sum_{i=1}^{k} q_i \mathscr{D}_i^A, \mathscr{D}_C\right) \\
& \leq \sqrt{\triangle\left(\sum_{i=1}^{k} q_i \mathscr{D}_i^A, \mathscr{D}_C\right)} && \text{By } d_H(\mathscr{P},\mathscr{Q}) \leq \sqrt{\triangle(\mathscr{P},\mathscr{Q})} \\
& \leq O\left(\eta^{1/4}\right) && \text{By (A.11)}
\end{aligned}$$

□

Proof of Theorem 15. The modified protocol is given in Figure A.1. In each run at most one sample is generated. The number of iterations is changed from $O\left(\frac{K}{\eta^2}\log(1/\delta)\right)$ to $O\left(\frac{1}{\eta^2}\log(1/\delta)\right)$. The biggest change is in the very last step of the protocol, where instead of returning the whole set S we return a random element from S. The reason behind this change will hopefully become clear at the end of the proof.

It suffices to prove the soundness as the completeness proof is analogous to the proof of Theorem 8.

Assume that **P** sends the states $\left|\psi_{\mathscr{D}_1^A}\right\rangle, \left|\psi_{\mathscr{D}_2^A}\right\rangle, \ldots, \left|\psi_{\mathscr{D}_N^A}\right\rangle$ to **V**. Let the rounds in which the type is 1 be $I \subseteq [N]$ and denote $|I|$ by k. Then for every $i \in I$ we have that **V** gets a sample according to a Bernoulli variable with parameter

$$\left\langle 0^n\right|_{\text{aux}} \left\langle \psi_{\mathscr{D}_i^A}\right|_{\text{adv}} \langle 0|_{\text{out}} G^\dagger \Pi_{\text{out}}^{(1)} G |0\rangle_{\text{out}} \left|\psi_{\mathscr{D}_i^A}\right\rangle_{\text{adv}} \left|0^n\right\rangle_{\text{aux}}.$$

Thus by Fact 2 and Corollary 5 we have that with probability $1-\frac{\delta}{2}$

$$\left| p - \frac{1}{k}\sum_{i\in I} f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)) \right| \leq \eta^2, \tag{A.12}$$

where $f(x) = \frac{1}{2}(1+(1-x^2)^2)$.

P succeeds with probability $1-\frac{\delta}{2}$ so by (A.12) and the union bound we get that with probability $1-\delta$

$$\frac{1}{k}\sum_{i\in I} f(d_H(\mathscr{D}_C, \mathscr{D}_i^A)) \geq 1 - 2\eta^2 - \eta^2 \geq 1 - 3\eta^2. \tag{A.13}$$

By Lemma 12 we get then

$$\mathbb{P}_I\left[d_H\left(\frac{1}{k}\sum_{i\in I} \mathscr{D}_i^A, \mathscr{D}_C\right) \geq O(\eta^{1/4}) \right] \leq \delta.$$

As I and $[N] \setminus I$ have the same distribution we also get

$$\mathbb{P}_I \left[d_H \left(\frac{1}{N-k} \sum_{i \notin I} \mathscr{D}_i^A, \mathscr{D}_C \right) \geq O(\eta^{1/4}) \right] \leq \delta.$$

Finally, it can be seen that the samples we collected in S came exactly from the distribution $S \sim \Pi_{i \notin I} \mathscr{D}_i^A$, so if we choose the sample to return x as a uniformly random element of S then $x \sim \frac{1}{N-k} \sum_{i \notin I} \mathscr{D}_i^A$. This concludes the proof as $\mathscr{D}^A = \frac{1}{N-k} \sum_{i \notin I} \mathscr{D}_i^A$.

□

A.3.2 Prover sending mixed states

In this section we explain what happens when instead of sending a pure state $|\psi_{\mathscr{D}^A}\rangle$, **P** is allowed to send a mixed state ρ_A. This means that **P** can prepare a state $|\psi\rangle_{\mathrm{E,F}}$ in a bigger space $(\mathbb{C}^2)_{\mathrm{E}}^{\otimes n} \otimes H_{\mathrm{F}}$ and send only the E part of the system to **V**. We still assume here that **P** acts in an i.i.d. fashion. In this setting the guarantee for soundness will deteriorate (as in Theorem 15) to $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta^{1/4})$ in comparison to $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta)$ as in Theorem 8. The slightly changed theorem becomes

Theorem 16 (Quantum Verifier with Mixed States)**.** *For every circuit C acting on n qubits, for every $\delta \in (0, \frac{1}{3})$, $K \in \mathbb{N}$ and all $\eta > 0$ sufficiently small there exists an interactive protocol between a quantum verifier* **V** *and a quantum prover* **P** *with the following properties. The protocol runs in $N = O(\frac{K}{\eta^2} \log(1/\delta))$ rounds and in each round* **P** *sends a (potentially mixed) quantum state on n qubits to* **V**. *At the end of the protocol* **V** *outputs $\perp$ when it rejects the interaction or it outputs $S = \{x_1, \ldots, x_{|S|}\}$, $x_i \in \{0,1\}^n$, when it accepts.*

- *(Completeness) There exists $\mathbf{P}^{\mathscr{D}(*)}$ such that for every $\mathscr{D} \in \mathfrak{D}(n)$ satisfying $d_H(\mathscr{D}, \mathscr{D}_C) \leq \eta$ the following holds. With probability $1 - \delta$ over the randomness in the protocol $\mathbf{P}^{\mathscr{D}(\mathscr{D})}$ succeeds, $S \sim_{i.i.d.} \mathscr{D}^{|S|}$ and $|S| \geq \Omega(K)$.*
- *(Soundness) For every* **P** *that succeeds with probability at least $\frac{2}{3}$ we have $S \sim_{i.i.d.} (\mathscr{D}^A)^{|S|}$ and $d_H(\mathscr{D}_C, \mathscr{D}^A) \leq O(\eta^{1/4})$.*

Proof of Theorem 16. We only need to verify the soundness property as for the completeness we know that **P** sends pure states. By the ensemble interpretation of density matrices we can express

$$\rho_A = \sum_{j=1}^{k} q_i \left| \psi_{\mathscr{D}_i^A} \right\rangle \left\langle \psi_{\mathscr{D}_i^A} \right|, \tag{A.14}$$

where $\left| \psi_{\mathscr{D}_i^A} \right\rangle \in (\mathbb{C}^2)^{\otimes n}$. This expression is not unique but it will not play a role for us. We observe that measuring ρ_A in the Z basis and collecting a sample is equivalent to collecting a sample from a distribution $\sum_{j=1}^{k} q_i \mathscr{D}_i^A$. By Corollary 5 we know that the probability of

obtaining outcome 1 when running G on $\left|\psi_{\mathscr{D}_i^A}\right\rangle$ and measuring out register is equal to the Bernoulli variable with parameter $f(d_H(\mathscr{D}_C,\mathscr{D}_i^A))$, for $f(x)=\frac{1}{2}(1+(1-x^2)^2)$. The distribution of measuring the out register when running G on ρ_A is thus equal to

$$\sum_{j=1}^{k} q_i \cdot f(d_H(\mathscr{D}_C,\mathscr{D}_i^A)).$$

By Fact 2 and the setting of N we have that with probability $1-\frac{\delta}{2}$

$$\left| p - \sum_{j=1}^{k} q_i \cdot f(d_H(\mathscr{D}_C,\mathscr{D}_i^A)) \right| \leq \eta^2$$

P succeeds with probability $\frac{2}{3}$ so by the union bound and the fact that $\frac{1}{3}+\frac{\delta}{2}<1$ we get that $\sum_{j=1}^{k} q_i \cdot f(d_H(\mathscr{D}_C,\mathscr{D}_i^A)) \geq p-\eta^2 \geq 1-3\eta^2$. Application of Lemma 12 finishes the proof. □

B Appendicies for Chapter 4

B.1 CCA-Secure Bit-Encryption from OWF

In this appendix, we describe a simple quantum public key bit encryption scheme that satisfies the strong notion of CCA security. The construction relies on a quantum-secure pseudorandom function

$$\mathsf{PRF}\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^{3\lambda}$$

which, as mentioned earlier in section 1.4.1, can be constructed from any quantum-secure one-way function. Then our quantum PKE scheme $\Pi = (\mathcal{G}en, \mathcal{QPKG}en, \mathcal{E}nc, \mathcal{D}ec)$ is defined as follows:

- The key generation algorithm $\mathcal{G}en(1^\lambda)$ samples two keys $\mathsf{dk}_0 \xleftarrow{\$} \{0,1\}^\lambda$ and $\mathsf{dk}_1 \xleftarrow{\$} \{0,1\}^\lambda$ and sets $\mathsf{dk} = (\mathsf{dk}_0, \mathsf{dk}_1)$. The public-key generation $\mathcal{QPKG}en(\mathsf{dk})$ prepares the states

  $$\left|qp\mathcal{k}_0\right\rangle = \sum_{x\in\{0,1\}^\lambda} \left|x, f_{\mathsf{dk}_0}(x)\right\rangle \quad \text{and} \quad \left|qp\mathcal{k}_1\right\rangle = \sum_{x\in\{0,1\}^\lambda} \left|x, f_{\mathsf{dk}_1}(x)\right\rangle.$$

  Where $\{f_{\mathsf{dk}}\}_{\mathsf{dk}}$ is a PRF. Moreover, both states are efficiently computable since the PRF can be efficiently evaluated in superposition. The quantum public key is then given by the pure state $\left|qp\mathcal{k}\right\rangle = \left|qp\mathcal{k}_0\right\rangle \otimes \left|qp\mathcal{k}_1\right\rangle$, whereas the classical secret key consists of the pair $\mathsf{dk} = (\mathsf{dk}_0, \mathsf{dk}_1)$.

- Given a message $\mathsf{pt} \in \{0,1\}$, the encryption algorithm $\mathcal{E}nc(\left|qp\mathcal{k}\right\rangle, \mathsf{pt})$ simply measures $\left|qp\mathcal{k}_{\mathsf{pt}}\right\rangle$ in the computational basis, and outputs the measurement outcome as the *classical* ciphertext $qc = (x, y)$ and the post measurement state $\left|x\right\rangle\left|y\right\rangle$.

- Given the ciphertext $qc = (x, y)$, the decryption algorithm $\mathcal{D}ec(\mathsf{dk}, qc)$ first checks whether $f_{\mathsf{dk}_0}(x) = y$ and returns 0 if this is the case. Next, it checks whether $f_{\mathsf{dk}_1}(x) = y$ and returns 1 in this case. Finally, if neither is the case, the decryption algorithm returns $\perp$.

Next, we establish the correctness of this scheme.

Theorem 17. *If PRF is a quantum-secure pseudorandom function, then the quantum PKE scheme* Π *is correct.*

Proof. Observe that the scheme is perfectly correct if the ranges of f_{dk_0} and f_{dk_1} are disjoint. By a standard argument, we can instead analyze the case of two truly random functions f_0 and f_1, and the same will hold for f_{dk_0} and f_{dk_1}, except on a negligible fraction of the inputs. Fix the range of f_0, which is of size at most 2^λ. Then the probability that any given element of f_1 falls into the same set is at most $2^{-2\lambda}$, and the desired statement follows by a union bound. □

Finally, we show that the scheme is CCA-secure. The main tool used in the proof is the one-way to hiding lemma [AHU19].

Lemma 13 (One-way to hiding)**.** *Let* $G, H : X \to Y$ *be random functions and* $S \subset X$ *an arbitrary set with the condition that* $\forall x \notin S, G(x) = H(x)$*, and let* z *be a random bitstring. Further, let* $A^H(z)$ *be a quantum oracle algorithm that queries* H *with depth at most* d*. Define* $\mathscr{B}^H(z)$ *to be an algorithm that picks* $i \in [d]$ *uniformly, runs* $A^H(z)$ *until just before its* i^{th} *round of queries to* H *and measures all query input registers in the computational basis and collects them in a set* T*. Let*

$$P_{left} = \Pr[1 \leftarrow A^H(z)], \quad P_{right} = \Pr[1 \leftarrow A^G(z)],$$
$$P_{guess} = \Pr[S \cap T \neq \emptyset \,|\, T \leftarrow \mathscr{B}^H(z)]$$

Then we have that

$$|P_{left} - P_{right}| \leq 2d\sqrt{P_{guess}} \quad \text{and} \quad |\sqrt{P_{left}} - \sqrt{P_{right}}| \leq 2d\sqrt{P_{guess}} \tag{B.1}$$

Theorem 18. *If* $\{f_{\mathsf{dk}}\}_{\mathsf{dk}}$ *is a quantum-secure pseudorandom function ensemble, then the quantum PKE scheme* Π *is CCA-secure.*

Proof. It suffices to show that the CCA experiment with the bit b fixed to 0 is indistinguishable from the same experiment but with b fixed to 1. To this end we consider a series of hybrids, starting with the former and ending with the latter:

- **Hybrid 0:** This is the original CCA experiment except that the bit b fixed to 0.
- **Hybrid 1:** In this (inefficient) hybrid, we modify hybrid 0 to instead compute $\left|qpk_0\right\rangle$ as

$$\left|qpk_0\right\rangle = \sum_{x \in \{0,1\}^\lambda} \left|x, f(x)\right\rangle,$$

  where f is a truly uniformly random function.

The indistinguishability between these two hybrids follows by a standard reduction against the quantum security of PRF: To simulate the desired n copies of $\left|qpk_0\right\rangle$, and to answer decryption queries (except the one that contains the challenge ciphertext), the reduction simply queries the oracle provided by the PRF security experiment (possibly in superposition). Note that whenever the oracle implements PRF, then the view of the distinguisher is identical to hybrid 0, whereas if the oracle implements a truly random function, then the view of the distinguisher is identical to hybrid 1.

- **Hybrid 2:** In this (inefficient) hybrid, we modify hybrid 1 such that the challenge ciphertext is sampled as
$$x \xleftarrow{\$} \{0,1\}^\lambda \quad \text{and} \quad y \xleftarrow{\$} \{0,1\}^{3\lambda}.$$

The indistinguishability of hybrids 1 and 2 follows from the one-way to hiding lemma (lemma 13). Let H be such that $H(x) = y$ and for all $x' \neq x$ we set $H(x') = f(x')$, and let $S = \{x\}$. Let A be the adversary playing the security experiment. We claim that A^f is the adversary playing in hybrid 1 whereas A^H corresponds to the adversary playing hybrid 2: Observe that the public keys can be simulated with oracle access to f (H, respectively) by simply querying on a uniform superposition of the input domain, whereas the decryption queries can be simulated by query basis states. Importantly, for all queries after the challenge phase, the adversary is not allowed to query x to $\mathcal{D}ec^*$. Hence the set T, collected by $\mathcal{B}$ is a set of at most n uniform elements from the domain of f, along with Q basis states, where Q denotes the number of queries made by the adversary to the decryption oracle *before* the challenge ciphertext is issued. By a union bound
$$P_{\text{guess}} = \Pr[T \cap \{x\} \neq \emptyset] \leq \frac{(n+Q)}{2^\lambda} = \mathsf{negl}(\lambda)$$
since x is uniformly sampled. Applying lemma 13, we deduce that $|P_{\text{left}} - P_{\text{right}}|$ is also negligible, i.e., which bounds the distance between the two hybrids.

- **Hybrid 3:** In this (efficient) hybrid, we modify hybrid 2 to compute $\left|qpk_0\right\rangle$ by using the pseudorandom function f_{dk_0} instead of the truly random function f. That is, we revert the change done in hybrid 1.

Indistinguishability follows from the same argument as above.

- **Hybrid 4:** In this (inefficient) hybrid, we modify hybrid 3 to compute $\left|qpk_1\right\rangle$ as
$$\left|qpk_1\right\rangle = \sum_{x\in\{0,1\}^\lambda} \left|x, f(x)\right\rangle$$
where f is a truly uniformly random function.

Indistinguishability follows from the same argument as above.

- **Hybrid 5:** In this (inefficient) hybrid, we modify hybrid 4 by fixing the bit b to 1 and computing the challenge ciphertext honestly, i.e., as

$$x \xleftarrow{\$} \{0,1\}^{\lambda} \quad \text{and} \quad y = f(x).$$

Indistinguishability follows from the same argument as above.

- **Hybrid 6:** In this (efficient) hybrid, we modify hybrid 5 to compute $\left|qpk_1\right\rangle$ by using the pseudorandom function f_{dk_1} instead of the truly random function f. That is, we revert the change done in hybrid 4.

Indistinguishability follows from the same argument as above. The proof is concluded by observing that the last hybrid is identical to the CCA experiment with the bit b fixed to 1. □

Bibliography

[AAC+22] Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Status report on the third round of the nist post-quantum cryptography standardization process. https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf, 2022.

[Aar18] Scott Aaronson. Shadow tomography of quantum states. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 325–338. ACM Press, June 2018.

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

[ACC+22] Per Austrin, Hao Chung, Kai-Min Chung, Shiuan Fu, Yao-Ting Lin, and Mohammad Mahmoody. On the impossibility of key agreements from quantum random oracles. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 165–194. Springer, Heidelberg, August 2022.

[AES01] Advanced Encryption Standard (AES). National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce, November 2001.

[AGM18] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 489–519. Springer, Heidelberg, April / May 2018.

[AGQY22] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. Pseudorandom (function-like) quantum state generators: New definitions and applications. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 237–265. Springer, Heidelberg, November 2022.

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2019.

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 208–236. Springer, Heidelberg, August 2022.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 430–454, 2015.

[BB84] Charles H. Bennett and Gilles Brassard. An update on quantum cryptography (impromptu talk). In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 475–480. Springer, Heidelberg, August 1984.

[BBDV20] Subhadeep Banik, Khashayar Barooti, F. Betül Durak, and Serge Vaudenay. Cryptanalysis of lowmc instances using single plaintext/ciphertext pair. *IACR Trans. Symmetric Cryptol.*, 2020(4):130–146, 2020.

[BBI19] Subhadeep Banik, Khashayar Barooti, and Takanori Isobe. Cryptanalysis of plantlet. *IACR Trans. Symm. Cryptol.*, 2019(3):103–120, 2019.

[BBSS23] Amit Behera, Zvika Brakerski, Or Sattath, and Omri Shmueli. Pseudorandomness with proof of destruction and applications. Cryptology ePrint Archive, Paper 2023/543, 2023. https://eprint.iacr.org/2023/543.

[BBVY21] Subhadeep Banik, Khashayar Barooti, Serge Vaudenay, and Hailun Yan. New attacks on lowmc instances with a single plaintext/ciphertext pair. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 303–331. Springer, 2021.

[BCC+10] Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir, and Bo-Yin Yang. Fast exhaustive search for polynomial systems in F_2. In *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, pages 203–218, 2010.

[BCC+23] Khashayar Barooti, Daniel Collins, Simone Colombo, Loïs Huguenin-Dumittan, and Serge Vaudenay. On active attack detection in messaging with immediate

decryption. Cryptology ePrint Archive, Paper 2023/880, 2023. https://eprint.iacr.org/2023/880.

[BCG+02a] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *43rd FOCS*, pages 449–458. IEEE Computer Society Press, November 2002.

[BCG+02b] Howard Barnum, Claude Crepeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. Cryptology ePrint Archive, Report 2002/082, 2002. https://eprint.iacr.org/2002/082.

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 467–496, Virtual Event, August 2021. Springer, Heidelberg.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018.

[BCM+21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *J. ACM*, 68(5), August 2021.

[BDF18] Charles Bouillaguet, Claire Delaplace, and Pierre-Alain Fouque. Revisiting and improving algorithms for the 3xor problem. *IACR Trans. Symmetric Cryptol.*, 2018(1):254–276, 2018.

[BdK+21] Carsten Baum, Cyprien de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. Banquet: Short and fast signatures from AES. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 266–297. Springer, Heidelberg, May 2021.

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. In *31st FOCS*, pages 16–25. IEEE Computer Society Press, October 1990.

[BGHD+23] Khashayar Barooti, Alex B. Grilo, Lo is Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-key encryption with quantum keys. Cryptology ePrint Archive, Paper 2023/877, 2023. https://eprint.iacr.org/2023/877.

[BGR23] Khashayar Barooti, Grzegorz Głuch, and Marc-Olivier Renou. How hard is it to fake entanglement? a complexity theoretic view of nonlocality and its applications to delegating quantum computation, 2023.

[BJ95] Nader H. Bshouty and Jeffrey C. Jackson. Learning dnf over the uniform distribution using a quantum example oracle. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, COLT '95, page 118–127, New York, NY, USA, 1995. Association for Computing Machinery.

[BL08] Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.

[BS23] Mohammed Barhoush and Louis Salvail. How to sign quantum messages. *arXiv preprint arXiv:2304.06325*, 2023.

[BZ13a] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013.

[BZ13b] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013.

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1825–1842, 2017.

[CDMW18] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. A black-box construction of non-malleable encryption from semantically secure encryption. *Journal of Cryptology*, 31(1):172–201, January 2018.

[CEV22] Céline Chevalier, Ehsan Ebrahimi, and Quoc Huy Vu. On security notions for encryption in a quantum world. In Takanori Isobe and Santanu Sarkar, editors, *Progress in Cryptology - INDOCRYPT 2022 - 23rd International Conference on Cryptology in India, Kolkata, India, December 11-14, 2022, Proceedings*, volume 13774 of *Lecture Notes in Computer Science*, pages 592–613. Springer, 2022.

[CLLW22] Kai-Min Chung, Yi Lee, Han-Hsuan Lin, and Xiaodi Wu. Constant-round blind classical verification of quantum sampling. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 707–736. Springer, Heidelberg, May / June 2022.

[Col23] Andrea Coladangelo. Quantum trapdoor functions from classical one-way functions. Cryptology ePrint Archive, Paper 2023/282, 2023. https://eprint.iacr.org/2023/282.

[dDOS19] Cyprien de Saint Guilhem, Lauren De Meyer, Emmanuela Orsini, and Nigel P. Smart. BBQ: Using AES in picnic signatures. In Kenneth G. Paterson and Douglas Stebila, editors, *SAC 2019*, volume 11959 of *LNCS*, pages 669–692. Springer, Heidelberg, August 2019.

[DEM15] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Higher-order cryptanalysis of lowmc. In *Information Security and Cryptology - ICISC 2015 - 18th International Conference, Seoul, South Korea, November 25-27, 2015, Revised Selected Papers*, pages 87–101, 2015.

[Din21] Itai Dinur. Cryptanalytic applications of the polynomial method for solving multivariate equation systems over gf(2). Cryptology ePrint Archive, Paper 2021/578, 2021. https://eprint.iacr.org/2021/578.

[DKP+19] Itai Dinur, Daniel Kales, Angela Promitzer, Sebastian Ramacher, and Christian Rechberger. Linear equivalence of block ciphers with partial non-linear layers: Application to lowmc. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 343–372. Springer, 2019.

[DKRS] Christoph Dobraunig, Daniel Kales, Chistian Rechberger, and Markus Schofnegger. Survey of key-recovery attacks onlowmc in a single plaintext/ciphertext scenario. https://raw.githubusercontent.com/lowmcchallenge/lowmcchallenge-material/master/docs/survey.pdf.

[DLMW15] Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang. Optimized interpolation attacks on lowmc. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 535–560. Springer, 2015.

[DN19] Itai Dinur and Niv Nadler. Multi-target attacks on the picnic signature scheme and related protocols. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 699–727. Springer, 2019.

[Dol20] Javad Doliskani. Efficient quantum public-key encryption from learning with errors. Cryptology ePrint Archive, Paper 2020/1557, 2020. https://eprint.iacr.org/2020/1557.

[FHcvM18] Joseph F. Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Phys. Rev. Lett.*, 120:040501, Jan 2018.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

[GBU23] Grzegorz Gluch, Khashayar Barooti, and Rüdiger Urbanke. Breaking a classical barrier for classifying arbitrary test examples in the quantum model. In *International Conference on Artificial Intelligence and Statistics*, pages 11457–11488. PMLR, 2023.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[GKKM20] Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[GLSV21] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in MiniQCrypt. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 531–561. Springer, Heidelberg, October 2021.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. ZKBoo: Faster zero-knowledge for Boolean circuits. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 1069–1083. USENIX Association, August 2016.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 171–185. Springer, Heidelberg, August 1987.

[Got05] Daniel Gottesman. Quantum public key cryptography with information-theoretic security. *https://www2.perimeterinstitute.ca/personal/dgottesman/Public-key.ppt*, 2005.

[HKP20] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *Proceedings of the 39th Annual ACM*

Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007, pages 21–30, 2007.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995.

[IR90] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 8–26. Springer, Heidelberg, August 1990.

[JAC+20] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. SIKE. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 126–152. Springer, Heidelberg, August 2018.

[JNV+22] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*=re, 2022.

[Jou09] Antoine Joux. *Algorithmic Cryptanalysis*. CRC Press, 2009.

[KKNY05] Akinori Kawachi, Takeshi Koshiba, Harumichi Nishimura, and Tomoyuki Yamakami. Computational indistinguishability between quantum states and its cryptographic application. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 268–284. Springer, Heidelberg, May 2005.

[KLVY22] Yael Tauman Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. Cryptology ePrint Archive, Report 2022/400, 2022. https://eprint.iacr.org/2022/400.

[KMNY23] Fuyuki Kitagawa, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum public-key encryption with tamper-resilient public keys from one-way functions. Cryptology ePrint Archive, Paper 2023/490, 2023. https://eprint.iacr.org/2023/490.

[KQST22] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. *arXiv preprint arXiv:2212.00879*, 2022.

[Kre21] William Kretschmer. Quantum pseudorandomness and classical complexity. In Min-Hsiu Hsieh, editor, *16th Conference on the Theory of Quantum Computation,*

Communication and Cryptography, TQC 2021, July 5-8, 2021, Virtual Conference, volume 197 of *LIPIcs*, pages 2:1–2:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[KSV02] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, USA, 2002.

[LDK+20] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[LIM20] Fukang Liu, Takanori Isobe, and Willi Meier. Cryptanalysis of Full LowMC and LowMC-M with Algebraic Techniques. *IACR Cryptol. ePrint Arch.*, 2020:1034, 2020.

[LIM21] Fukang Liu, Takanori Isobe, and Willi Meier. A simple algebraic attack on 3-round lowmc. *IACR Cryptol. ePrint Arch.*, 2021:255, 2021.

[LS19] Gaëtan Leurent and Ferdinand Sibleyras. Low-memory attacks against two-round even-mansour using the 3-xor problem. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, pages 210–235, 2019.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267, 2018.

[MCEM97] Cleve Ekert Macchiavello, B Y R. Cleve, A. Ekert, and C. Macchiavello. Quantum algorithms revisited. In *Proceedings of the Royal Society of London A*, pages 339–354, 1997.

[Ms09] Steven Myers and abhi shelat. Bit encryption is complete. In *50th FOCS*, pages 607–616. IEEE Computer Society Press, October 2009.

[MW23] Giulio Malavolta and Michael Walter. Non-interactive quantum key distribution. Cryptology ePrint Archive, Paper 2023/500, 2023. https://eprint.iacr.org/2023/500.

[MY22a] Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. Cryptology ePrint Archive, Paper 2022/1336, 2022. https://eprint.iacr.org/2022/1336.

[MY22b] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2022.

[Nan15] Mridul Nandi. Revisiting security claims of XLS and COPA. *IACR Cryptol. ePrint Arch.*, 2015:444, 2015.

[NC16] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016.

[NI09] Georgios M. Nikolopoulos and Lawrence M. Ioannou. Deterministic quantum-public-key encryption: Forward search attack and randomization. *Phys. Rev. A*, 79:042327, Apr 2009.

[NS15] Ivica Nikolic and Yu Sasaki. Refinements of the k-tree algorithm for the generalized birthday problem. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 683–703, 2015.

[NYC15] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436. IEEE Computer Society, 2015.

[OTU00] Tatsuaki Okamoto, Keisuke Tanaka, and Shigenori Uchiyama. Quantum public-key cryptosystems. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 147–165. Springer, Heidelberg, August 2000.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. https://eprint.iacr.org/2006/145.

[RST18] Christian Rechberger, Hadi Soleimany, and Tyge Tiessen. Cryptanalysis of low-data instances of full lowmcv2. *IACR Trans. Symmetric Cryptol.*, 2018(3):163–181, 2018.

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

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

[SZS+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016.

[Vid22] Thomas Vidick. Interactive proofs with quantum devices. Lecture Notes from Fondation Sciences Mathématiques de Paris, 2022. http://users.cms.caltech.edu/~vidick/teaching/fsmp/fsmp.pdf.

[Wag02] David A. Wagner. A generalized birthday problem. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–303. Springer, 2002.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, jan 1983.

[YTL+19] Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 952–966. IEEE, 2019.

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd FOCS*, pages 69–74. IEEE Computer Society Press, October / November 2022.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012.

Khashayar Barooti

Lausanne, Switzerland
+41-782570066 khashayar.barooti@epfl.ch Personal Webpage

EDUCATION

University of Tehran **2014 – 2018**
Bachelor's Degree in Applied Mathematics and Computer Science - ***GPA: 18.69*** */20* *Tehran, Iran*

Ecole polytechnique fédérale de Lausanne (EPFL) **2018 – Present**
PhD in computer science - Laboratory of Cryptography and Security (LASEC) *Lausanne, Switzerland*
Under Supervision of *Prof. Serge Vaudenay*
Expected date of graduation: October 2023

PUBLICATIONS

- On Active Attack Detection in Messaging with Immediate Decryption
  Khashayar Barooti*, Daniel P. Collins, Simone Colombo, Loïs Huguenin, Serge Vaudenay*
  IACR-CRYPTO 2023 (to appear)

- Breaking a Classical Barrier for Classifying Arbitrary Test Examples in the Quantum Model
  Khashayar Barooti*, Grzegorz Gluch, Rüdiger Urbanke*
  AISTATS 2023

- New Attacks on LowMC Instances with a Single Plaintext/Ciphertext Pair, eprint
  Subhadeep Banik, ***Khashayar Barooti****, Serge Vaudenay, Hailun Yan*
  IACR-ASIACRYPT 2021

- Cryptanalysis of LowMC instances using singleplaintext/ciphertext pair, ToSC
  Subhadeep Banik, ***Khashayar Barooti****, Betül Durak, Serge Vaudenay*
  IACR-FSE 2022
  Best Paper Award

- Cryptanalysis of Plantlet, eprint
  Subhadeep Banik, ***Khashayar Barooti****, Takanori Isobe*
  IACR-FSE 2020

PREPRINTS/MANUSCRIPTS IN SUBMISSION

- Public-Key Encryption with Quantum Keys, eprint
  Khashayar Barooti*, Alex B. Grilo, Loïs Huguenin, Giulio Malavolta, Or Sattath, Quoc-Huy Vu and Michael Walter*
  Manuscript in submission

- How hard is it to fake entanglement? A complexity theoretic view of nonlocality and its applications to delegating quantum computation, arxiv
  Khashayar Barooti*, Grzegorz Gluch, Marc-Olivier Renou*
  Manuscript in submission

ONGOING WORK

- Under Computational Assumptions All Entangled States are Non-Local
  Joint work with *Alexandru Gheorghiu, Grzegorz Gluch, Marc-Olivier Renou*

RESEARCH VISITS/INTERNSHIPS

Max-Planck Institute for Security and Privacy **Sep 2022 – Dec 2022**
Research visit hosted by Giulio Malavolta *Bochum, Germany*

TEACHING

Quantum Interactive Protocols **Fall 2021**
Lecturer, jointly with Grzegorz Gluch *EPFL*

- About the lecture series: In this lecture series we cover 2 breakthrough results in quantum complexity theory, namely classical delegation of quantum computation and $\mathsf{MIP}^* = \mathsf{RE}$
- Links: webpage, tube

Analysis of Boolean Functions **Summer 2021**
Lecturer *EPFL*

- About the lecture series: In this lecture series we studied boolean functions from a Fourier Analysis point of view. We covered concepts such as social choice, learning theory, decision trees, query complexity, etc.

Cryptography and Security **Fall 2019-2020-2021**
Teaching Assistant, Graduate course given by Serge Vaudenay *EPFL*

Security Protocols and Applications **Spring 2020**
Teaching Assistant, Graduate course given by Serge Vaudenay *EPFL*

Software Security **Spring 2019**
Teaching Assistant, Graduate course given by Mathias Payer *EPFL*

SKILLS

Mathematics and Theoretical Computer Science: Quantum Information, Cryptography, Computational Complexity, Probabilistic Proofs, Provable Security, Fourier Analysis, Algebraic Geometry, Algebraic Number Theory

Programming Languages: C++ (& NTL), Python (& SAGE, NumPy, Tensorflow), Java, LaTeX

Technologies/Frameworks: Linux, Git

Spoken Languages: Farsi (Native), Turkish (Native), English (Fluent), French (Biginner)

HONOURS and AWARDS

- Best paper award at IACR-FSE 2022
- 3 round winner of LowMC cryptoanalysis challenge
- Bronze Medal in Iranian Mathematics Society's (IMS) National Mathematics Olympiads
- Honorary member of Iranian Mathematics Society
- Member of Iran National Elites Foundation

RESEARCH INTERESTS

I am interested in a broad range of problems related to theoretical cryptography, quantum information and complexity theory and always look forward to learning new topics and techniques. Here is a summary of topics and areas I am actively working on at the moment:

- (Post-)Quantum Cryptography: Cryptography from minimal assumptions, quantum-secure reductions, etc.
- Quantum Information and Foundations: Complexity theoretic view of concepts in quantum information foundations, such as non-locality, steering, etc, and their implications on (quantum) cryptography
- Computational Complexity: Low-depth cryptography