CRYPTOGRAPHIC APPLICATIONS OF THE QUANTUM NO-CLONING PRINCIPLE FOR MESSAGES, PROGRAMS, AND ADVICE

Sébastien Lord

Thesis submitted to the Faculty of Science
in partial fulfillment of the requirements for the degree of
Doctorate in Philosophy Mathematics and Statistics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The Ph.D. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

The no-cloning principle tells us that it is impossible to perfectly replicate the state of an arbitrary quantum system. From its early beginnings with Wiesner, Bennett, and Brassard, quantum cryptography has often relied, either implicitly or explicitly, on this principle to achieve security properties beyond the reach of classical cryptography. We examine in this thesis various such cryptographic applications of quantum uncloneability from two broad perspectives: uncloneability for messages and uncloneability for functionalities.

Starting with uncloneability for messages, we study relations between tamper-evident encryption schemes — symmetric-key schemes for classical messages producing quantum encodings which allows the honest recipient to detect meaningful eavesdropping — and various other security notions. We show that tamper evidence implies encryption, that it can be used to construct revocation schemes (and vice-versa), and we formalize Gottesman's construction of quantum money from schemes satisfying this notion.

Moving on to uncloneability for functionalities, we begin by detailing our contributions to the study of quantum copy-protection and secure software leasing. These are two similar but distinct notions of uncloneability for *families* of functionalities. We show how to instantiate a scheme for point functions which is secure against honest-malicious adversaries in the quantum copy-protection setting and show that this is, generically, a sufficient condition to construct a secure software leasing scheme for the same functions.

We then complete this thesis by reporting work in which we initiate the formal study of *uncloneable quantum complexity theory*. We accomplish this by defining a novel complexity class denoted **neglQP**/upoly — the class of promise problems and languages which can be solved with bounded error by efficiently describable quantum circuits with the help of a quantum advice state exhibiting a type of functional uncloneability — and we show, unconditionally, that this class is non-empty. This relates to uncloneable functionalities as we also formalize how uncloneable advice can be seen as a form of quantum copy-protection for *specific* functions.

Résumé

Le principe d'impossibilité du clonage quantique nous indique qu'il est infaisable de reproduire parfaitement l'état d'un système quantique arbitraire. Depuis ses débuts avec Wiesner, Bennett et Brassard, la cryptographie quantique s'est souvent appuyée sur ce principe pour obtenir des propriétés de sécurité qui sont hors de portée avec des méthodes purement classiques. Cette thèse étudie plusieurs applications cryptographiques de ce principe selon deux perspectives: l'impossibilité de cloner des messages et celle de cloner des fonctionnalités.

En ce qui concerne les messages, nous étudions des relations entre le chiffrement avec témoin d'intégrité — des algorithmes de chiffrement à clé symétrique pour messages classiques produisant des états quantiques permettant la détection d'écoute clandestine — et divers autres notions cryptographiques. Nous démontrons que le témoignage d'intégrité implique le chiffrement et qu'il est possible de construire un protocole de révocation à partir de tels protocoles (et vice-versa). Enfin, nous formalisons la construction de Gottesman de monnaie quantique à partir de tels protocoles.

Concernant les fonctionnalités, nous détaillons nos contributions à l'étude de la protection anti-copie quantique et à la location sécuritaire de logiciels. Ce sont deux notions similaires mais distinctes d'inclonabilité pour des *familles* de fonctionnalités. Nous décrivons l'instanciation d'un protocole de protection anti-copie quantique pour fonctions à point unique qui est sécure contre des adversaires honnêtes-malicieux et démontrons que ceci est, génériquement, une condition suffisante pour construire un protocole de location sécuritaire de logiciels pour les mêmes fonctions.

Nous complétons cette thèse en présentant nos travaux qui ont initié l'étude formelle de la *théorie de la complexité quantique inclonable*. Nous avons accompli ceci en définissant une nouvelle classe de complexité dénotée **neglQP**/upoly — la classe de problèmes pouvant être résolus d'une façon efficace par un algorithme quantique avec l'aide d'un conseil quantique possédant un type d'inclonabilité fonctionnel — et nous démontrons de façon inconditionnelle que cette classe n'est pas vide. Ceci est relié à la notion de fonctionnalités inclonables car nous montrons qu'un conseil inclonable est une forme d'anti-copie quantique pour une fonction *spécifique*.

Acknowledgements

First and foremost, I wish to thank my supervisor Dr. Anne Broadbent for her boundless support, patience, and insights. Whenever I ruled out too quickly a lead or idea she proposed, I always found myself reexamining it soon after with fruitful outcomes.[2]

I also thank my other collaborators with whom I had the privilege to work with during this program: Stacey Jeffery, Martti Karvonen, Supartha Podder, and Aarthi Sundaram. The works presented in this thesis would not have been possible without their contributions.

I must give thanks to the staff at the University of Ottawa's Department of Mathematics and Statistics who were always happy to help with any administrative task or issue which came my way.

Finally, I want to thank my mother Diane, my father Bernard, and my partner Mireille for their unfailing love and encouragements throughout the years.

[2]To any future student of Dr. Broadbent who may be reading this, I humbly suggest you take note.

Contents

Preface

This is a thesis by articles. Essentially all herein has already been published as articles in peer reviewed venues or as manuscripts in a preprint repository. We list these publication here in chronological order. Two of these were produced in collaboration with coauthors. While the contributions of all were integral, we highlight some of the author's specific contributions and non-contributions for these works.

1. **Secure Software Leasing Without Assumptions**

   Produced in collaboration with Anne Broadbent, Stacey Jeffery, Supartha Podder, and Aarthi Sundaram, this work was published in the proceedings of the 19th Theory of Cryptography Conference [BJL+21a]. It is essentially the content of Chapter 3 here.

   Notes on contributions:

   - The author of this thesis initially suggested the construction of a secure software scheme for point functions from any total quantum authentication scheme.
   - The "$x_1 \neq p$" case of the proof of Theorem 3.23 was largely the work of a collaborator.
   - The technical details of Section 3.6.3 was largely the work of a collaborator.

2. **Uncloneable Quantum Advice**

   Produced in collaboration with Anne Broadbent and Martti Karvonen, this work was published in the journal *IACR Communications in Cryptology* [BKL24]. It is essentially the content of Chapter 4 here.

   Notes on contributions:

   - The author of this thesis first sketched the notion of a computably ingenerable sequence of bit strings and how they could be used to derandomize certain cryptographic games, thus yielding uncloneable advice.
   - A collaborator was the first to sketch the existence of sequences of bit strings which would be ingenerable for efficient quantum circuits but computable in longer time by classical Turing machines via simulations.

3. **Relating Quantum Tamper-Evident Encryption to Other Cryptographic Notions**

   The author is the sole contributor to this work which is currently available on the arXiv [Lor24]. It is essentially the content of Chapter 2 here.

   While this is a single-author work, many fruitful discussions on this topic were had with Anne Broadbent. In particular, the results connecting the notions of tamper-evident encryption and encryption with certified deletion were inspired by one of her remarks noting that a party demonstrating that they have deleted a certain piece of information is, implicitly, demonstrating that they possessed this piece of information in the first place.

A remark on orthography. There has been an ongoing implicit debate in the literature on the proper spelling of the word *uncloneable*. Many authors, perhaps even a majority, prefer *unclonable*, i.e. they omit the *e* from *clone* before appending the *-able* suffix. While the *Oxford English Dictionary* recognizes both *clonable* and *cloneable* (notably, neither appears in this reference with the *un-* prefix), it expresses a preference for *clonable* [OED22]. However, to the best of our knowledge, the earliest work in the field of quantum cryptography which uses the word *uncloneable* as an element of technical terminology [Got03] keeps the *e*. We will continue to follow this convention.

Note that this debate is not limited to the field of quantum cryptography. See, for example, Maes' discussion on this subject in their textbook on physically uncloneable functions [Mae13, sec. 2.3.1].

Chapter 1

Introduction

> Information is inevitably tied to a physical representation and therefore to restrictions and possibilities related to the laws of physics and the parts available in the universe.
>
> Rolf Landauer, *The physical nature of information*, 1996

Information is physical [Lan96], and so a complete theory of information processing and computation must necessarily be — or at least incorporate part of — a physical theory. It follows that a complete theory of cryptography, the field which studies computation and information processing *in the presence of malicious parties* [KL15], must also account for our physical reality. This is not simply an epistemological stand for what we think is the "right" or "proper" way to frame the theory of cryptography. It is a position with serious real-world implications. If an adversary is able to leverage quirks of physics to mount more powerful attacks against our cryptographic schemes, should we not account for this in our design processes? And if we are able to utilize aspects of nature to help us better protect our information, would it not be negligent to leave such opportunities unexplored?

Consider now the theory of quantum mechanics. Quantum systems must obey the laws of quantum mechanics, as unintuitive these may initially seem to us typically classical[1] beings. Such systems may, for example, exist in *superpositions* of states, may be *entangled* with one another, and may exhibit *non-locality* when observed. The study of quantum information processing and quantum computing is essentially to ask if quantum mechanical phenomena such as these, which have no direct classical analogues, can be exploited to obtain an information processing advantage. Largely, the answers here have been positive: such advantages can indeed be found! See, for example, the textbook of Nielsen and Chuang for an introduction to this topic [NC10].

While in principle *all* physical systems are quantum systems, quantum phenomenons diverging from the predictions of classical physics like the ones previously mentioned are

[1] Here and throughout this work we use the word "classical" as roughly synonymous with "non-quantum".

typically only observable at microscopic scales such as single photons or atoms. It is a non-trivial task to exercise control on small systems like these to the level of precision required to execute quantum computations. So while quantum computing has been a fruitful area of research, it is important to acknowledge that there remains important engineering challenges to move from theory to practice. However, progress continues to be made and the results of more and more complex quantum computations continue to be announced (e.g. [vDLL+24]).

1.1 Cryptography and Quantum Information Processing

The impact of quantum computing and quantum information processing on cryptography can be broadly split into three almost non-overlapping categories:

1. The application of quantum computation to break classical cryptography.
2. The development of "standard" cryptographic tools, such as encryption and authentication schemes, for quantum states.
3. The leveraging of quantum mechanical phenomena to achieve cryptographic goals otherwise impossible with only classical means.

We reserve the term "quantum cryptography" to refer to the last two of these points, while the first may be more aptly called "quantum cryptanalysis" as the cryptosystems themselves are not quantum in nature.

On the first of these fronts, we can credit Shor with providing the first quantum algorithm with a clear and extremely significant cryptographic application relative to existing classical schemes when he published an efficient *quantum* factoring algorithm [Sho94]. Shor's work had direct implications for the effective security of many widespread cryptographic schemes, such as the RSA cryptosystem [RSA78], should it be implemented on an actual quantum machine. Grover's more widely applicable, but less impactful, generic algorithm for unstructured search [Gro96] also merits mention here. These, paired with technical advancements in efforts to build quantum computers, has spurred an effort to move to "post-quantum cryptography"[2] (e.g. [CJL+16]). However, our work does not touch upon this research direction.

Instead, it is the *third* point which is the main topic of this thesis. The two canonical examples of quantum cryptographic schemes of this flavour are Wiesner's quantum money scheme [Wie83] and Bennett and Brassard's quantum key distribution scheme [BB84]. With quantum money, a bank is able to mint unforgeable banknotes by incorporating into them

[2]We contend that "quantum-resistant cryptography" is a better term, but the community has largely settled on "post-quantum".

quantum systems whose precise states are kept secret. With quantum key distribution, two parties are able to establish a shared secret key over an untrusted quantum communication channel with an additional public-but-authenticated classical communication channel. Both of these schemes achieve goals impossible with only classical information and, in both cases, a high-level argument of security can be obtained by appealing to the *no-cloning principle* and the *information-disturbance principle* of quantum information processing. We review these principles shortly in Section 1.2.

We only touch upon the second topic — "standard" cryptographic tools for quantum states — incidently and inasmuch as it relates to our main topic. In particular, schemes to authenticate the state of quantum systems [BCG+02, GYZ17] will play a significant role in many of our results. Nonetheless, understanding such "standard" cryptographic tools for quantum information, including public-key encryption schemes (e.g. [ABF+16]), may become quite important with the potential rise of a quantum internet (e.g. [CBD+24]) or other widespread quantum communication networks presenting large attack surfaces.

1.2 The No-Cloning and Information-Disturbance Principles

The cornerstone of this thesis is the following theorem, which we will formalize in Section 1.4.

Theorem (informal). In general, it is impossible to clone an unknown quantum state.

We refer to the above as *the no-cloning principle*. This is not only due to the lack of technical specificity of this statement, but also due to the fact that there are various distinct, valid, and interesting ways to formalize this statement, giving rise to various specific no-cloning theorems.

For a review of the history of the no-cloning principle, we refer the interested reader to the work of Ortigoso [Ort18] and the references therein. In summary, Dieks, Wooters, and Zurek [Die82, WZ82] are commonly credited with giving the first formal statements and proofs of a no-cloning theorem in 1982. However, an essentially identical statement and proof had already been presented twelve years prior in a publication by Parks [Par70].

The no-cloning principle is also closely related to the *information-disturbance principle*.

Theorem (informal). In general, it is impossible to obtain information on an unknown quantum state without disturbing this state.

Fuchs has identified this as "the engine which powers quantum cryptography" [Fuc96], a statement that was particularly true at a time where this field essentially reduced to quantum key distribution and quantum money but which still holds weight today.

At a high-level, these two principles imply one another. If we were able to make perfect copies of a quantum state, then we could extract the desired information from the copy without touching or disturbing the original state. In the other direction, if we were able to extract information from a state without disturbing it, we could eventually learn enough about it to reconstruct a second example of it.

In a cryptographic setting, we can appeal to these principles to argue that adversaries are constrained in interesting ways when attacking quantum cryptographic protocols. Specifically, the no-cloning principle prohibits adversaries from endlessly duplicating a quantum resource they have obtained from an honest parties, while the information-disturbance principle can be leveraged by the honest parties to detect the actions of malicious parties.

So, equipped with these ideas, we will in this thesis construct and study quantum cryptographic schemes which encode *classical information* into *quantum states* to achieve interesting security notions. We review these in Section 1.3.

Finally, we note that our formal security proofs do not directly appeal to these principles. This is either because they are formulated without the necessary rigour, or because they are not directly applicable when properly formalized. Instead, they are used as tools for our intuition and otherwise lurk in the background.

1.3 Organization and Overview of Contributions

The remainder of this thesis is organized as three chapters, each a self-contained work.

In Chapter 2, we examine the notion of *tamper-evident encryption*, in Chapter 3 we construct a *secure software leasing scheme*, and, finally, in Chapter 4 we study *uncloneable advice* in the context of quantum complexity theory. All three of these notions can be seen as cryptographic properties which can only be achieved by leveraging the no-cloning and information-disturbance principles offered by quantum mechanics.

We briefly review here the main contributions we make in each of these chapters. We also summarize these in Table 1.1 at the end of this chapter on Page 13.

A note on ordering. These three chapters can, in principle, be read in any order as there is no strict dependence between them and each is complete in its own right. However, the ordering we have picked is not without reason. In particular, the preliminaries of Chapter 2 are much more extensive and complete than those of the other two chapters and it may be beneficial to read Chapter 3 before Chapter 4 due to the conceptual connection between them. Finally, the topic of Chapter 2 is quite distinct from those of Chapters 3 and 4.

1.3.1 Chapter 2: Tamper-Evident Encryption

Context. In 2003, Gottesman introduced the notion of tamper-evident encryption [Got03]. This is a quantum cryptographic primitive which can be thought of as a direct application of the information-disturbance principle to the notion of an encryption scheme with the aim of detecting eavesdropping attempts.

In more details, a symmetric-key encryption scheme for classical messages is said to be *tamper-evident* if it is infeasible for an eavesdropping adversary to collect meaningful information on a transmitted ciphertext *without being detected by the honest recipient*. We say that an eavesdropping attempt has obtained "meaningful information" if an adversary who later learns the secret key can deduce information on the underlying plaintext. An eavesdropping adversary who remains ignorant of the plaintext even after learning the secret key has clearly gained no meaningful information.

Evidently, it is impossible to achieve this security notion if the ciphertext is a classical piece of information as an eavesdropper could make a copy undetected. If they ever learn the secret key, they can simply use the honest decryption procedure to recover the plaintext. However, an appeal to the information-disturbance principle allows us to argue that it may be possible to encode our classical plaintext into a *quantum* ciphertext such that any eavesdropper extracting information from it would disturb it in a manner detectable by the honest recipient.

Gottesman formalized this security notion, established that any quantum authentication scheme [BCG+02] was also necessarily a tamper-evident scheme, and constructed an additional tamper-evident scheme based on the Shor-Presskill proof of security of Bennett and Brassard's quantum key distribution scheme [SP00, BB84].

Contribution. We contribute to our understanding of tamper-evident encryption by clarifying the relationship between it and various other cryptographic notions in an information-theoretic setting.

Our first result is to show that tamper evidence implies the encryption of classical messages. Gottesman was careful to distinguish between two distinct security properties: that of *encryption*, i.e. a ciphertext is meaningless without the secret key, and that of *tamper evidence*, i.e. the honest recipient can detect meaningful eavesdropping. Gottesman explicitly defined a tamper-evident encryption scheme as one possessing both these properties, but did wonder if tamper evidence implied encryption, hence making the encryption requirement superfluous. We give a positive answer to this question: tamper-evidence does indeed imply encryption.

As an analogy, we can imagine a real-world communication mechanism which does ex-

hibit tamper evidence but not encryption: a letter in plain English delivered via an envelope closed by a wax seal. Anyone willing to break the seal would be able to easily read the content of the letter, but such tampering would be evident to anyone subsequently receiving the letter. Can a similar system be cryptographically constructed? Our result demonstrates that no, such simple "sealed envelopes" cannot be instantiated in the information-theoretic setting. Any scheme allowing the detecting of eavesdropping *must also* encrypt the message.

Our second result is to formalize a construction for a quantum money scheme from any tamper-evident scheme which was later proposed by Gottesman [Got11]. Here, our main contribution is to properly quantify the level of security achieved by this construction.

Third, we relate the notion of tamper-evident encryption with that of encryption with revocation [Unr15] or with certified deletion [BI20]. An encryption scheme with revocation, or simply a revocation scheme, is one in which the recipient of a quantum ciphertext can essentially return it to the original sender with a guarantee that no information on the message was learned, provided that they never knew the secret key. An encryption scheme with certified deletion, or simply a certified-deletion scheme, is then a revocation scheme where this return can be completed with only one round of classical communication. We formalize the relation between these two notions and tamper evidence by noting that all three seek to allow an honest party to detect "unauthorized" attempts to extract information from a ciphertext. More precisely, we show how to construct a revocation scheme from any tamper-evident scheme and vice-versa. From these constructions and the fact that we have also shown that tamper evidence implies encryption, we obtain as a direct corollary that revocation also implies encryption.

Finally, we also demonstrate a handful of *separations* between tamper evidence and other cryptographic notions. In particular, we show that a tamper-evident scheme need not be an uncloneable encryption scheme *à la* Broadbent and Lord [BL20], nor does it need to be an authentication scheme. Specifically, we show that it is possible to construct schemes where an adversary *cannot extract* information on the ciphertext without detection, but *can modify* it without detection so that the wrong plaintext is later recovered.

1.3.2 Chapter 3: Secure Software Leasing

Context. In 2009, Aaronson conjectured that it would be possible to "copy-protect" certain functionalities by encoding them as quantum states and leveraging the no-cloning principle [Aar09]. More precisely, a quantum copy-protection scheme would allow a "vendor" to encode a function f sampled from some family $\mathcal{F}$ as a quantum state ρ_f. The scheme is correct if a user who is *a priori* ignorant of f can use a prescribed honest evaluation procedure on input of (ρ_f, x) to obtain $f(x)$. The scheme is secure if it is infeasible to

create two states allowing the evaluation of f from a single instance of ρ_f, even by use of a malicious evaluation procedure deviating from the one prescribed by the scheme.

Aaronson's main contribution — beyond identifying the concept of quantum copy protection — was to show the existence of a specific quantum oracle with respect to which such schemes could be constructed for a wide class of function families $\mathcal{F}$. It remained an open question to construct a quantum copy-protection scheme without the use of an oracle.

More than a decade later, Ananth and La Placa formulated the closely related notion of *secure software leasing* (SSL) [AL21]. Here, once again, a function $f \in \mathcal{F}$ is encoded as a quantum state ρ_f which allows the holder of this state to evaluate f on arbitrary inputs. However, the security guarantee is changed: instead of requiring it to be infeasible to share a program state ρ_f, a secure software leasing scheme allows the user to return the program state to the vendor in a verifiable manner. If the vendor accepts the return, they can be assured that the user has forfeited the ability to evaluate the function f.

Ananth and La Placa instantiated an SSL scheme for a wide class of function families $\mathcal{F}$ under fairly heavy cryptographic assumptions pertaining to obfuscators and set-up assumptions such as a common reference string.

Contributions. Our main contribution to the study of secure software leasing is to construct the first such scheme which is not only unconditionally secure, in the sense that it does not rely on any unproven assumptions, but that is also secure against computationally unbounded adversaries. In other words, we achieve an information-theoretically secure software leasing scheme. As a drawback, our construction is limited to point functions, or their generalization as compute-and-compare functions via the techniques of Coladangelo, Majenz, and Poremba [CMP24].[3] Our contribution is done in three steps.

We begin by defining a notion of "honest-malicious" security for quantum copy-protection schemes. A copy-protection scheme comes equipped with a prescribed, or "honest", evaluation procedure which sets out how a user is expected to compute $f(x)$ from an input x and a program state ρ_f for the function f. However, adversaries are not *a priori* forced to use this evaluation procedure when trying to compute f after an attempt to split the program state ρ_f. In general, the security game considered for quantum copy-protection ends with two adversaries, each having received part of a quantum state derived from a single instance of ρ_f, using arbitrary methods to compute f on the input on which they were challenged. In our taxonomy, we call this "malicious-malicious" copy-protection. We change this security game to require one specific adversary to use the evaluation procedure

[3] A compute-and-compare function is characterized by an underlying function f and an element y in the codomain of f. On some input x, the resulting compute-and-compare function outputs 1 if $f(x) = y$ and outputs 0 otherwise. A point function is a compute-and-compare function where f is the identity.

prescribed by the scheme and call this weaker notion "honest-malicious" security.[4]

Next, we formalize how to generically construct a secure software leasing scheme from any honest-malicious secure quantum copy-protection scheme.

The idea is relatively straightforward. If it is infeasible for an honest evaluation and a malicious evaluation to both correctly compute f, as is guaranteed by an honest-malicious secure quantum copy-protection scheme, then it is highly unlikely for a malicious evaluation to succeed conditioned on an honest evaluation having already succeeded. Hence, in the SSL setting, a software vendor can verify the return of a program state simply by attempting to evaluate it on a random input and verifying that the correct output is obtained. If the program state was generated by an honest-malicious secure quantum copy-protection scheme, then the vendor can be assured that the user has indeed forfeited the ability to compute f.

Our final contribution is to instantiate the theoretical framework that we have set up to obtain an SSL scheme by constructing a honest-malicious secure quantum copy-protection scheme for point functions.

Our construction builds on an existing quantum cryptographic primitive: total quantum authentication schemes [GYZ17]. A total quantum authentication is a symmetric-key authentication scheme for quantum states which can not only detect any modification to an authenticated quantum state which high probability, but can also detect with high probability any non-trivial interaction an adversary may have had with the authenticated state. This includes attempts to obtain information on the state without changing it. We leave the details of our construction to Chapter 3.

We note, however, that information-theoretically secure total quantum authentication schemes do exist unconditionally [GYZ17]. Moreover, we observe that this level of security is maintained throughout our constructions, which is what enables us to achieve an information-theoretically secure SSL scheme.

Finally, we highlight that follow-up work has explored the feasibility of implementing our scheme for point functions on actual quantum devices [WAB21]. Ultimately, the authors concluded that error mitigation remains a challenge.

1.3.3 Chapter 4: Uncloneable Quantum Advice

Context. A central goal of complexity theory is to classify computational problems with respect to the computational resources that are needed to solve them. Here, a problem is understood as a map or a partial map $P : \{0,1\}^* \to \{0,1\}$ partitioning its domain into "yes"

[4] Ananth and La Placa also introduced a notion of *infinite-term* secure software leasing corresponding to "honest-honest" quantum copy-protection where both adversaries must use the honest evaluation procedure.

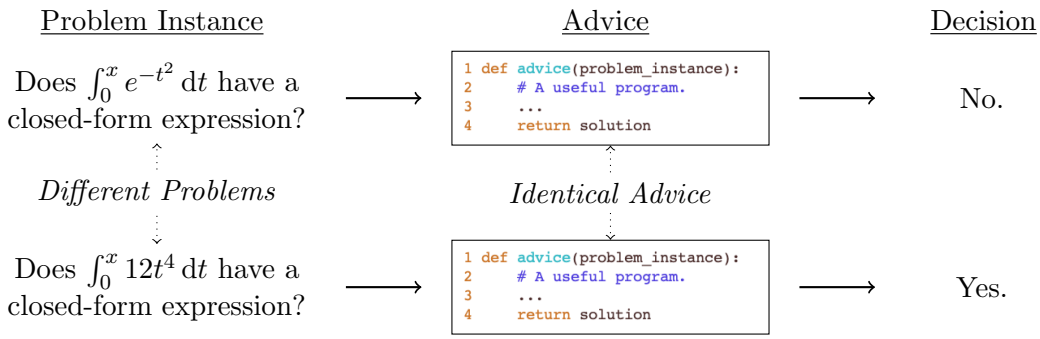


Figure 1.1: Advice as a program to solve problem instances.

instances of the problem, i.e.: $P^{-1}(1)$, and "no" instances of the problem, i.e.: $P^{-1}(0)$. We identify the ability to solve the problem P with the ability to correctly compute $P(x)$ when given any element, or problem instance, x in its domain.[5] The cost of computing $P(x)$ is generally expressed in terms of $|x|$, the length of the particular problem instance x.

An important concept in complexity theory is that of *advice*. Advice is a non-uniform trusted piece of information which is made available to an algorithm with the aim to help it solve instances of a given problem P. "Non-uniform", in this setting, means that we do not constrain or consider the computational cost of *generating* this advice. In other words, it is a resource which is given "for free" and which can even encode information that would otherwise be impossible to compute. We emphasize that advice cannot depend on the specific problem instance $x \in \{0,1\}^*$ which it is used to solve, only on the general problem P and the *length* of the problem instance x. Finally, advice can also be understood as a *program* which a party uses to solve instances of a problem: On input of a problem x, one executes the program encoded in the advice corresponding to the length of x to obtain the correct solution $P(x)$ [Wat09]. This is illustrated in Figure 1.1. The general question concerning advice is whether or not it enlarges the class of problems which can be solved under other constraints: When is computation with advice strictly more powerful than computation without advice? And what precisely is the difference?

While advice was initial conceived as a classical resource, i.e. bit strings, Nishimura and Yamakami formulated during their time at the University of Ottawa a notion of *quantum*

[5] Note the distinction between general problems $P : \{0,1\}^* \to \{0,1\}$, e.g.: "determine if a number is prime", and specific problem instances $x \in \{0,1\}^*$, e.g.: "is the number with binary representation x prime?" One needs to be able to solve all problem instances to be able to solve a problem.

advice where the advice is a quantum state [NY04]. This idea was then subsequently refined by Aaronson [Aar05]. The natural question to ask of quantum advice is to determine in which contexts, if any, it is more powerful than classical advice. See, for example, work by Aaronson and Drucker on this topic [AD14].

Alternatively, we can draw inspiration from the topic of quantum copy-protection and the similarities between advice and programs to ask if we can construct *uncloneable* quantum advice states. This problem was highlighted by Aaronson during QCrypt 2016 [Aar16b].

Contributions. In Chapter 4, we give the first formal definition of a complexity class with *uncloneable advice* and we provide a constructive proof that this class is non-empty. This, and other related observations, represent the first formal results in the field of uncloneable complexity theory.[6]

Our complexity class is denoted **neglQP**/upoly. It is precisely the class of problems $P : \{0,1\}^* \to \{0,1\}$ for which there exists a sequence of quantum advice states $(\rho_\lambda)_{\lambda\in\mathbb{N}}$ which (1) allow a polynomial-time quantum algorithm to compute $P(x)$ with negligible error when given $(\rho_{|x|}, x)$ as input and (2) exhibit a particular type of "functional uncloneability" similar to quantum copy-protection. Specifically, it is infeasible for an adversary to split an advice state into two parts, each allowing separated parties to solve the problem P.

While quantum copy-protection and uncloneable quantum advice achieve similar properties, namely granting parties an uncloneable resource allowing them to compute some function, we highlight two key distinctions between these notions:

1. In the quantum copy-protection scenario, adversaries do not know anything about the function f for which they receive a program state beyond the fact that it was sampled at random from some larger known family $\mathcal{F}$. This randomness is often, if not always, necessary to prove the security of specific quantum copy-protection schemes.

   In contrast, for uncloneable quantum advice, adversaries know in principle the exact function P which the advice state is meant to help them compute, even if actually computing P is beyond their computational means. There is *no randomness* here.

2. In the quantum copy-protection scenario, it must be possible to efficiently generate the program state. No such constraint is placed on uncloneable advice states.

The novel tool which we developed for this work are *ingenerable sequence of bit strings*. In short, an infinite sequence of bit strings, each individually of finite length, is ingenerable if every algorithm — not only efficient ones! — can correctly generate with non-negligible probability only a finite number of elements in this sequence. Using a counting argument

[6] Our preprint [BKL23] was followed shortly by works on uncloneable *proofs* [JK23, GMR23, BGPS24].

over Turing machines, we demonstrate the unconditional existence of such sequences. We then show that any algorithm which "succeeds", for whatever that may mean in the algorithm's specific context, with negligible probability on input of a uniformly random bit string must also succeed with negligible probability when given an element of an ingenerable sequence as input. The proof is essentially to note that an algorithm which typically fails on random inputs but which performs above expectation on elements of an ingenerable sequence can be used to identify these elements among all other bit strings, thus allowing them to be generated with an unacceptably high probability.

Using this property of ingenerable sequences, we derandomize existing cryptographic scenarios which challenge adversaries to clone information contained in quantum states selected at random. We then show how to construct problems in **neglQP**/upoly from these derandomized scenarios. We do so twice: once for a monogamy-of-entanglement game to show the unconditional existence of certain partial maps in **neglQP**/upoly and once assuming the existence of a quantum copy-protection scheme for certain classes of functions to show the conditional existence of certain total maps in **neglQP**/upoly.

1.4 A Technical Formulation of the No-Cloning Theorem

We conclude this introductory chapter by providing, for the sake of completeness, one technical formulation of the no-cloning theorem. Otherwise, such a statement would not appear in these pages, an odd omission for a concept so central to our work.

While we have not yet provided the necessary technical definitions for this theorem (see, for example, Section 2.2 for such definitions), they are all very standard notions in quantum information. Nonetheless, we recall that pure quantum states are modeled as rank-one density operators, physical processes by completely positive trace-preserving maps, *i.e.* channels, and that the trace distance, *i.e.* half the Schatten 1-norm, is an appropriate and normalized measure of distance between quantum states.

Our theorem gives a bound on how well any channel Φ can clone a pair of pure states by giving a lower-bound on the sum of the trace distances between the outputs of Φ and the outputs of a hypothetical perfect cloning operation. In particular, we see that no channel can clone two pure states ρ and σ unless they are equal, i.e. $\mathrm{Tr}(\rho\sigma) = 1$, or orthogonal, i.e. $\mathrm{Tr}(\rho\sigma) = 0$. The idea for this proof can be found in lecture notes by Aaronson [Aar16a], but we do not believe this precise quantification has previously appeared in the literature.

Theorem 1.1. For every rank-one density operators ρ and σ acting on a finite-dimensional complex Hilbert space H and every completely positive trace-preserving map Φ taking linear

operators on H to linear operators on $\mathsf{H} \otimes \mathsf{H}$, we have that

$$\begin{aligned}\frac{1}{2}\|\rho \otimes \rho - \Phi(\rho)\|_1 + \frac{1}{2}\|\sigma \otimes \sigma - \Phi(\sigma)\|_1 &\geq \sqrt{1 - \mathrm{Tr}(\rho\sigma)^4} - \sqrt{1 - \mathrm{Tr}(\rho\sigma)^2} \\ &\geq \left(\sqrt{2} - 1\right) \cdot \mathrm{Tr}(\rho\sigma)^2 \cdot \sqrt{1 - \mathrm{Tr}(\rho\sigma)^2}.\end{aligned} \tag{1.1}$$

Proof: The triangle inequality yields that

$$\frac{1}{2}\|\sigma \otimes \sigma - \Phi(\sigma)\|_1 + \frac{1}{2}\|\Phi(\sigma) - \Phi(\rho)\|_1 + \frac{1}{2}\|\Phi(\rho) - \rho \otimes \rho\|_1 \geq \frac{1}{2}\|\sigma \otimes \sigma - \rho \otimes \rho\|_1 \tag{1.2}$$

which implies that

$$\frac{1}{2}\|\sigma \otimes \sigma - \Phi(\sigma)\|_1 + \frac{1}{2}\|\Phi(\rho) - \rho \otimes \rho\|_1 \geq \frac{1}{2}\|\sigma \otimes \sigma - \rho \otimes \rho\|_1 - \frac{1}{2}\|\Phi(\sigma) - \Phi(\rho)\|_1. \tag{1.3}$$

Since the trace distance is contractive under the action of completely positive trace-preserving maps, we have that

$$\frac{1}{2}\|\sigma \otimes \sigma - \Phi(\sigma)\|_1 + \frac{1}{2}\|\rho \otimes \rho - \Phi(\rho)\|_1 \geq \frac{1}{2}\|\sigma \otimes \sigma - \rho \otimes \rho\|_1 - \frac{1}{2}\|\sigma - \rho\|_1. \tag{1.4}$$

Now, recalling that the trace distance between two rank-one density operators ρ and σ is given by $\sqrt{1 - \mathrm{Tr}(\rho\sigma)^2}$ and that $\mathrm{Tr}((\rho \otimes \rho)(\sigma \otimes \sigma)) = \mathrm{Tr}(\rho\sigma)^2$ yields our first inequality.

Our second inequality follows from the observation that

$$\sqrt{1 - \mathrm{Tr}(\rho\sigma)^4} - \sqrt{1 - \mathrm{Tr}(\rho\sigma)^2} = \left(\sqrt{1 + \mathrm{Tr}(\rho\sigma)^2} - 1\right)\sqrt{1 - \mathrm{Tr}(\rho\sigma)^2} \tag{1.5}$$

and that $\sqrt{1 + \mathrm{Tr}(\rho\sigma)^2} - 1 \geq (\sqrt{2} - 1)\,\mathrm{Tr}(\rho\sigma)^2$ whenever $0 \leq \mathrm{Tr}(\rho\sigma) \leq 1$, as must be the case for density operators. Note that the right-hand side of this inequality is simply the linear interpolation of $x \mapsto \sqrt{1 + x} - 1$ between $x = 0$ and $x = 1$ with $x = \mathrm{Tr}(\rho\sigma)^2$. ∎

Topic	Result	Section
Tamper Evidence	Showing that tamper evidence implies encryption.	Section 2.4
	Formalizing an existing construction of quantum money schemes from tamper-evident schemes.	Section 2.5
	Constructing revocation schemes from tamper-evident schemes, and vice-versa.	Section 2.6
	Showing that any revocation scheme must also be an encryption scheme.	Section 2.6.4
	Showing that tamper evidence implies neither uncloneable encryption nor authentication.	Section 2.7
Secure Software Leasing	Defining "honest-malicious" security for copy-protection schemes.	Section 3.3.1
	Constructing secure software leasing schemes from certain honest-malicious secure copy-protection schemes.	Section 3.4.3
	Constructing a secure software leasing scheme for point functions from total authentication schemes.	Section 3.5
Uncloneable Advice	Defining and proving the existence of ingenerable sequences of bit strings.	Section 4.3
	Defining cloning complexity classes and making simple observations about them.	Section 4.4
	Defining a computational complexity class with uncloneable quantum advice.	Section 4.5.2
	Constructing a promise problem with uncloneable quantum advice.	Section 4.5.3
	Constructing a language with uncloneable quantum advice under some assumptions.	Section 4.5.4

Table 1.1: Main novel contributions in this thesis.

Chapter 2

Tamper-Evident Encryption*

Chapter Abstract

A tamper-evident encryption scheme is a non-interactive symmetric-key encryption scheme mapping classical messages to quantum ciphertexts such that an honest recipient of a ciphertext can detect with high probability any meaningful eavesdropping. This quantum cryptographic primitive was first introduced by Gottesman in 2003. Beyond formally defining this security notion, Gottesman's work had three main contributions: showing that any quantum authentication scheme is also a tamper-evident scheme, noting that a quantum key distribution scheme can be constructed from any tamper-evident scheme, and constructing a prepare-and-measure tamper-evident scheme using only Wiesner states inspired by Shor and Preskill's proof of security for the BB84 quantum key distribution scheme.

In this chapter, we further our understanding of tamper-evident encryption by formally relating it to other quantum cryptographic primitives in an information-theoretic setting. In particular, we show that tamper evidence implies encryption, answering a question left open by Gottesman, we show that it can be constructed from any encryption scheme with revocation and vice-versa, and we formalize an existing sketch of a construction of quantum money from any tamper-evident encryption scheme. These results also yield as a corollary that any scheme allowing the revocation of a message must be an encryption scheme. We also show separations between tamper evidence and other primitives, notably showing that tamper evidence does not imply authentication and does not imply uncloneable encryption.

*This chapter reflects work which has been made available on the arXiv [Lor24]. It has been lightly edited for inclusion in this thesis.

Chapter Contents

2.1 Introduction

Consider the following scenario: Alice wants to send a classical message to Bob over a communication channel which is accessible to a malicious and eavesdropping Eve. Alice and Bob wish for Eve to never learn any information on this message. To help them, they have a shared secret key which, at the time of transmission, is unknown to Eve.

Alice and Bob can achieve their goal by using any information-theoretically secure classical symmetric-key encryption scheme, such as the one-time pad. However, if they use a scheme which produces a classical ciphertext, Eve can always make and keep a copy of it. Moreover, it is impossible for Alice and Bob to detect this eavesdropping. The secrecy of the message is then maintained if and only if Eve never learns the secret key at a later time. This general scenario is illustrated in Figure 2.1.

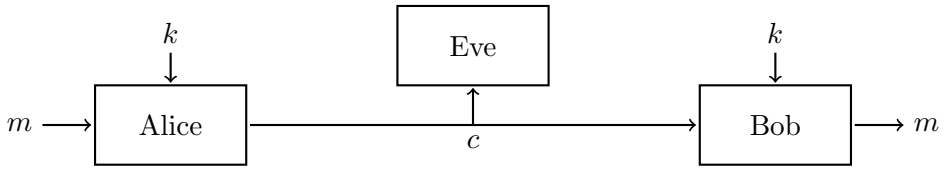


Figure 2.1: A classical symmetric-key encryption scenario. Alice and Bob share a classical key k which Alice uses to encrypt a classical plaintext m as a classical ciphertext c from which Bob recovers m. An eavesdropping Eve obtains, undetected, a copy of c.

In the theoretical study of a cryptographic system, it is often asserted and assumed that an eavesdropper never learns the secret key k and a proof of security in then derived in this context. In practice, however, ensuring that this assumption holds over a long period of time is a non-trivial task which falls under the umbrella of *key management*. We do not discuss this topic any further in this work, but we direct the interested reader to guidelines provided by the Unites States' *National Institute of Standards and Technology* (NIST) on this topic [Bar20].

However, due to the *information-disturbance principle* [Fuc96],[2] also known as the *information-disturbance trade-off*, it may be impossible for Eve to make, undetected, a copy of the ciphertext and wait to learn the secret key at a later point in time, if this ciphertext is a quantum state. Indeed, this principle implies that extracting previously unknown information from a quantum state must, in general, change it.

By leveraging this property of quantum mechanics, Alice and Bob can detect attempts

[2]The information-disturbance principle is closely related to the *no-cloning principle* [Par70, Die82, WZ82, Ort18] which states that it is, in general, impossible to make a perfect copy of an unknown quantum state. Notably, if one could make such perfect copies, then information on a state could be obtained without disturbing it by making and measuring such a copy.

by Eve to eavesdrop on their communication. More precisely, we can devise schemes where Bob, while decrypting a ciphertext, also determines whether he *accepts* the transmission — indicating that he did not detect any eavesdropping — or *rejects* it. This is illustrated in Figure 2.2. If Bob accepts, then he is assured that Eve cannot learn anything about the message *even if she later learns the key*. Of course, such a method to *detect* eavesdropping does not *prevent* it. However, we can suppose that Alice and Bob may go above-and-beyond in protecting their key or implement other mitigating measures if they detect any eavesdropping.

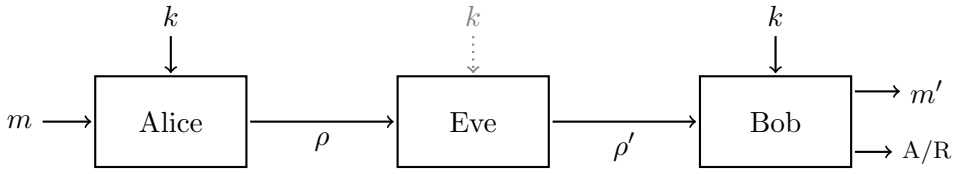


Figure 2.2: The communication scenario considered for tamper-evident schemes. Alice and Bob share a classical key k which Alice uses to encrypt a classical message m as a *quantum* ciphertext ρ. Eve attempts to eavesdrop on this ciphertext, possibly introducing some changes. Bob then decrypts, possibly obtaining different a message, and either accepts or rejects the transmission to indicate if he detected Eve eavesdropping or not. Eve only learns the key k *after* her eavesdropping attempt.

In 2003, Gottesman formalized this security notion and gave two constructions achieving it [Got03]. Following prior work [BL20], we refer to schemes achieving this notion as *tamper-evident schemes*.[3] Gottesman's first construction consisted in showing that any quantum authentication scheme[4] [BCG+02] was already, without any modification, a tamper-evident scheme. The second was a specific construction obtained by adapting Bennet and Brassard's quantum key distribution scheme [BB84] and the Shor-Preskill proof of its security [SP00].

While tamper evidence and quantum key distribution are similar in the sense that they both attempt to detect an eavesdropper, they do exhibit important differences. The latter is typically interactive, does not assume a pre-shared key, and produces a shared random string between Alive and Bob, while the former is non-interactive, assumes a pre-shared key, and allows Alice to send a *specific* string to Bob.

The specifics of Gottesman's work on tamper evidence has attracted little attention in the twenty years since its publication. Tamper evidence is often cited as an example

[3] Gottesman originally called tamper-evident schemes "uncloneable encryption" schemes, but conceded that this may not be the proper term. Indeed, nothing *prevents* Eve from cloning the ciphertext, or at least the information it encodes. She simply cannot do without leaving evidence of her tampering.

[4] Quantum authentication schemes are essentially message authentication codes for quantum states.

of a security notion offered by quantum cryptography which cannot be achieved with only classical tools, but little work has been done to formally situate tamper evidence with respect to other quantum cryptographic primitives. This leaves the following as an open avenue of research:

> *How does tamper evidence fit in the landscape of cryptographic security notions? Which notions imply it? Which does it imply? Which are independent?*

We offer in this work multiple answers to these questions.

Organization. We complete this section by reviewing prior work on tamper evidence in Section 2.1.1, providing an overview of our contributions and results in Section 2.1.2, and highlighting a few open questions in Section 2.1.3.

As for the remainder, we review in detail the necessary mathematical background and notions from quantum information theory in Section 2.2. This includes an in-depth discussion of the trace distance, a crucial notion in this work, as well as a lemma pertaining to gentle measurements. We then state the basic definitions surrounding tamper evidence in Section 2.3. Finally, we conclude with the full details of our results in Sections 2.4 to 2.7.

Acknowledgements. The author thanks Anne Broadbent for many fruitful discussions over the course of this work, in particular concerning the connections between tamper evidence, revocation, and certified deletion. The author also acknowledges support from the Natural Sciences and Engineering Research Council of Canada (NSERC) during his time at the University of Ottawa.

2.1.1 The State of the Art

We review the works on tamper-evident schemes and quickly summarize their results.

Gottesman's original work [Got03] showed that every quantum authentication scheme is also a tamper-evident scheme. It also highlighted a connection with quantum key distribution (QKD) by sketching a construction of a QKD scheme from any tamper-evident scheme and by obtaining a tamper-evident scheme based on Bennet and Brassard's QKD scheme and its analysis by Shor and Preskill. Finally, a few years later, Gottesman sketched a construction of a quantum money scheme from any tamper-evident scheme [Got11].

Leermaker and Škorić established the existence of a tamper-evident scheme which also achieved the notion of *key recycling* [LŠ20], essentially meaning that a portion of a key can be reused with little security degradation [FS17].

Finally, van der Vecht, Coiteux-Roy, and Škorić have also given a tamper-evident scheme with short keys [vdVCRŠ22].

2.1.2 Our Contributions

Before giving an overview of our contributions, we emphasize one place where we do *not* make a contribution: we do not define any new tamper-evident schemes, nor any other type of cryptographic schemes, "from scratch" or from first principles. Instead, we define various constructions mapping arbitrary tamper-evident schemes to other cryptographic schemes or generically constructing tamper-evident schemes from other cryptographic primitives.

Due to this level of abstraction and the wide variety of notions under consideration, we place a particular level of care in stating our various definitions and formally proving our results. We emphasize that Gottesman has already shown the unconditional existence of arbitrarily good tamper-evident schemes [Got03]. This is recorded here in Theorem 2.22. Thus, the constructions we define and study can be instantiated and the conclusions we derive hold unconditionally.

Our contributions can be divided into four categories. We have three "positive" categories: we show that tamper evidence implies encryption, we formalize how it it yields quantum money schemes, and we relate it to encryption schemes with revocation. These are developed in Sections 2.4 to 2.6, respectively. We conclude this chapter in Section 2.7 with our final category where we collect various "negative" results. There, we provide counterexamples by constructing various schemes which are tamper evident but fail to achieve other properties of interest.

In what follows, we review the three positive categories and we indicate the relevant counterexamples from the fourth one as they arise.

2.1.2.1 Tamper Evidence, Encryption, and Authentication

Gottesman defines a tamper-evident scheme as an *encryption scheme* which *permits the honest receiver to detect meaningful eavesdropping*. We refer to this second property as the "tamper-evident requirement". Gottesman then states the following [Got03]:

> Note that [a tamper-evident scheme] by definition must also encrypt the message, so that Eve, even if she gets caught, cannot read the message until she learns the key. It is unclear whether this is an independent condition or whether it would follow from the [tamper-evident] requirement alone. (A classical message sent completely unencrypted can always be copied, but it may be possible to create partially encrypted messages which are [tamper-evident].)

Contribution. We answer this question in Section 2.4 by showing that encryption is *not* an independent property and that *it does* follow from the tamper-evident requirement alone.

There was already some evidence pointing towards this result. Gottesman had shown that quantum authentication schemes were also, without any modification, tamper-evident schemes [Got03] and nearly concurrent work by Barnum, Crépau, Gottesman, Smith, and Tapp had shown that any quantum authentication scheme is, without any modification, a quantum encryption scheme [BCG+02]. Thus, it was already known that tamper evidence and encryption were somewhat related inasmuch that they are both implied by a common cryptographic property:[5]

$$\text{Quantum Authentication} \implies (\text{Tamper Evidence} \wedge \text{Encryption of Quantum States})\,. \tag{2.1}$$

We show that a finer-grained version of the previous implication holds, namely that

$$\text{Quantum Authentication} \implies \text{Tamper Evidence} \implies \text{Encryption of Classical Messages} \tag{2.2}$$

and that the converses do not hold. Specifically, we show that the second implication holds in Theorem 2.30 and we show that the converse of the first does not hold in Section 2.7.3.1. The first implication was shown by Gottesman and the second converse trivially does not hold. We note that Gottesman did *conjecture* the existence of non-authenticating tamper-evident schemes, but did not prove this fact [Got03].

From Equation 2.1 to Equation 2.2, we have shifted from the encryption of *quantum states* to the more restrictive — and hence less powerful — encryption of *classical messages*, seen as a subset of quantum states. This is a necessary specification as we show in Section 2.7.3.2 how to construct a tamper-evident scheme which does not encrypt all quantum states. Thus, a more appropriate representation of the relations between encryption, tamper-evidence, and authentication exhibited in this work is given by the Venn diagram in Figure 2.3.

Proving that tamper evidence implies encryption. Our proof method for showing that tamper evidence implies encryption is similar to Barnum *et al.*'s proof that quantum authentication implies the encryption of quantum states [BCG+02]. We proceed, as they did, in three steps.

Step 1: Show that bad encryption implies bad tamper evidence. The main novel observation underpinning our proof is that a bad encryption scheme must be a bad tamper-evident scheme (Theorem 2.24). Indeed, consider an encoding scheme which admits two messages whose encodings are near-perfectly distinguishable, even without knowledge of the key. By definition, such a scheme cannot be a good encryption scheme. Moreover, via a gentle

[5] These implications hold with at most a polynomial loss of security, and no loss of correctness.

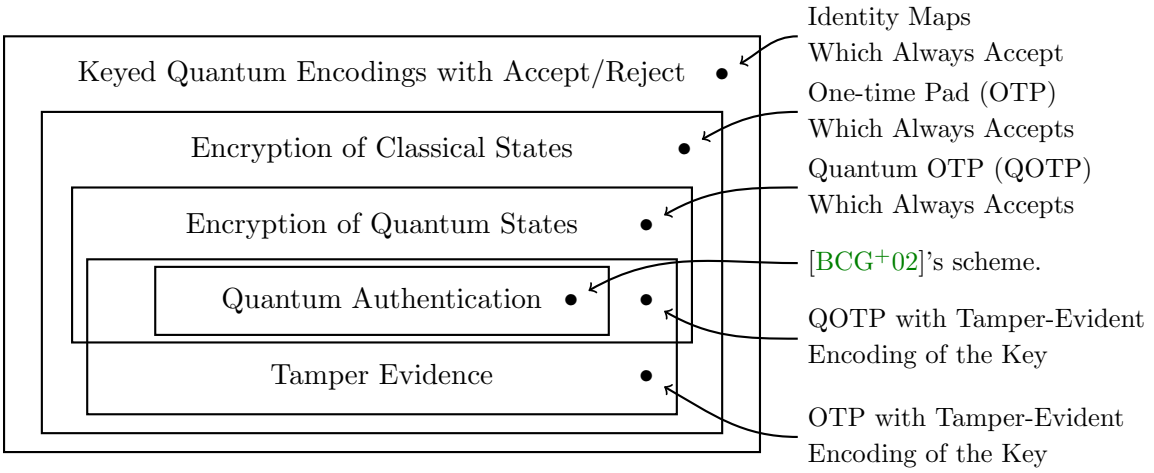


Figure 2.3: A Venn diagram showing the relations between quantum authentication schemes, tamper-evident schemes, and encryption schemes within all keyed quantum encoding schemes where the decoding map must also produce an accept/reject flag. Some schemes are pinpointed to illustrate that each discernable region shown here is distinct. The identity, OTP, and QOTP schemes have decoding maps augmented to always accept.

measurement, the encoded states can be distinguished essentially without disturbing them. This allows an eavesdropper to obtain information on the ciphertext, and hence on the underlying message, in a way which will not be detected by the honest receiver with high probability. Such a scheme thus fails to be sufficiently tamper-evident.

Unfortunately, we cannot simply take the contrapositive of this result and be satisfied. The precise quantification we obtain is not enough to conclude via this method that good tamper-evident schemes are good encryption schemes. A bit more work is needed.

Step 2: Analyse the parallel composition of tamper-evident schemes. Next, we show that the parallel composition of tamper-evident schemes only yields a *linear* loss of security (Theorem 2.27). This is in contrast to the fact made explicit by Barnum *et al.* that the parallel composition of encryption schemes yields an *exponential* loss in security [BCG+02].

Step 3: Leverage these different rates of security degradation. We play this linear loss of security as a tamper-evident scheme against the exponential loss of security as an encryption scheme to obtain our result.

Indeed, if a good tamper-evident scheme was not a sufficiently good encryption scheme, then the repeated parallel composition of this scheme would yield a contradiction: It would pass the threshold of being a bad encryption scheme before passing the threshold of being a bad tamper-evident scheme, contradicting our prior result.

Proving that tamper evidence does not imply authentication. To show in Section 2.7.3.1 that tamper evidence does not imply authentication, we construct a tamper-evident scheme which makes it easy for an adversary to change the encoded message without detection. The idea is to encrypt a message m with a classical one-time pad p, and then encrypt p with a tamper-evident scheme. The overall ciphertext is then composed of two parts: the classical string $m \oplus p$ and the tamper-evident encryption of p.

Upon reception of a ciphertext from this scheme, the honest recipient can check if any eavesdropping occurred on the tamper-evident encoding of p. If none is detected, then they can be assured that an adversary is unlikely to be able to learn m even if they kept a copy of $m \oplus p$ and later learn the key used by the tamper-evident scheme. Indeed, the tamper-evident guarantee ensures that they are unlikely to learn p in this scenario.

On the other hand, the only part of the ciphertext which depends on the message — the classical string $m \oplus p$ — is sent without any additional layers of protection. This means that an adversary can pick any string e and modify this part of the ciphertext to become $(m \oplus e) \oplus p$ without being detected by the honest recipient. Indeed, the recipient will simply believe that $m \oplus e$ was the original message sent. In other words, this construction yields a tamper-evident scheme which is *malleable* because the one-time pad is itself malleable.

Other separations. As for the remaining separating examples illustrated in Figure 2.3, we claim that the always-accepting identity maps, the always-accepting one-time pad, and the always-accepting quantum one-time pad are trivial to situate in this diagram. As for the quantum one-time pad with a tamper-evident encoded key, we claim without further proof that it can be situated with an argument analogous to the one we have just made for the classical one-time pad with a tamper-evident encoded key.

2.1.2.2 Formalizing Quantum-Money from Tamper Evidence

Quantum money was perhaps the first widely known quantum cryptographic primitive, having been published by Wiesner in 1983 [Wie83] about a decade after it was initially conceived.[6] In a quantum money scheme, a bank issues "quantum banknotes" which, in general, consist of a quantum state ρ_s and a classical piece of information s often understood to be the serial number of the banknote. Any user may return a banknote to the bank to have it validated to ensure that it is not a forgery. The relevant security guarantee here is that it should be infeasible for any adversary to turn one honestly produced banknote (ρ_s, s) into *two* banknotes (σ', s') and (σ'', s'') which would *both* be accepted by the bank. The basic template for the bank's verification process is for it to maintain a database associating each

[6] Those interested should read Brassard's account of the early days of quantum cryptography [Bra05].

public serial number s to a secret key k_s which the bank can use to validate the quantum state ρ_s.

In this foundational work, Wiesner presented an explicit quantum money scheme and sketched an argument that it should achieve the desired security notion. A formal proof of security was presented about thirty years later [MVW13], prompted by discussions on an internet forum. During these same discussions, Gottesman sketched a way to transform any tamper-evident scheme into a quantum money scheme [Got11].

Gottesman's idea was fairly straightforward: The bank samples a random key k, message m, and serial number s. It keeps (s, k, m) in its database and encodes the message m with the tamper-evident scheme using the key k to obtain a quantum state ρ. The bank then issues (ρ, s) as a banknote. To verify the authenticity of a supposed banknote (ρ', s'), the bank recovers the corresponding key k' and message m' from its database and decodes the state ρ' with the tamper-evident scheme using the key k'. The bank accepts the banknote if and only if the message m' is recovered *and* if it does not detect any eavesdropping. At a high-level, the security of this quantum money scheme follows from the security of the tamper-evident scheme. If the bank does not detect any eavesdropping, then a counterfeiter is highly unlikely to have obtained any information on the message, making it difficult for the counterfeiter to have produced a second valid encoding of this same message.

Contribution. We formalize Gottesman's construction and formally prove and quantify the security of the resulting quantum money scheme in Section 2.5. Notably, we account for necessary technical details which were not directly addressed in the initial sketch.

Related counterexamples. Interestingly, we later show in Section 2.7 that it is *necessary* in this construction for the bank to check both that no eavesdropping has occurred *and* that the correct message is recovered. We do so by constructing two explicit counterexamples.

First, we construct in Section 2.7.1 a tamper-evident scheme which allows an adversary given one honestly produced ciphertext to produce two states which will both decode to the original plaintext, but neither of which will pass the eavesdropping test. Our construction for this counterexample is a scheme which produces two tamper-evident encodings of the same message m with independently sampled keys and that this overall ciphertext passes the eavesdropping check if and only if *both* encryptions of m pass their individual eavesdropping check. In particular, this implies that tamper-evident schemes need not be uncloneable encryption schemes [BL20].

For our second class of counterexamples, we construct in Section 2.7.2 secure tamper-evident schemes allowing an adversary given one honestly produced ciphertext to create two states which will both pass the eavesdropping check, but neither of which are likely to

decode to the correct message. Our counterexample construction splits the plaintext with a simple group-based 2-out-of-2 classical secret sharing scheme (e.g.: [Sha79]) and then encode both of these shares with a tamper-evident scheme using independent keys. The overall ciphertext passes the eavesdropping check if at least one, but not necessarily both, encrypted shares pass their individual eavesdropping check. By splitting the shares, an adversary can create two states passing verification, but that cannot individually be used to recover the plaintext.

2.1.2.3 Revocation From Tamper Evidence, and Vice-Versa

An encryption scheme with revocation, or simply a *revocation scheme*, is a symmetric-key encryption scheme equipped with an additional revocation procedure. This procedure allows the recipient of a ciphertext, Bob, to return it to the sender, Alice. If Alice accepts the return *and* Bob did not yet know the key, then Alice is assured that Bob has learned no information on the plaintext and that this will remain the case *even if he later learns the key*. Evidently, it is impossible to achieve this if the ciphertext is classical, but it does become possible by using quantum ciphertexts due to the information-disturbance principle. In general, a revocation procedure could consist in multiple rounds of classical or quantum communication between Alice and Bob. In this work however, we will only consider revocation procedures which consists of a single classical or quantum message from Bob to Alice.

Revocation schemes were first studied by Unruh [Unr15]. There, the revocation procedure consisted of Bob simply returning the quantum ciphertext followed by Alice verifying that the ciphertext state was unchanged. A few years later, Broadbent and Islam identified an important subclass of revocation schemes, still with quantum ciphertexts, which they called *encryption with certified deletion* [BI20], or simply a certified deletion scheme. In a nutshell, a certified deletion scheme is a revocation scheme where the revocation procedure only requires a single *classical* message. In other words, Bob produces a classical bit string, *i.e.* a *certificate*, proving to Alice that he has effectively deleted the ciphertext. Broadbent and Islam also constructed a certified deletion scheme and provided a complete proof of security.[7]

Certified deletion has since become an important, promising, and fruitful area of research in quantum cryptography. Research on this topic has focused on expanding the scope of primitives for which a notion of certified deletion can be defined and achieved — *e.g.:* the works of Hiroka, Morimae, Nishimaki, and Yamakawa [HMNY21] and of Bartusek and Khu-

[7] Contemporary and independent work by Coiteux-Roy and Wolf [CRW19] also examined a notion and protocol similar to the ones presented by Broadbent and Islam but gave no satisfactory proof of security.

rana [BK23] which consider public-key, attribute-based, and fully-homomorphic encryption schemes with certified deletion — or on reducing the assumptions needed to achieve this notion, *e.g.*: Kundu and Tan's device-independent protocol for certified deletion [KT23].

Contribution. We demonstrate in Section 2.6 a close relationship between tamper evidence and the notions of revocation and certified deletion. We do so in two ways.

First, we provide in Section 2.6.2, specifically Definition 2.44, an information-theoretic definition of security for revocation schemes which is similar to Gottesman's definition of security for tamper-evident schemes. We then demonstrate that, up to a linear loss in the security quantification, this new definition is equivalent to the existing game-based definition of revocation found in the literature.

Second, we give in Section 2.6.3 two generic constructions relating tamper-evidence and revocation. We describe in Section 2.6.3.1 a construction which transforms any tamper-evident scheme into a revocation scheme and in Section 2.6.3.2 a construction which transforms any revocation scheme — including any certified deletion scheme — into a tamper-evident scheme. In both of these constructions, we show that there is at most a polynomial loss of security and correctness.

Conceptually, the relationship between tamper evidence and revocation follows from the fact that we can, at a high-level, identify the roles of parties in a revocation scheme with roles of parties in a tamper-evident scheme. First, we note that we can identify the role of "revocation Bob" with "tamper-evident Eve" as they both attempt to extract undetected information from the quantum ciphertext. To achieve either security notion, the remaining honest parties must ensure that these adversaries remain ignorant of the plaintext even if they later learn the key. To complete the comparison, we see "revocation Alice" as playing the part of both "tamper-evident Alice" while generating the ciphertext and of "tamper-evident Bob" when determining if she accepts the state she received. This is shown in Figure 2.4.

Our construction of a revocation scheme from a tamper-evident scheme is also illustrated by Figure 2.4. We show that Bob returning the ciphertext unmodified and Alice checking the flag produced by the tamper-evident decryption procedure yields a secure revocation procedure.

The construction of a tamper-evident scheme from a revocation scheme is a bit more involved. The main observation we use here is that being able to prove that you have deleted the only instance of some information also implicitly proves that you were the only party in possession of this information.[8] Hence, Bob in a tamper-evident scenario can convince himself that there was no meaningful eavesdropping by successfully executing

[8] We thank Anne Broadbent for first making and sharing this observation.

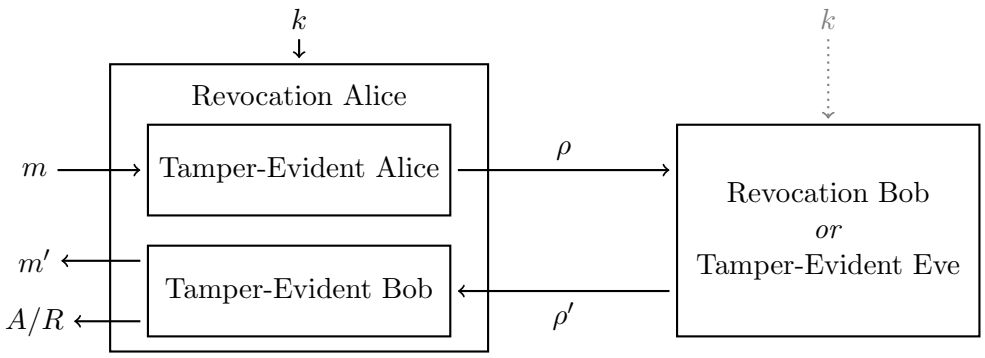


Figure 2.4: Conceptual identification between the parties in a tamper-evident scheme and revocation scheme. Contrast with Figure 2.2. Note that in the revocation scenario Alice does not output a decoded message m' and ρ' may be on a different space than ρ.

and verifying a revocation procedure on the ciphertext by himself. At first glance, this is not quite sufficient as Bob, in a tamper-evident scheme, must also recover the message. However, if Bob executes a revocation procedure, perhaps even one from a certified deletion scheme, how can he hope to also recover the message? Has he not deleted the ciphertext? To overcome this issue, we have Bob instead execute a gentle-measurement version of the complete revocation procedure and subsequent verification where only the final accept/reject output, and not the intermediate state that would be returned, is measured.

Finally, we obtain as a direct corollary that every revocation scheme must also be an encryption scheme. This follows from the fact that tamper evidence implies encryption, that we can construct a tamper-evident scheme from any revocation scheme, and that this construction leaves both the encryption and key procedures unmodified. This is recorded in Section 2.6.4.

2.1.3 Open Questions

While our work advances the understanding of the relations between tamper evidence and other cryptographic security notions, it leaves much to be done. We highlight here a few questions which remain largely open.

Do our results on tamper evidence hold in a computational setting? The whole of this work is completed in the information-theoretic setting where we do not formally consider the computational costs to implement or attack cryptographic schemes. However, tamper evidence can also be studied in a setting where such computational costs are considered. A natural question then is to ask which of our results hold in such a setting where schemes

and attacks must be efficiently implementable. We expect most of them to hold, and we believe most of our constructions remain efficient when applied to efficient schemes.

One notable exception is our formalization of Gottesman's construction of a quantum money scheme in Section 2.5.2 when the underlying tamper-evident scheme is not perfectly correct. There, we require the bank to sample key-message pairs which will lead to a correct decryption with sufficiently high probability. It is unclear if such a sampling can always be done efficiently. However, as we discuss at the end of Section 2.5.2, this problem can be side-stepped by weakening the correctness notion for quantum money schemes from one where we require *all* banknotes produced by a bank to be accepted with high probability, to one where we only require *the large majority* of banknotes to be accepted with high probability.

Can we formalize the connection between tamper evidence and quantum key distribution? Gottesman noted that a quantum key distribution scheme can be easily obtained from a tamper-evident scheme [Got03]. Indeed, have Alice encode a random string s with a random key k using a tamper-evident scheme and send the resulting quantum state to Bob. Bob then acknowledges reception of the state over an authenticated classical channel, after which Alice discloses k over this same channel. Bob can now decode and verify the quantum state with k. If Bob accepts, they can be sure that an eavesdropping Eve has essentially no information on the string s, even if Eve also learned k by listening to the authenticated channel. Moreover, Gottesman sketched how minimal changes to Bennet and Brassard's QKD scheme [BB84] yields a tamper-evident scheme. At first glance, this construction is specific to the BB84 scheme and may not be applicable to all QKD schemes.

This state of affairs yields two open questions which we have not broached in this work:

1. Can one formally prove and quantify the security of a QKD scheme obtained from Gottesman's generic construction based on any tamper-evident scheme?

2. Can one formulate a minimal set of conditions on QKD schemes such that satisfying these conditions ensures that it is possible to generically transform it into a tamper-evident scheme? And, if so, what is the level of security achieved by the resulting tamper-evident scheme?

We expect answering the first question to be a straightforward exercise. However, the second appears more complex to tackle.

Is there a device-independent tamper-evident scheme? A device-independent quantum cryptographic scheme is one in which the honest users need not have complete control

of their quantum devices to still achieve their desired security notions. Such schemes exist for both QKD (*e.g.*: [VV14]) and certified deletion (*e.g.* [KT23]). As both of these security notions are related to tamper evidence, it would appear likely that the methods used to attain them in a device-independent manner should also allow us to construct a device-independent tamper-evident scheme. Is this indeed the case? In particular, it is unclear if our construction of a tamper-evident scheme from any certified deletion scheme would preserve device-independence, and we are inclined to believe it would not.

2.2 Preliminaries

We review in Section 2.2.1 the basic mathematical concepts used in this work, how they relate to quantum mechanics, and the notation we use. Next, in Section 2.2.2, we examine in some detail the notion of the trace distance between two linear operators, a concept central to this work. In particular, we highlight how some well known properties of the trace distance between density operators generalize to larger classes of linear operators, something which will be necessary for us. Finally, in Section 2.2.3, we examine the concept of a gentle measurement.

2.2.1 Notation and Basic Concepts

Except for some differences in notation, we generally follow Watrous [Wat18]. We assume the reader is familiar with the notion of tensor products on Hilbert spaces.

Sets, alphabets, and Hilbert spaces. Sets will generally be denoted with "calligraphic" uppercase characters such as $\mathcal{X}$ or $\mathcal{M}$. An *alphabet* is a non-empty finite set. All Hilbert spaces in this work are complex and finite dimensional and hence isomorphic to $\mathbb{C}^n$ for a suitable value of $n \in \mathbb{N}$. We often use the Dirac, or "bra-ket" notation [Dir39], to denote the elements of a vector space and their duals, *i.e.:* the dual of a vector $|\psi\rangle$ is $|\psi\rangle^\dagger = \langle\psi|$. For any alphabet $\mathcal{A}$, we let $\mathbb{C}^{\mathcal{A}}$ be a Hilbert space which admits the set $\{|a\rangle : a \in \mathcal{A}\}$ as an orthonormal basis and we call this distinguished basis the *computational basis* of the space.[9] We will typically denote Hilbert spaces using a sans-serif font, such as H, and moreover aim to systematically match symbols between alphabets and Hilbert spaces constructed from them, for example $\mathsf{X} = \mathbb{C}^{\mathcal{X}}$.

[9] More formally, we take $\mathbb{C}^{\mathcal{A}}$ to be the set of all maps $f : \mathcal{A} \to \mathbb{C}$ equipped with the usual point-wise addition, i.e.: $(f+g)(a) = f(a) + g(a)$, the usual scalar multiplication, i.e. $(c \cdot f)(a) = c \cdot f(a)$, and the inner product $(f, g) \mapsto \sum_{a \in \mathcal{A}} f(a)^\dagger \cdot g(a)$ where $c^\dagger$ denotes the complex conjugate of c. For all $a \in \mathcal{A}$, we identify the element $|a\rangle$ with the map taking a to 1 and all other elements of $\mathcal{A}$ to 0. See [Wat18, Sec. 1.1.1] for more details on this construction. The details of this construction are never explicity used in this work.

Unless otherwise specified, we take the norm on a Hilbert space to be the square root of the inner-product of an element with itself: $\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}$.

We identify $\mathbb{C}$ with $\mathbb{C}^{\mathcal{S}}$ for any set $\mathcal{S}$ composed of a single element. A common example is $\mathbb{C} = \mathbb{C}^{\{\varepsilon\}}$ where $\varepsilon \in \{\mathtt{0},\mathtt{1}\}^0$ is the unique bit string of length 0. Note that we will distinguish between the numbers $0, 1 \in \mathbb{N}$ and the symbols $\mathtt{0}$ and $\mathtt{1}$ used to denote the states of a classical bit.[10]

Operators on Hilbert spaces, their adjoints, and the trace. Let H and H' be Hilbert spaces. We denote the set of linear operators whose domain and codomain are H and H', respectively, by $\mathcal{L}(\mathsf{H},\mathsf{H}')$ and we let $\mathcal{U}(\mathsf{H},\mathsf{H}') \subseteq \mathcal{L}(\mathsf{H},\mathsf{H}')$ denote the subset of these operators which are *isometries*, i.e.: those which preserve the norm. If $\mathsf{H}' = \mathsf{H}$, we abbreviate these notations to $\mathcal{L}(\mathsf{H})$ and $\mathcal{U}(\mathsf{H})$, respectively. Thus, $\mathcal{U}(\mathsf{H})$ is the set of *unitary* operators on H.

The identity map on a Hilbert space H is denoted I_{H} and is unitary. For any linear operator $L \in \mathcal{L}(\mathsf{H},\mathsf{H}')$, we define its *adjoint*, denoted $L^\dagger$, as the unique element of $\mathcal{L}(\mathsf{H}',\mathsf{H})$ satisfying $\left(L^\dagger |a\rangle\right)^\dagger |b\rangle = \langle a| L |b\rangle$ for all $|a\rangle \in \mathsf{H}'$ and $|b\rangle \in \mathsf{H}$. For any Hilbert space H, the *trace* $\mathrm{Tr}_{\mathsf{H}} : \mathcal{L}(\mathsf{H}) \to \mathbb{C}$ is the unique linear map such that $\mathrm{Tr}_{\mathsf{H}}(|\psi\rangle\langle\phi|) = \langle\phi|\psi\rangle$ for all elements $|\psi\rangle, |\phi\rangle \in \mathsf{H}$. Recall that the trace is *cyclical*: $\mathrm{Tr}_{\mathsf{C}}(ABC) = \mathrm{Tr}_{\mathsf{A}}(BCA) = \mathrm{Tr}_{\mathsf{B}}(CAB)$ for any linear operators $A \in \mathcal{L}(\mathsf{A},\mathsf{C})$, $B \in \mathcal{L}(\mathsf{B},\mathsf{A})$, and $C \in \mathcal{L}(\mathsf{C},\mathsf{B})$ and that it also satisfies $\mathrm{Tr}_{\mathsf{H}}(L) = \sum_{i\in\mathcal{J}} \langle b_j| L |b_j\rangle$ for any $L \in \mathcal{L}(\mathsf{H})$ and any orthonormal basis $\{|b_j\rangle\}_{j\in\mathcal{J}}$ for H.

A linear operator $X \in \mathcal{L}(\mathsf{H})$ is said to be *positive semidefinite* if there exists a linear operator $Y \in \mathcal{L}(\mathsf{H})$ such that $X = Y^\dagger Y$. This implies that $\langle\psi| X |\psi\rangle$ is a real non-negative number for any element $|\psi\rangle \in \mathsf{H}$. We let $\mathrm{Pos}(\mathsf{H})$ denote the set of all positive operators on H. We let $\mathcal{D}(\mathsf{H})$ denote the set of *density* operators, which is to say positive semidefinite operators with a trace of one. We also define $\mathcal{D}_\bullet(\mathsf{H})$ be the set of *subnormalized density operators*, which is to say positive semidefinite operators whose trace is an element of the real interval $[0, 1]$. Finally, an operator $X \in \mathcal{L}(\mathsf{H})$ is *Hermitian* if it is self-adjoint, which is to say that $X^\dagger = X$. We denote the set of Hermitian operators on H by $\mathrm{Herm}(\mathsf{H})$. Every Hermitian operator X admits a *square root* $\sqrt{X}$ which is the unique positive semidefinite operator satisfying $\sqrt{X}\sqrt{X} = X$.

Thus, for any Hilbert space H we have the inclusions

$$\mathcal{D}(\mathsf{H}) \subseteq \mathcal{D}_\bullet(\mathsf{H}) \subseteq \mathrm{Pos}(\mathsf{H}) \subseteq \mathrm{Herm}(\mathsf{H}) \subseteq \mathcal{L}(\mathsf{H}) \quad \text{and} \quad \mathcal{U}(\mathsf{H}) \subseteq \mathcal{L}(\mathsf{H}). \tag{2.3}$$

Channels. We will be interested in a class of linear maps taking linear operators to linear operators, namely *channels*. These are completely positive and trace-preserving linear maps $\Phi : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathsf{H}')$ for Hilbert spaces H and H'. By *completely positive*, we mean that for

[10] In some edge cases not explicitly used in this work, this distinction can avoid some notational collisions.

any Hilbert space Z the map $\Phi \otimes \mathrm{Id}_{\mathsf{Z}}$ — where Id_{Z} is the identity on $\mathcal{L}(\mathsf{Z})$ — maps positive operators to positive operators, *i.e.*: $(\Phi \otimes \mathrm{Id}_{\mathsf{Z}})(\mathrm{Pos}(\mathsf{H} \otimes \mathsf{Z})) \subseteq \mathrm{Pos}(\mathsf{H}' \otimes \mathsf{Z})$. By *trace preserving*, we mean that $\mathrm{Tr}_{\mathsf{H}'}(\Phi(L)) = \mathrm{Tr}_{\mathsf{H}}(L)$ for all $L \in \mathcal{L}(\mathsf{H})$, *i.e.*: $\mathrm{Tr}_{\mathsf{H}'} \circ \Phi = \mathrm{Tr}_{\mathsf{H}}$. The set of all channels with domain $\mathcal{L}(\mathsf{H})$ and codomain $\mathcal{L}(\mathsf{H}')$ is denoted $\mathrm{Ch}(\mathsf{H}, \mathsf{H}')$. Note that channels map density operators to density operators. Channels are also called *CPTP maps*.

The trace $\mathrm{Tr}_{\mathsf{H}} : \mathcal{L}(\mathsf{H}) \to \mathbb{C}$ and conjugation $L \mapsto VLV^\dagger$ by any isometry $V \in \mathcal{L}(\mathsf{H}, \mathsf{H}')$ are channels, as well as any composition or tensor product of two channels. These observations are sufficient to characterize and construct *all* channels [Wat18, Cor. 2.27].

Measurements and swap channels. Let $\mathcal{A}$ be an alphabet. A *measurement* on a Hilbert space H with outcomes in $\mathcal{A}$ is a map $\mu : \mathcal{A} \to \mathrm{Pos}(\mathsf{H})$ such that $\sum_{a \in \mathcal{A}} \mu(a) = I_{\mathsf{H}}$. This yields a channel $\mathrm{Ms}(\mu) : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathbb{C}^{\mathcal{A}})$ defined by $L \mapsto \sum_{a \in \mathcal{A}} \mathrm{Tr}(\mu(A)L) \cdot |a\rangle\langle a|$.[11]

For any alphabet $\mathcal{X}$, we define the channel $\Delta_{\mathcal{X}} : \mathcal{L}(\mathbb{C}^{\mathcal{X}}) \to \mathcal{L}(\mathbb{C}^{\mathcal{X}})$ by

$$\rho \mapsto \sum_{x \in \mathcal{X}} |x\rangle\langle x| \, \rho \, |x\rangle\langle x| = \sum_{x \in \mathcal{X}} \langle x| \, \rho \, |x\rangle \cdot |x\rangle\langle x| = \sum_{x \in \mathcal{X}} \mathrm{Tr}(|x\rangle\langle x| \, \rho) \cdot |x\rangle\langle x| \tag{2.4}$$

often called the *completely dephasing channel*. It coincides with a measurement in the computational basis, *i.e.*: $\Delta_{\mathcal{X}} = \mathrm{Ms}(\mu_{\text{c.b.}})$ where $\mu_{\text{c.b.}} : \mathcal{X} \to \mathrm{Pos}(\mathsf{X})$ is defined by $x \mapsto |x\rangle\langle x|$.

Finally, we define "swap" channels which reorder the Hilbert spaces of their domains. Let $\mathsf{H}_1, \ldots \mathsf{H}_n$ be a sequence of n Hilbert spaces and let $\pi : \{1, \ldots n\} \to \{1, \ldots n\}$ be a permutation. Then, $\mathrm{Swap}^{\pi}_{\mathsf{H}_1, \ldots, \mathsf{H}_n} : \mathcal{L}(\mathsf{H}_1 \otimes \ldots \otimes \mathsf{H}_n) \to \mathcal{L}(\mathsf{H}_{\pi^{-1}(1)} \otimes \ldots \otimes \mathsf{H}_{\pi^{-1}(n)})$ is the channel which maps any simple tensor $|h_1\rangle\langle h_1| \otimes \ldots \otimes |h_n\rangle\langle h_n|$ to $\left|h_{\pi^{-1}(1)}\right\rangle\!\left\langle h_{\pi^{-1}(1)}\right| \otimes \ldots \left|h_{\pi^{-1}(n)}\right\rangle\!\left\langle h_{\pi^{-1}(n)}\right|$ and extends linearly to all other elements. In practice, we will denote the permutation π with the "one-line" notation. For example, $\mathrm{Swap}^{(2,3,1)}_{\mathsf{H},\mathsf{H}',\mathsf{H}''}$ maps $\mathcal{L}(\mathsf{H} \otimes \mathsf{H}' \otimes \mathsf{H}'')$ to $\mathcal{L}(\mathsf{H}'' \otimes \mathsf{H} \otimes \mathsf{H}')$.

Quantum mechanics. How is this prior mathematical machinery relevant to quantum mechanics? Quantum mechanics asserts that every quantum system is associated to a Hilbert space of non-zero dimension called its *state space*. At any given moment, the state of a quantum system whose state space is H is described by a density operator $\rho \in \mathcal{D}(\mathsf{H})$ on this space. Any action on a quantum system is described by a channel Φ and the state of the system after the action is given by $\Phi(\rho)$ if the state prior to the action was ρ. Finally, the joint state space of two quantum systems when the first system has state space H and the second H' is the tensor product $\mathsf{H} \otimes \mathsf{H}'$.

[11] From our prior characterization, the fact that this is a channel follows by composing conjugation by the isometry $\sum_{a \in \mathcal{A}} \sqrt{\mu(a)} \otimes |a\rangle \in \mathcal{L}(\mathsf{H}, \mathsf{H} \otimes \mathbb{C}^{\mathcal{A}})$ with the channel $(\mathrm{Tr}_{\mathsf{H}} \otimes \mathrm{Id}_{\mathbb{C}^{\mathcal{A}}}) : \mathcal{L}(\mathsf{H} \otimes \mathbb{C}^{\mathcal{A}}) \to \mathcal{L}(\mathbb{C}^{\mathcal{A}})$.

A measurement $\mu : \mathcal{A} \to \mathrm{Pos}(\mathsf{H})$, and the channel $\mathrm{Ms}(\mu)$ it yields, models an observation of a quantum system which provides classical information. More specifically, this measurement is applied to a state $\rho \in \mathcal{D}(\mathbb{C}^{\mathcal{A}})$, the outcome $a \in \mathcal{A}$ is obtained with probability $\mathrm{Tr}(\mu(a)\rho)$, destroying the state ρ and leaving a new system in the state $|a\rangle\langle a|$ to record this outcome. In particular, the cyclicity of the trace implies that measuring a state $\rho \in \mathcal{D}(\mathbb{C}^{\mathcal{A}})$ in the computational basis yields outcome $a \in \mathcal{A}$ with probability $\mathrm{Tr}(|a\rangle\langle a| \rho) = \langle a| \rho |a\rangle$.

For more details, the reader is directed to the textbooks of Nielsen and Chuang [NC10] and Watrous [Wat18] and the references therein.

Markov's inequality. We recall Markov's inequality (*e.g.*: [Bil95]). This is a concentration inequality which gives an upper bound on the probability that a non-negative random variable exceeds a given threshold in terms of the expectation of this random variable.

Theorem 2.1. Let X be a random variable distributed on the non-negative reals $\mathbb{R}_0^+$. Then, for any strictly positive $\alpha \in \mathbb{R}^+$, we have that

$$\Pr[X \geq \alpha] \leq \frac{\mathbb{E}(X)}{\alpha} \quad \text{and} \quad \Pr[X < \alpha] \geq 1 - \frac{\mathbb{E}(X)}{\alpha}. \tag{2.5}$$

The following can be seen as an analogue to Markov's inequality if we can also assume that X almost surely falls in an interval $[0, \beta]$.

Lemma 2.2. Let X be a random variable distributed on $\mathbb{R}_0^+$ and let $\beta \in \mathbb{R}_0^+$ be such that $\Pr[0 \leq X \leq \beta] = 1$. Then, for any strictly positive $\alpha \in \mathbb{R}^+$ satisfying $\alpha < \beta$ we have that

$$\Pr[X > \alpha] \geq \frac{\mathbb{E}(X) - \alpha}{\beta - \alpha} \quad \text{and} \quad \Pr[X \leq \alpha] \leq 1 - \frac{\mathbb{E}(X) - \alpha}{\beta - \alpha}. \tag{2.6}$$

Proof: Since $\Pr[X > \beta] = 0$, we have that

$$\begin{aligned} \mathbb{E}(X) &\leq \beta \cdot \Pr[X > \alpha] + \alpha \cdot \Pr[X \leq \alpha] \\ &= \beta \cdot \Pr[X > \alpha] + \alpha \cdot (1 - \Pr[X > \alpha]) \\ &= \alpha + \Pr[X > \alpha] \cdot (\beta - \alpha) \end{aligned} \tag{2.7}$$

which yields the first equation. Noting that $\Pr[X \leq \alpha] = 1 - \Pr[X < \alpha]$ then yields the second. ∎

Final miscellaneous conventions. We freely identify $\mathbb{C}$ with $\mathcal{L}(\mathbb{C})$ in this work, as well as H, $\mathbb{C} \otimes \mathsf{H}$, and $\mathsf{H} \otimes \mathbb{C}$ for any Hilbert space H.

To lighten notation when there is no risk of confusion, we may identify an element of an alphabet $a \in \mathcal{A}$ with the corresponding density operator $|a\rangle\langle a| \in \mathcal{D}(\mathbb{C}^{\mathcal{A}})$. For example, if $\Phi : \mathcal{L}(\mathbb{C}^{\mathcal{A}}) \to \mathsf{H}$ is a channel, we may write $\Phi(a - a')$ instead of $\Phi(|a\rangle\langle a| - |a'\rangle\langle a'|)$.

We emphasize that an empty quantum system has $\mathbb{C}$ as its state space and that there is a unique density operator on this space, namely 1. By identifying $\mathbb{C}$ with $\mathbb{C}^{\{\varepsilon\}}$ — as previously discussed — and using the previous convention, we can also denote the state of an empty quantum system as ε.

When convenient, we may treat a density operator $\rho \in \mathcal{D}(\mathbb{C}^{\mathcal{A}})$ as a random variable on the set $\mathcal{A}$ where the probability of sampling $a \in \mathcal{A}$ from ρ is given by $\mathrm{Tr}(|a\rangle\langle a|\,\rho)$. In other words, $a \leftarrow \rho$ means that a is sampled by measuring ρ in the computational basis. In particular, if $A : \mathbb{C} \to \mathcal{L}(\mathbb{C}^{\mathcal{A}})$ is a channel, then $a \leftarrow A(\varepsilon)$ means that a is sampled from the set $\mathcal{A}$ by measuring in the computational basis the output of A on the empty input.

On occasion and to add clarity, we will add a subscript to a ket, bra, or linear operator from a Hilbert space to itself to make explicit the Hilbert space related to this object. For example, we may write $|\psi\rangle_{\mathsf{F}} \in \mathsf{F}$ instead of simply $|\psi\rangle \in \mathsf{F}$. In this case, $\langle\psi|_{\mathsf{F}}$ is the associated dual. Similarly, we may write L_{H} if $L \in \mathcal{L}(\mathsf{H})$. Context will make clear when a subscript denotes a Hilbert space in this manner, or when it is used to convey other information.

Finally, we occasionally refer to a quantum system as a *register*, and to any quantum system with states space $\mathbb{C}^{\{0,1\}}$ as a *qubit*, the quantum analogue of a bit.

2.2.2 Trace Distance

This work will make extensive use of the trace distance between linear operators and so we review here its definition, some related basic facts, as well as a few simple — but less well known — technical properties.

First, recall that the trace distance between two linear operators $A, B \in \mathcal{L}(\mathsf{H})$ is defined to be half the Schatten 1-norm of their difference:

$$\frac{1}{2}\|A - B\|_1 = \frac{1}{2}\,\mathrm{Tr}\,\sqrt{(A-B)^\dagger(A-B)}. \tag{2.8}$$

It is a metric on the space of linear operators. In particular it satisfies the triangle inequality. Moreover, for pairs of density operators, its value is between zero and one.

The importance of the trace distance in quantum information theory comes from the fact that it is an appropriate "operational" measure of the distance between quantum states. Indeed, suppose a party is presented uniformly at random with one of two quantum systems whose states are described by two density operators $\rho_0, \rho_1 \in \mathcal{D}(\mathsf{H})$. Then, it can be shown that the party can correctly determine with which system it was presented with a probability

of at most $\frac{1}{2}\left(1+\frac{1}{2}\|\rho_0-\rho_1\|_1\right)$ [FdG99, Prop. 1][12], a value which ranges from one half to one as the trace distance ranges from zero to one. Standard proofs of this fact can be found in many textbooks (e.g.: [NC10, Sec. 9.2.1]) and we will offer our own, for a slightly more general setting, shortly in Lemma 2.3.

Before, however, we recall the following two other facts concerning the trace distance:

- The trace distance is contractive under the action of channels. More formally, we mean that for any linear operators $A, B \in \mathcal{L}(\mathsf{H})$ and any channel Φ whose domain is $\mathcal{L}(\mathsf{H})$, it holds that $\frac{1}{2}\|\Phi(A)-\Phi(B)\|_1 \leq \frac{1}{2}\|A-B\|_1$. *Operationally, this implies that applying channels cannot increase the distinguishability of two quantum systems.*
- The trace distance is invariant under conjugation by isometries. More formally, we mean that for any isometry $V \in \mathcal{U}(\mathsf{H},\mathsf{H}')$ and any linear operators $A, B \in \mathcal{L}(\mathsf{H})$ it holds that $\frac{1}{2}\left\|VAV^\dagger - VBV^\dagger\right\|_1 = \frac{1}{2}\|A-B\|_1$.

As mentioned, the trace distance between two density operators is essentially a measure of how well quantum systems whose states are represented by these density operators can be distinguished. For our work, it will be useful to frame this via bounds concerning channels which attempt to map one state to $\mathtt{0}$ and the other to $\mathtt{1}$. We formalize this idea below for a class of linear operators which extends beyond only density operators, including, in particular, any pair of subnormalized density operators. We give an explicit proof of this lemma as it is a slightly unorthodox way to state this result and most standard references only treat the more restrictive case considering pairs of density operators.

Lemma 2.3. Let $A, B \in \mathcal{L}(\mathsf{H})$ be two linear operators such that $A-B$ is Hermitian. Then, for all channels $\Phi : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathbb{C}^{\{\mathtt{0},\mathtt{1}\}})$ we have that

$$|\langle \mathtt{0}|\,\Phi(A-B)\,|\mathtt{0}\rangle| + |\langle \mathtt{1}|\,\Phi(A-B)\,|\mathtt{1}\rangle| \leq \|A-B\|_1 \tag{2.9}$$

which implies that

$$\langle \mathtt{0}|\,\Phi(A)\,|\mathtt{0}\rangle + \langle \mathtt{1}|\,\Phi(B)\,|\mathtt{1}\rangle \leq \frac{\mathrm{Tr}(A)+\mathrm{Tr}(B)}{2} + \frac{1}{2}\|A-B\|_1. \tag{2.10}$$

Moreover, there exists a channel, which depends on A and B, saturating these inequalities.

We make two remarks on this lemma before giving its proof.

[12] Note that Fuchs and van de Graaf consider in their proposition the probability of error when trying to distinguish between ρ_0 and ρ_1, while we consider the probability of success.

1. If $\rho_0, \rho_1 \in \mathcal{D}(\mathsf{H})$ are density operators, implying that $\mathrm{Tr}(\rho_0) = \mathrm{Tr}(\rho_1) = 1$, then

$$\frac{\langle \mathtt{0}| \Phi(\rho_0) |\mathtt{0}\rangle + \langle \mathtt{1}| \Phi(\rho_1) |\mathtt{1}\rangle}{2} \leq \frac{1}{2}\left(1 + \frac{1}{2}\|\rho - \sigma\|_1\right). \tag{2.11}$$

   The left-hand side of this inequality is the success probability of a party attempting to distinguish, with the channel Φ, between the state ρ_0 (by mapping it to $\mathtt{0}$) and the state ρ_1 (by mapping it to $\mathtt{1}$) if presented with one of these states sampled uniformly randomly. The right-hand side is precisely the previously advertised upper bound on the success probability of such an attempt.

2. If $\mathrm{Tr}(A) = \mathrm{Tr}(B)$, then

$$|\langle \mathtt{0}| \Phi(A - B) |\mathtt{0}\rangle| = |\mathrm{Tr}(A - B) - \langle \mathtt{1}| \Phi(A - B) |\mathtt{1}\rangle| = |\langle \mathtt{1}| \Phi(A - B) |\mathtt{1}\rangle| \tag{2.12}$$

   as Φ is trace preserving. This implies that $|\langle b| \Phi(A - B) |b\rangle| \leq \frac{1}{2}\|A - B\|_1$ for both values of $b \in \{\mathtt{0}, \mathtt{1}\}$. However, in the general case, such as if A and B are two subnormalized density operators with distinct traces, the right-hand side of this inequality may be too small: it may only hold that $|\langle b| \Phi(A - B) |b\rangle| \leq \|A - B\|_1$.[13] Cases like these — which we will encounter — are often omitted from standard treatments.

Proof: We first show the two inequalities and then demonstrate the existence of a saturating channel.

Define the channel $\tilde{\Phi} = \Delta_{\{\mathtt{0},\mathtt{1}\}} \circ \Phi$. Then,

$$\tilde{\Phi}(A) - \tilde{\Phi}(B) = \langle \mathtt{0}| \Phi(A - B) |\mathtt{0}\rangle \cdot |\mathtt{0}\rangle\langle \mathtt{0}| + \langle \mathtt{1}| \Phi(A - B) |\mathtt{1}\rangle \cdot |\mathtt{1}\rangle\langle \mathtt{1}| \tag{2.13}$$

from which it follows, by direct computation of the relevant Schatten 1-norm, that

$$\left\|\tilde{\Phi}(A) - \tilde{\Phi}(B)\right\|_1 = |\langle \mathtt{0}| \Phi(A - B) |\mathtt{0}\rangle| + |\langle \mathtt{1}| \Phi(A - B) |\mathtt{1}\rangle|. \tag{2.14}$$

As the trace distance is contractive under the action of channels, we have that

$$|\langle \mathtt{0}| \Phi(A - B) |\mathtt{0}\rangle| + |\langle \mathtt{1}| \Phi(A - B) |\mathtt{1}\rangle| \leq \|A - B\|_1 \tag{2.15}$$

[13] *E.g.*: Taking $\Phi = \mathrm{Id}_{\mathbb{C}^{\{\mathtt{0},\mathtt{1}\}}}$, $A = |\mathtt{0}\rangle\langle \mathtt{0}|$, and $B = \frac{1}{2}A$ yields $|\langle \mathtt{0}| \Phi(A - B) |\mathtt{0}\rangle| = \frac{1}{2}$ and $\|A - B\|_1 = \frac{1}{2}$.

which yields the first inequality. To obtain the second inequality, we note that

$$\begin{aligned} 2\langle 0|\Phi(A)|0\rangle - \mathrm{Tr}(A) &= 2\langle 0|\Phi(A)|0\rangle - \mathrm{Tr}\circ\Phi(A) \\ &= 2\langle 0|\Phi(A)|0\rangle - \langle 0|\Phi(A)|0\rangle - \langle 1|\Phi(A)|1\rangle \\ &= \langle 0|\Phi(A)|0\rangle - \langle 1|\Phi(A)|1\rangle \end{aligned} \tag{2.16}$$

and, similarly, that $2\langle 1|\Phi(B)|1\rangle - \mathrm{Tr}(B) = \langle 1|\Phi(B)|1\rangle - \langle 0|\Phi(B)|0\rangle$. Thus,

$$\begin{aligned} 2\langle 0|\Phi(A)|0\rangle + 2\langle 1|\Phi(B)|1\rangle - \mathrm{Tr}(A+B) &= \langle 0|\Phi(A-B)|0\rangle + \langle 1|\Phi(B-A)|1\rangle \\ &\leq |\langle 0|\Phi(A-B)|0\rangle| + |\langle 1|\Phi(A-B)|1\rangle| \\ &\leq \|A-B\|_1 \end{aligned} \tag{2.17}$$

from which the desired result follows after simple algebraic manipulations.

We now show the existence of a channel $\Psi : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}})$ which saturates these bounds. This part of the proof follows closely the corresponding proof given by Nielsen and Chuang [NC10, Sec. 9.1.2]. Note that since $A - B$ is Hermitian by assumption, we may write

$$A - B = \sum_{j\in\mathcal{J}} \lambda_j \Pi_j = \sum_{\substack{j\in\mathcal{J} \\ \lambda_j \geq 0}} \lambda_j \Pi_j + \sum_{\substack{j\in\mathcal{J} \\ \lambda_j < 0}} \lambda_j \Pi_j \tag{2.18}$$

for some indexing set $\mathcal{J}$ and where each λ_j is a real number and the Π_j's are rank one projectors on H which together sum to the identity I_{H}.[14] At this point, we recall an alternative characterization of the Schatten 1-norm of an operator, namely that it is the sum of its singular values [Wat18, Sec. 1.1]. Thus, $\|A-B\|_1 = \sum_{j\in\mathcal{J}} |\lambda_j|$. Now, let

$$P = \sum_{\substack{j\in\mathcal{J} \\ \lambda_j \geq 0}} \Pi_j \quad \text{and} \quad Q = \sum_{\substack{j\in\mathcal{J} \\ \lambda_j < 0}} \Pi_j \tag{2.19}$$

and note that both are positive semidefinite operators which sum to the identity. Hence, we may define the measurement $\mu : \{0,1\} \to \mathrm{Pos}(\mathsf{H})$ by $\mu(0) = P$ and $\mu(1) = Q$ and consider the channel $\Psi = \mathrm{Ms}(\mu)$, *i.e.*: $\rho \mapsto \mathrm{Tr}(P\rho)\cdot|0\rangle\langle 0| + \mathrm{Tr}(Q\rho)\cdot|1\rangle\langle 1|$. Then,

$$\langle 0|\Psi(A-B)|0\rangle = \mathrm{Tr}\left(P(A-B)\right) = \sum_{\substack{j\in\mathcal{J} \\ \lambda_j \geq 0}} \lambda_j = \sum_{\substack{j\in\mathcal{J} \\ \lambda_j \geq 0}} |\lambda_j| \tag{2.20}$$

[14] This is the spectral decomposition of $A - B$ [Wat18, Th. 1.3]. The fact that the λ_j's are strictly real, and not complex, follows from the fact that $A - B$ is Hermitian.

and

$$\langle 1|\,\Psi(B-A)\,|1\rangle = \mathrm{Tr}\,(Q(B-A)) = -\,\mathrm{Tr}\,(Q(A-B)) = -\sum_{\substack{j\in\mathcal{J}\\ \lambda_j<0}}\lambda_j = \sum_{\substack{j\in\mathcal{J}\\ \lambda_j<0}}|\lambda_j| \tag{2.21}$$

from which it follows that

$$\begin{aligned}|\langle 0|\,\Psi(A-B)\,|0\rangle| + |\langle 1|\,\Psi(A-B)\,|1\rangle| &= \langle 0|\,\Psi(A-B)\,|0\rangle + \langle 1|\,\Psi(B-A)\,|1\rangle \\ &= \sum_{j\in\mathcal{J}}|\lambda_j| \\ &= \|A-B\|_1,\end{aligned} \tag{2.22}$$

which yields the desired result. ∎

We finish this exposition on the trace distance with three technical lemmas giving values or bounds of this norm in particular cases.

First, we recall an explicit computation of the trace distance for certain simple cases.

Lemma 2.4 ([Wat18, Eq. 1.184])**.** Let $|\psi\rangle, |\phi\rangle$ be arbitrary vectors in a Hilbert space H. Then,

$$\frac{1}{2}\||\psi\rangle\langle\psi| - |\phi\rangle\langle\phi|\|_1 = \sqrt{\left(\frac{\langle\psi|\psi\rangle + \langle\phi|\phi\rangle}{2}\right)^2 - |\langle\psi|\phi\rangle|^2}. \tag{2.23}$$

Second, the following lemma pertains to the trace distance between multiple copies of two density operators. The proof can be found in the full version of [BCG+02]. Conceptually, this can be understood as giving a lower bound on the ability to distinguish between n copies of a state ρ and n copies of a state σ by attempting n times to distinguish between a single copy of ρ and a single copy of σ and taking a majority vote of the result.

Lemma 2.5 ([BCG+02])**.** Let $\rho, \sigma \in \mathcal{D}(\mathsf{H})$ be two density operators. Then, for all $t \in \mathbb{N}$, we have that

$$\frac{1}{2}\left\|\rho^{\otimes t} - \sigma^{\otimes t}\right\|_1 \geq 1 - 2\exp\left(-\frac{t}{2}\left(\frac{1}{2}\|\rho-\sigma\|_1\right)^2\right) \tag{2.24}$$

where $\rho^{\otimes t} = \rho\otimes\rho\otimes\cdots\otimes\rho$ is the tensor product of t copies of ρ, and similarly for $\sigma^{\otimes t}$.

And, third, we give a lemma concerning the trace distance between two linear operators with a particular tensor decomposition. For completion, we provide a proof of this lemma.

Lemma 2.6. Let $\mathcal{X}$ be an alphabet. For any linear operators $\{A_x\}_{x\in\mathcal{X}}, \{B_x\}_{x\in\mathcal{X}} \subseteq \mathcal{L}(\mathsf{H})$

it holds that

$$\left\| \sum_{x \in \mathcal{X}} x \otimes A_x - \sum_{x \in \mathcal{X}} x \otimes B_x \right\|_1 = \sum_{x \in \mathcal{X}} \|A_x - B_x\|_1. \tag{2.25}$$

Proof: First, factor the common x terms. Then, by definition, we have that

$$\begin{aligned}
\left\| \sum_{x \in \mathcal{X}} x \otimes (A_x - B_x) \right\|_1 &= \operatorname{Tr} \sqrt{\left(\sum_{x \in \mathcal{X}} x \otimes (A_x - B_x) \right)^\dagger \left(\sum_{x \in \mathcal{X}} x \otimes (A_x - B_x) \right)} \\
&= \operatorname{Tr} \sqrt{\sum_{x \in \mathcal{X}} x \otimes (A_x - B_x)^\dagger (A_x - B_x)} \\
&= \operatorname{Tr} \left(\sum_{x \in \mathcal{X}} x \otimes \sqrt{(A_x - B_x)^\dagger (A_x - B_x)} \right) \\
&= \sum_{x \in \mathcal{X}} \operatorname{Tr} \left(\sqrt{(A_x - B_x)^\dagger (A_x - B_x)} \right) \\
&= \sum_{x \in \mathcal{X}} \|A_x - B_x\|_1
\end{aligned} \tag{2.26}$$

where the second and third equalities follow from the fact that $xx' = |x\rangle\langle x|\,|x'\rangle\langle x'|$ is the zero operator if $x \neq x'$ and is $x = |x\rangle\langle x|$ if $x = x'$. The fourth equality follows from the linearity of the trace, the fact that $\operatorname{Tr}(Y \otimes Z) = \operatorname{Tr}(Y) \cdot \operatorname{Tr}(Z)$, and that $\operatorname{Tr}(x) = 1$. ∎

2.2.3 The Coherent Gentle Measurement

In general, a measurement is a destructive or at least perturbative operation on a quantum state. However, it is well known that when a particular measurement outcome is highly likely, a measurement may not disturb a state by much. Formalizations of this idea include *gentle measurements* [Win99], *coherent gentle measurements* [Wil17, Sec. 9.4], and the "*almost as good as new*" lemma [Aar16a].

Here, we give a formalization of this idea by constructing a mapping that takes any channel Φ to its "coherent gentle measurement" counterpart $\mathrm{CGM}(\Phi)$. This construction satisfies the following gentle measurement-like property: Suppose ρ was a state such that measuring $\Phi(\rho)$ in the computational basis yielded a particular outcome with very high probability. Then, $\mathrm{CGM}(\Phi)(\rho)$ yields a state very close to ρ tensored with the likely measurement outcome. In other words, $\mathrm{CGM}(\Phi)$ allows us to keep ρ *and* obtain the outcome of measuring $\Phi(\rho)$ in the computational basis, up to some well understood error.

Our formalization is a slight extension of an exercise in Wilde's textbook [Wil17, Ex. 9.4.2]. Our result is directly applicable to all channels instead of only measurements,

yields a well defined map instead of only providing a proof of existence, and is applicable to all positive semidefinite operators, not only density operators.

We will proceed in two steps. We first define the map $\Phi \mapsto \mathrm{CGM}(\Phi)$ and then we prove that it possesses the desired properties. First, however, we recall the following theorem which states the existence of a one-to-one correspondence between measurements and a certain class of channels.

Theorem 2.7 ([Wat18, Th. 2.37])**.** Let $\mathcal{A}$ be an alphabet and $\Phi : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathbb{C}^{\mathcal{A}})$ be a channel. Then, there exists a unique measurement $\mu_\Phi : \mathcal{A} \to \mathrm{Pos}(\mathsf{H})$ satisfying

$$\Delta_{\mathcal{A}} \circ \Phi = \mathrm{Ms}(\mu_\Phi). \tag{2.27}$$

Definition 2.8. Let X be a Hilbert space and $\mathcal{Y}$ be an alphabet. The *coherent gentle measurement* map $\mathrm{CGM} : \mathrm{Ch}(\mathsf{X}, \mathbb{C}^{\mathcal{Y}}) \to \mathrm{Ch}(\mathsf{X}, \mathsf{X} \otimes \mathbb{C}^{\mathcal{Y}})$ is defined by

$$\Phi \mapsto \left[\rho \mapsto \left(\sum_{y \in \mathcal{Y}} \sqrt{\mu_\Phi(y)} \otimes |y\rangle \right) \rho \left(\sum_{y \in \mathcal{Y}} \sqrt{\mu_\Phi(y)}^\dagger \otimes \langle y| \right) \right] \tag{2.28}$$

where $\mu_\Phi : \mathcal{Y} \to \mathrm{Pos}(\mathsf{X})$ is the unique measurement satisfying $\Delta_{\mathcal{Y}} \circ \Phi = \mathrm{Ms}(\mu_\Phi)$.

Remark 2.9. Definition 2.8 asserts that CGM maps channels to *channels*. This needs to be shown. To show that $\mathrm{CGM}(\Phi)$ is a channel, it suffices to show that $V = \sum_{y \in \mathcal{Y}} \sqrt{\mu(y)} \otimes |y\rangle$ is an isometry in $\mathcal{U}(\mathsf{X}, \mathsf{X} \otimes \mathsf{Y})$. This follows from the fact that

$$V^\dagger V = \sum_{y, y' \in \mathcal{Y}} \left(\sqrt{\mu(y)}^\dagger \otimes \langle y| \right) \left(\sqrt{\mu(y')} \otimes |y'\rangle \right) = \sum_{y \in \mathcal{Y}} \mu(y) = I_{\mathsf{X}} \tag{2.29}$$

where we recall that $\sqrt{\mu(y)}^\dagger = \sqrt{\mu(y)}$ since $\sqrt{\mu(y)}$ is positive and hence self-adjoint. By the same token, we can replace the $\sqrt{\mu(y)}^\dagger$ terms in Equation 2.28 with $\sqrt{\mu(y)}$, which is what we will do in practice.

The following theorem gives two properties of the coherent gentle measurement map. The first bounds the trace distance between the real output $\mathrm{CGM}(\Phi)(\rho)$ and an "ideal" output $\rho \otimes |\tilde{y}\rangle\langle\tilde{y}|$ based on the probability that measuring $\Phi(\rho)$ in the computational basis would yield $\tilde{y}$ as outcome. The second states that measuring the Y register of either $\Phi(\rho)$ or $\mathrm{CGM}(\Phi)(\rho)$ in the computational basis yields the same results.

Theorem 2.10. Let X be a Hilbert space and $\mathcal{Y}$ be an alphabet. Then, the coherent gentle measurement map $\mathrm{CGM} : \mathrm{Ch}(\mathsf{X}, \mathbb{C}^{\mathcal{Y}}) \to \mathrm{Ch}(\mathsf{X}, \mathsf{X} \otimes \mathbb{C}^{\mathcal{Y}})$ given in Definition 2.8 satisfies the following two properties.

1. For any channel $\Phi : \mathcal{L}(\mathsf{X}) \to \mathcal{L}(\mathsf{Y})$, any positive semidefinite operator $\rho \in \mathrm{Pos}(\mathsf{X})$, and any $y \in \mathcal{Y}$, we have that

$$\frac{1}{2}\|\mathrm{CGM}(\Phi)(\rho) - \rho \otimes |y\rangle\langle y|\|_1 \leq \sqrt{\mathrm{Tr}(\rho)^2 - (\langle y| \Phi(\rho) |y\rangle)^2}. \tag{2.30}$$

2. For any channel $\Phi : \mathcal{L}(\mathsf{X}) \to \mathcal{L}(\mathsf{Y})$, we have that

$$(\mathrm{Tr}_{\mathsf{X}} \otimes \Delta_{\mathcal{Y}}) \circ \mathrm{CGM}(\Phi) = \Delta_{\mathcal{Y}} \circ \Phi. \tag{2.31}$$

Proof: We begin by showing the second point. Let $\mu_\Phi : \mathcal{Y} \to \mathrm{Pos}(\mathsf{X})$ be the unique measurement satisfying $\Delta_{\mathcal{Y}} \circ \Phi = \mathrm{Ms}(\mu_\Phi)$. Then, for any linear operator $\rho \in \mathcal{L}(\mathsf{X})$ we have that

$$\begin{aligned}
&(\mathrm{Tr}_{\mathsf{X}} \otimes \Delta_{\mathcal{Y}}) \circ \mathrm{CGM}(\Phi)(\rho) \\
&= (\mathrm{Tr}_{\mathsf{X}} \otimes \Delta_{\mathcal{Y}}) \left(\sum_{y,y' \in \mathcal{Y}} \sqrt{\mu_\Phi(y)} \rho \sqrt{\mu_\Phi(y')} \otimes |y\rangle\langle y'| \right) \\
&= (\mathrm{Tr}_{\mathsf{X}} \otimes \mathrm{Id}_{\mathsf{Y}}) \left(\sum_{\tilde{y} \in \mathcal{Y}} \sum_{y,y' \in \mathcal{Y}} \sqrt{\mu_\Phi(y)} \rho \sqrt{\mu_\Phi(y')} \otimes \langle \tilde{y}|y\rangle \langle y'|\tilde{y}\rangle \cdot |\tilde{y}\rangle\langle\tilde{y}| \right) \\
&= (\mathrm{Tr}_{\mathsf{X}} \otimes \mathrm{Id}_{\mathsf{Y}}) \left(\sum_{\tilde{y} \in \mathcal{Y}} \sqrt{\mu_\Phi(\tilde{y})} \rho \sqrt{\mu_\Phi(\tilde{y})} \otimes |\tilde{y}\rangle\langle\tilde{y}| \right) \\
&= \sum_{\tilde{y} \in \mathcal{Y}} \mathrm{Tr}\left(\mu_\Phi(\tilde{y})\rho\right) \cdot |\tilde{y}\rangle\langle\tilde{y}| \\
&= \mathrm{Ms}(\mu_\Phi)(\rho) \\
&= \Delta_{\mathcal{Y}} \circ \Phi(\rho)
\end{aligned} \tag{2.32}$$

which yields the desired result.

We now show the first point. Once again, let $\mu_\Phi : \mathcal{Y} \to \mathrm{Pos}(\mathsf{X})$ be the unique measurement satisfying $\Delta_{\mathcal{Y}} \circ \Phi = \mathrm{Ms}(\mu_\Phi)$ and let $V = \sum_{\tilde{y} \in \mathcal{Y}} \sqrt{\mu_\Phi(\tilde{y})} \otimes |\tilde{y}\rangle$. As $\rho \in \mathrm{Pos}(\mathsf{X})$ is a positive operator, there is a vector $|\psi\rangle \in \mathsf{X} \otimes \mathsf{X}$ such that $\rho = (\mathrm{Tr}_{\mathsf{X}} \otimes \mathrm{Id}_{\mathsf{X}})(|\psi\rangle\langle\psi|)$.[15] Note that $|\psi\rangle$ need not have norm one as we are not assuming that ρ is a density operator, only that it is positive semidefinite. As the trace distance is contractive under the action

[15] Such a vector is known as a *purification* of ρ and is guaranteed to exist [Wat18, Th. 2.20].

of channels, we have that

$$\begin{aligned}
&\frac{1}{2}\|\mathrm{CGM}(\Phi)(\rho) - \rho \otimes |y\rangle\langle y|\|_1 \\
&= \frac{1}{2}\|\left(\mathrm{Tr}_{\mathsf{X}} \otimes \mathrm{CGM}(\Phi)\right)(|\psi\rangle\langle\psi|) - \left(\mathrm{Tr}_{\mathsf{X}} \otimes \mathrm{Id}_{\mathsf{X}\otimes\mathsf{Y}}\right)(|\psi\rangle\langle\psi| \otimes |y\rangle\langle y|)\|_1 \\
&\leq \frac{1}{2}\|\left(\mathrm{Id}_{\mathsf{X}} \otimes \mathrm{CGM}(\Phi)\right)(|\psi\rangle\langle\psi|) - (|\psi\rangle\langle\psi| \otimes |y\rangle\langle y|)\|_1. \\
&= \frac{1}{2}\left\|(I_{\mathsf{X}} \otimes V)\,|\psi\rangle\langle\psi|\,(I_{\mathsf{X}} \otimes V^\dagger) - |\psi\rangle\langle\psi| \otimes |y\rangle\langle y|\right\|_1.
\end{aligned} \tag{2.33}$$

We can then use Lemma 2.4 to compute this final trace distance. As V is an isometry, we have that

$$\mathrm{Tr}\left((I_{\mathsf{X}} \otimes V)\,|\psi\rangle\langle\psi|\,(I_{\mathsf{X}} \otimes V^\dagger)\right) = \langle\psi|\,(I_{\mathsf{X}} \otimes V^\dagger V)\,|\psi\rangle = \langle\psi|\psi\rangle\,. \tag{2.34}$$

We also have that

$$\mathrm{Tr}(|\psi\rangle\langle\psi| \otimes |y\rangle\langle y|) = \langle\psi|\psi\rangle \cdot \langle y|y\rangle = \langle\psi|\psi\rangle\,. \tag{2.35}$$

Further, both these quantities are also equal to $\mathrm{Tr}(\rho)$. Thus, by Lemma 2.4, it follows that

$$\begin{aligned}
\frac{1}{2}\left\|(I_{\mathsf{X}} \otimes V)\,|\psi\rangle\langle\psi|\,(I_{\mathsf{X}} \otimes V^\dagger) - |\psi\rangle\langle\psi| \otimes |y\rangle\langle y|\right\|_1 &= \sqrt{\mathrm{Tr}(\rho)^2 - |(\langle\psi| \otimes \langle y|)\,(I_{\mathsf{X}} \otimes V)\,|\psi\rangle|^2} \\
&= \sqrt{\mathrm{Tr}(\rho)^2 - \left|\langle\psi|\,(I_{\mathsf{X}} \otimes \sqrt{\mu(y)})\,|\psi\rangle\right|^2}
\end{aligned} \tag{2.36}$$

Now, note that

$$\langle\psi|\,(I_{\mathsf{X}} \otimes \sqrt{\mu(y)})\,|\psi\rangle = \mathrm{Tr}_{\mathsf{X}\otimes\mathsf{X}}\left(\left(I_{\mathsf{X}} \otimes \sqrt{\mu(y)}\right)|\psi\rangle\langle\psi|\right) = \mathrm{Tr}_{\mathsf{X}}\left(\sqrt{\mu(y)}\rho\right) \tag{2.37}$$

and that this quantity must be real and non-negative by virtue of $I_{\mathsf{X}} \otimes \sqrt{\mu(y)}$ being a positive semidefinite operator. We now claim that $\mathrm{Tr}_{\mathsf{X}}(\sqrt{\mu(y)}\rho) \geq \mathrm{Tr}_{\mathsf{X}}(\mu(y)\rho)$. This is obtained by considering the spectral decomposition $\mu(\tilde{y}) = \sum_{j\in[\dim \mathsf{X}]} \lambda_j \Pi_j$ where each Π_j is a projector on orthogonal subspaces and noting that that every λ_j is real, non-negative, and no greater than one. The first two properties of these λ_j's follow from the fact that $\mu(y) \in \mathrm{Pos}(\mathsf{X})$ and the last from the fact that $\sum_{\tilde{y}\in\mathcal{Y}} \mu(\tilde{y}) = I_{\mathsf{X}}$. Thus, we find that $\sqrt{\mu(y)} = \sum_{j\in[\dim \mathsf{X}]} \sqrt{\lambda_j}\Pi_j$ and so

$$\mathrm{Tr}_{\mathsf{X}}\left(\sqrt{\mu(y)}\rho\right) = \sum_{j\in[\dim \mathsf{X}]} \sqrt{\lambda_j}\,\mathrm{Tr}\left(\Pi_j\rho\right) \geq \sum_{j\in[\dim \mathsf{X}]} \lambda_j\,\mathrm{Tr}\left(\Pi_j\rho\right) = \mathrm{Tr}\left(\mu(y)\rho\right). \tag{2.38}$$

Finally, $\mathrm{Tr}(\mu(y)\rho) = \langle y|\,\mathrm{Ms}(\mu)(\rho)\,|y\rangle = \langle y|\,\Delta_{\mathcal{Y}} \circ \Phi(\rho)\,|y\rangle = \langle y|\,\Phi(\rho)\,|y\rangle$. Collecting the above

facts immediately yields that

$$\frac{1}{2}\|\mathrm{CGM}(\Phi)(\rho) - \rho \otimes |y\rangle\langle y|\|_1 \leq \sqrt{\mathrm{Tr}(\rho)^2 - \left(\langle y| \Phi(\rho) |y\rangle\right)^2} \tag{2.39}$$

which is the desired result. ∎

2.3 Defining Tamper Evidence

The goal of this section is twofold. First, we formally define the main objects which we will study in this chapter, namely *quantum encryptions of classical messages* (QECM) schemes and their *augmented* variants, AQECM schemes. Second, we formalize the notion of *tamper evidence*, a property which can be possessed by augmented quantum encryption of classical messages schemes.

Recall that QECMs and AQECMs are keyed encoding schemes for classical messages which produce quantum states. While such schemes were implicit in the literature since at least the early 2000s, such as in Gottesman's work [Got03], this terminology and precise formalization was first introduced in our prior work on uncloneable encryption [BL20] for QECMs, and subsequently expanded by Broadbent and Islam to AQECMs in the course of their work on certfied deletion [BI20].

2.3.1 Quantum Encryption of Classical Messages Schemes

A quantum encryption of classical messages (QECM) scheme is a triplet of channels (K, E, D) called, respectively, the *key generation*, *encoding*, and *decoding* channels. These channels are required to satisfy certain constraints on their domains and codomains to ensure that they are compatible with one another. Implicitly, these requirements on the domains and codomains also define two alphabets $\mathcal{K}$ and $\mathcal{M}$ which represent, respectively, the set of classical keys and messages for this scheme.

Definition 2.11. A *quantum encryption of classical messages* (QECM) scheme is a triplet of channels (K, E, D) of the form

$$K : \mathcal{L}(\mathbb{C}) \to \mathcal{L}(\mathsf{K}), \quad E : \mathcal{L}(\mathsf{K} \otimes \mathsf{M}) \to \mathcal{L}(\mathsf{C}), \quad \text{and} \quad D : \mathcal{L}(\mathsf{K} \otimes \mathsf{C}) \to \mathcal{L}(\mathsf{M}) \tag{2.40}$$

for Hilbert spaces K, M, and C where, $\mathsf{K} = \mathbb{C}^{\mathcal{K}}$ and $\mathsf{M} = \mathbb{C}^{\mathcal{M}}$ for alphabets $\mathcal{K}$ and $\mathcal{M}$. We call the elements of these alphabets, respectively, the *keys* and *messages* of the scheme.

To lighten notation, for every key $k \in \mathcal{K}$ we define the channels $E_k : \mathcal{L}(\mathsf{M}) \to \mathcal{L}(\mathsf{C})$ and $D_k : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{M})$ by $\rho \mapsto E(k \otimes \rho)$ and $\rho \mapsto D(k \otimes \rho)$, respectively.

From this point on, to ease notation, whenever we talk of a "QECM scheme (K, E, D) as given in Definition 2.11", we are assuming that the domains and codomains are defined and denoted exactly as in Definition 2.11.

We now define a quantitative notion of *correctness* for QECM schemes. In short, a QECM scheme is ϵ-correct if for every message $m \in \mathcal{M}$ the probability that it is correctly recovered by the decoding map D_k after being processed by the encoding map E_k is at least $1 - \epsilon$. Here, the probability is taken over the action of the maps E_k and D_k as well as the randomness of sampling the key $k \leftarrow K(\varepsilon)$ from the key generation procedure.

Definition 2.12. Let $\epsilon \in \mathbb{R}$. A QECM scheme (K, E, D) as given in Definition 2.11 is *ϵ-correct* if for all messages $m \in \mathcal{M}$ we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle m | \, D_k \circ E_k \left(m \right) | m \rangle \geq 1 - \epsilon. \tag{2.41}$$

We say that a QECM scheme is *perfectly correct*, or simply *correct*, if it is 0-correct.

We emphasize that correctness for QECMs only considers *classical* messages $m \in \mathcal{M}$. A QECM need not be able to recover any arbitrary quantum state to be correct. As an example, consider the QECM with $\mathcal{M} = \{0, 1\}$ and $\mathcal{K} = \{\varepsilon\}$ where the channel E_ε is given by $\rho \mapsto Z\rho Z^\dagger$ for the Pauli operator $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ and $D_\varepsilon = \mathrm{Id}$. A direct calculation yields $D_\varepsilon \circ E_\varepsilon(|+\rangle\langle +|) \neq |+\rangle\langle +|$ where $|+\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right)$. However, for both $b \in \{0, 1\}$, it holds that $D_\varepsilon \circ E_\varepsilon(|b\rangle\langle b|) = |b\rangle\langle b|$ and so this QECM is perfectly correct.

We now proceed to define *augmented encryption of classical messages* (AQECM) schemes. In short, an AQECM scheme is identical to a QECM scheme, with the exception that the decoding channel D produces an additional "flag" qubit as part of its output. We will denote the state space of this qubit by F.

Definition 2.13. An *augmented encryption of classical messages* (AQECM) scheme is a triplet of channels (K, E, D) of the form

$$K : \mathcal{L}(\mathbb{C}) \to \mathcal{L}\left(\mathsf{K}\right), \quad E : \mathcal{L}\left(\mathsf{K} \otimes \mathsf{M}\right) \to \mathcal{L}\left(\mathsf{C}\right), \quad \text{and} \quad D : \mathcal{L}\left(\mathsf{K} \otimes \mathsf{C}\right) \to \mathcal{L}\left(\mathsf{M} \otimes \mathsf{F}\right) \tag{2.42}$$

for Hilbert space K, M, C, and F where, in particular, $\mathsf{K} = \mathbb{C}^{\mathcal{K}}$, $\mathsf{M} = \mathbb{C}^{\mathcal{M}}$, and $\mathsf{F} = \mathbb{C}^{\{0,1\}}$ for alphabets $\mathcal{K}$ and $\mathcal{M}$. As in Definition 2.11, the elements of $\mathcal{K}$ and $\mathcal{M}$ are called the keys and messages of the scheme. From $D : \mathcal{L}(\mathsf{K} \otimes \mathsf{M}) \to \mathcal{L}(\mathsf{M} \otimes \mathsf{F})$, we also define the map $\overline{D} : \mathcal{L}(\mathsf{K} \otimes \mathsf{C}) \to \mathcal{L}(\mathsf{M})$ by

$$\rho \mapsto \left(I_{\mathsf{M}} \otimes \langle 1 |_{\mathsf{F}} \right) D(\rho) \left(I_{\mathsf{M}} \otimes |1\rangle_{\mathsf{F}} \right), \tag{2.43}$$

which is to say that $\overline{D}$ first applies the channel D and then "reduces" to the case where the qubit is measured to be in the $\mathtt{1}$ state, followed by discarding this qubit.

For every key $k \in \mathcal{K}$, we also define the maps $E_k : \mathcal{L}(\mathsf{M}) \to \mathcal{L}(\mathsf{C})$, $D_k : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{M})$, and $\overline{D}_k : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{M})$ by $\rho \mapsto E(k \otimes \rho)$, $\rho \mapsto D(k \otimes \rho)$, and $\rho \mapsto \overline{D}(k \otimes \rho)$, respectively.

As for QECM schemes, from this point on whenever we mention an "AQECM scheme (K, E, D) as given in Definition 2.13" we are assuming that the domains and codomains are defined and denoted exactly as in Definition 2.13.

We emphasize that the *only* difference between the syntaxes of a QECM and an AQECM scheme is the additional "flag" qubit in the codomain of the decoding map D.

Note that the map $\overline{D}$, and by extension every map $\overline{D}_k$, may not be a channel. Indeed, the map $\rho \mapsto \langle \mathtt{1} | \rho | \mathtt{1} \rangle$ applied to the F register produced by D to obtain $\overline{D}$ is not, in general, trace preserving. It may in fact be trace *decreasing*, i.e. $\mathrm{Tr}\big(\overline{D}(\sigma)\big) < \mathrm{Tr}(\sigma)$. Nonetheless, these maps will be useful mathematical tools to define and study properties of AQECM schemes. We take a moment to look at them from a more operational perspective.

For any density operator ρ and key $k \in \mathcal{K}$, we may express $D_k(\rho)$ as

$$D_k(\rho) = \rho_{\mathtt{00}} \otimes |\mathtt{0}\rangle\langle\mathtt{0}| + \rho_{\mathtt{01}} \otimes |\mathtt{0}\rangle\langle\mathtt{1}| + \rho_{\mathtt{10}} \otimes |\mathtt{1}\rangle\langle\mathtt{0}| + \rho_{\mathtt{11}} \otimes |\mathtt{1}\rangle\langle\mathtt{1}| \tag{2.44}$$

where both $\rho_{\mathtt{00}}$ and $\rho_{\mathtt{11}}$ are subnormalized density operators whose traces sum to one, which is to say that $\rho_{\mathtt{00}}, \rho_{\mathtt{11}} \in \mathcal{D}_{\bullet}(\mathsf{M})$ and $\mathrm{Tr}(\rho_{\mathtt{00}}) + \mathrm{Tr}(\rho_{\mathtt{11}}) = 1$. If a computational basis measurement is then applied to the F register, the resulting state is given by

$$(\mathrm{Id}_{\mathsf{M}} \otimes \Delta_{\mathsf{F}})\, D_k(\rho) = \rho_{\mathtt{00}} \otimes |\mathtt{0}\rangle\langle\mathtt{0}| + \rho_{\mathtt{11}} \otimes |\mathtt{1}\rangle\langle\mathtt{1}|\,. \tag{2.45}$$

Then, the map $\overline{D}_k$ simply "picks out" the $\rho_{\mathtt{11}}$ operator from this expression: $\overline{D}_k(\rho) = \rho_{\mathtt{11}}$. Note that $\mathrm{Tr}(\rho_{\mathtt{11}})$ is precisely the probability that the measurement outputs $\mathtt{1}$. From this perspective, it is easy to see that $\overline{D}_k(\rho)$ is simply the state on the M register after application of D_k to ρ, conditioned on the F register being measured to be in the $\mathtt{1}$ state and scaled by the probability that this particular measurement outcome is produced.

The precise meaning of the flag qubit produced by the decryption map of an AQECM scheme can and will vary by context. In general, the flag qubit is set to the $\mathtt{0}$ state if the decryption channel D_k "rejects" the input state and is set to $\mathtt{1}$ if it "accepts" it, where the precise meaning of these words depends of the context. For example, Broadbent and Islam [BI20] use this flag to study AQECM schemes as authentication schemes for classical messages which produce quantum outputs. They call these *robust AQECM* schemes. There, a flag qubit in the state $\mathtt{0}$ is meant to indicate that noise or adversarial action changed the encoded state sufficiently that the wrong message will be recovered. In the context of tamper

evidence, the flag qubit will be set to the state $\mathtt{0}$ if non-trivial eavesdropping is detected.[16] Nonetheless, in all contexts we will expect an AQECM to accept any unmodified encodings of its messages when the same key is used.

This discussion leads us to the following definition of correctness for AQECM schemes.

Definition 2.14. Let $\epsilon \in \mathbb{R}$. An AQECM scheme (K, E, D) as given in Definition 2.13 is *ϵ-correct* if for all messages $m \in \mathcal{M}$ we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle m| \overline{D}_k \circ E_k(m) |m\rangle \geq 1 - \epsilon. \tag{2.46}$$

Our expectation that the flag qubit should be found in the $\mathtt{1}$ state is represented in the above definition by the fact that it uses the maps $\overline{D}_k$ and not the channels D_k.

It is intuitively clear that AQECM schemes are, as the name suggests, an augmentation of QECM schemes. The following lemma formalizes this idea by confirming that discarding the flag register F of an AQECM scheme yields a QECM scheme and that correctness is preserved through this transformation.

Lemma 2.15. Let (K, E, D) be an ϵ-correct AQECM scheme as given in Definition 2.13. Then, $(K, E, (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D)$ is a QECM scheme which is ϵ-correct.

Proof: We directly see that $(\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D$ is a channel from $\mathcal{L}(\mathsf{K} \otimes \mathsf{C})$ to $\mathcal{L}(\mathsf{M})$ and so the triplet $(K, E, (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D)$ satisfies the syntactical conditions to be a QECM. It now suffices to show that this QECM is ϵ-correct.

By recalling that $\mathrm{Tr}_{\mathsf{F}}(\rho) = \sum_{b \in \{\mathtt{0},\mathtt{1}\}} \langle b| \rho |b\rangle$ and the definition of $\overline{D}$, for every key $k \in \mathcal{K}$ and message $m \in \mathcal{M}$ we find that

$$\begin{aligned} (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D_k \circ E_k(m) &= \sum_{b \in \{\mathtt{0},\mathtt{1}\}} (I_{\mathsf{M}} \otimes \langle b|) (D_k \circ E_k)(m) (I_{\mathsf{M}} \otimes |b\rangle) \\ &= \overline{D}_k \circ E_k(m) + (I_{\mathsf{M}} \otimes \langle \mathtt{0}|) (D_k \circ E_k)(m) (I_{\mathsf{M}} \otimes |\mathtt{0}\rangle) \end{aligned} \tag{2.47}$$

from which it follows that

$$\langle m| (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D_k \circ E_k(m) |m\rangle \geq \langle m| \overline{D}_m \circ E_k(m) |m\rangle \tag{2.48}$$

since $(\langle m| \otimes \langle \mathtt{0}|) (D_k \circ E_k)(m) (|m\rangle \otimes |\mathtt{0}\rangle) \geq 0$ by virtue of $D_k \circ E_k(m)$ being a positive semidefinite operator. Thus, for all $m \in \mathcal{M}$, we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle m| (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D_k \circ E_k(m) |m\rangle \geq \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle m| \overline{D}_k \circ E_k(m) |m\rangle \geq 1 - \epsilon, \tag{2.49}$$

[16] As previously discussed, we sketch later in Section 2.7.3.1 a separation robustness and tamper evidence.

as the original AQECM scheme is assumed to be ϵ-correct, which is the desired result. ∎

2.3.2 Tamper Evidence

With the notion of AQECM schemes in hand, we can now define tamper evidence.

In a nutshell, an AQECM scheme exhibits tamper evidence if the party which receives a ciphertext — the one implementing the decoding channel D — can detect any non-trivial eavesdropping. This notion was first presented by Gottesman [Got03], and we review it here. To place Gottesman's notion of tamper evidence in our AQECM framework, we proceed in two steps. We first formalize in Definition 2.16 *tamper attacks* against a given AQECM schem and we then state in Definition 2.17 the condition that an AQECM scheme must satisfy against all tamper attacks to be considered *tamper-evident*. Up to a difference in presentation and some minor differences discussed later, this is the definition of tamper evidence given in [Got03].

A tamper attack can be understood as an adversary's attempt to extract, undetected, information from an intercepted ciphertext $E_k(m) \in \mathcal{D}(\mathsf{C})$. We model any information that the adversary managed to extract as a quantum system held in a register A. Moreover, we assume that the eavesdropping adversary must allow *something*, i.e. a possibly modified ciphertext, to reach the receiver. Otherwise, we may assume that the eavesdropping will necessarily be detected. Under this model, we can represent the adversary's action as a channel which acts on the space of the encoded messages C to produce a state on the space $\mathsf{C} \otimes \mathsf{A}$. This is formalized in Definition 2.16 and sketched in Figure 2.5.

Definition 2.16. Let (K, E, D) be an AQECM scheme as given in Definition 2.13. A *tamper attack* against it is a channel $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ where A is a Hilbert space.

We can now formalize the notion of tamper evidence. To do so will require us to consider certain subnormalized states acting on the additional Hilbert space A produced by a tamper attack A. We take a moment to examine these subnormalized states in some detail before defining tamper evidence.

For any key k, message m, and tamper attack $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ against an AQECM scheme (K, E, D), we consider the operator

$$\left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m), \tag{2.50}$$

and we note that this is a subnormalized density operator on the additional register A produced by the tamper attack A, *i.e.* an element of $\mathcal{D}_\bullet(\mathsf{A})$. In Figure 2.5, this is the operator $\rho_{k,m}$.

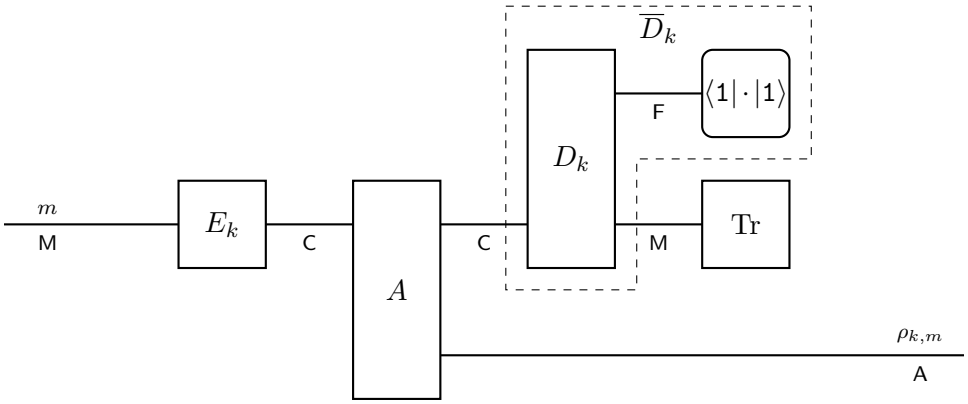


Figure 2.5: An AQECM scheme (K, E, D) subject to a tamper attack A, as considered in Definition 2.16 and Definition 2.17. For any given message m and key k, the final subnormalized state held by the adversary if the receiver does not detect any eavesdropping is $\rho_{k,m} \in \mathcal{D}_\bullet(\mathsf{A})$.

Operationally, this represents the state the adversarial eavesdropper holds when the message m is sent using the key k, *conditioned* on the receiver accepting *and scaled* by the probability of this acceptance.

An AQECM scheme will be tamper evident if there is a high probability, taken over the choice the of key k, that the trace distance between the subnormalized states $\rho_{k,m}$ and $\rho_{k,m'}$ is small for any two messages $m, m' \in \mathcal{M}$. This is formalized in Definition 2.17. Recall that $\left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k : \mathcal{L}(\mathsf{M}) \to \mathcal{L}(\mathsf{A})$ is a linear operator. Hence, the difference between the image of m and m' under the action of this map can be more succinctly expressed as $\left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m - m')$. We will often make use of this property to make our notation more compact.

Definition 2.17. Let $\delta \in \mathbb{R}$. An AQECM scheme (K, E, D) as given in Definition 2.13 is *δ-tamper-evident* if for all messages $m, m' \in \mathcal{M}$ and any tamper attack A against it we have that

$$\Pr_{k \leftarrow K(\varepsilon)} \left[\frac{1}{2} \left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m - m') \right\|_1 \leq \delta \right] \geq 1 - \delta. \tag{2.51}$$

We know that we can interpret the trace-distance between two density operators as a measure of how well two quantum states can be distinguished. But, how should we interpret the trace distance between two *subnormalized* density operators in this context? The following lemma can be understood as stating that if the trace distance between two subnormalized density operators is small, then either the trace distance between their cor-

responding renormalized operators is small, or both of their traces are small.[17]

Lemma 2.18. Let $\rho_0, \rho_1 \in \mathcal{D}(\mathsf{A})$ be two density operators and let $t_0, t_1 \in \mathbb{R}$ be two non-negative real numbers. Then,

$$\frac{\max\{t_0, t_1\}}{2} \cdot \frac{1}{2}\|\rho_0 - \rho_1\|_1 \leq \frac{1}{2}\|t_0\rho_0 - t_1\rho_1\|_1. \tag{2.52}$$

Proof: First, note that $\frac{1}{2}|t_0 - t_1| \leq \frac{1}{2}\|t_0\rho_0 - t_1\rho_1\|_1$ by the fact that the trace distance is contractive under the action of any channel, including, as needed for this case, the trace. Without loss of generality, assume that $t_0 = \max\{t_0, t_1\}$. Then,

$$\begin{aligned} \frac{t_0}{2}\|\rho_0 - \rho_1\|_1 &= \frac{1}{2}\|t_0\rho_0 - t_1\rho_1 + (t_1 - t_0)\rho_1\|_1 \\ &\leq \frac{1}{2}\|t_0\rho_0 - t_1\rho_1\|_1 + \frac{1}{2}\cdot|t_0 - t_1| \end{aligned} \tag{2.53}$$

which, with the previous inequality, yields that $\frac{t_0}{2}\|\rho_0 - \rho_1\|_1 \leq \|t_0\rho_0 - t_1\rho_1\|_1$. Dividing both sides by two then yields the desired result. ∎

In the context of tamper-evident encryption, this means that if $\frac{1}{2}\left\|\rho_{k,m} - \rho_{k,m'}\right\|_1$ is small, then either the eavesdropping attempt will be detected with high probability (in the case that the traces of both $\rho_{k,m}$ and $\rho_{k,m'}$ are small), or it will yield very little information on the plaintext if it is undetected (in the case where the trace distance between the renormalized states $\tilde{\rho}_{k,m} = \mathrm{Tr}(\rho_{k,m})^{-1}\rho_{k,m}$ and $\tilde{\rho}_{k,m'} = \mathrm{Tr}\big(\rho_{k,m'}\big)^{-1}\rho_{k,m'}$ is small, assuming for simplicity here that neither has trace 0). Indeed, in this second case, the adversary will not be able to distinguish between the scenario where the message m is sent or the one where the message m' is sent, *even if they later learn the key k which was used*.

Definition 2.17 has two notable differences with the definition of tamper evidence originally given by Gottesman. First, we do not require the AQECM scheme to satisfy any type of encryption security guarantee to be a tamper-evident scheme. We consider tamper evidence and encryption completely separately for the time being, but we will show later in Theorem 2.30 that any sufficiently good tamper-evident scheme will indeed be an encryption scheme. Second, Gottesman does not make an explicit statement on the probability of the trace distance being small. Instead, it is required that the trace distance considered in Equation 2.51 be at most δ for at least $(1 - \delta)\cdot|\mathcal{K}|$ keys in $\mathcal{K}$. Of course, these notions coincide if the keys are sampled uniformly at random by the key generation procedure, but our formulation of this definition is a bit more general.

[17] This point was also made, but not quantified as we did here, by Gottesman [Got03].

Before proceeding further, we make explicit two simple but useful facts which immediately follow from Definition 2.17.

Fact 2.19. A δ-tamper-evident AQECM scheme is also δ'-tamper evident if $\delta' > \delta$.

Fact 2.20. Every AQECM scheme is 1-tamper-evident, none is δ-tamper-evident if $\delta < 0$.

The definition of the security notion of tamper evidence is a bit unorthodox because it is not expressed as an expectation over the choice of the key. The following lemma expresses sufficient and necessary conditions for an AQECM scheme to be δ-tamper-evident as bounds concerning the expected trace distance

Lemma 2.21. Let $S = (K, E, D)$ be an AQECM as given in Definition 2.13.

1. If for all tamper attacks $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ and all messages $m, m' \in \mathcal{M}$ we have that
$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \circ \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m - m') \right\|_1 \leq \delta \tag{2.54}$$
then S is $\sqrt{\delta}$-tamper-evident.

2. If S is δ-tamper-evident, then for all tamper attacks $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ and all messages $m, m' \in \mathcal{M}$ we have that
$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \circ \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m - m') \right\|_1 \leq 2\delta - \delta^2 \leq 2\delta. \tag{2.55}$$

Proof: We begin by showing the first point. By Markov's inequality (Theorem 2.1) and our assumed upper bound on the expectation we have
$$\left(1 - \Pr_{k \leftarrow K(\varepsilon)} \left[\frac{1}{2} \left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \circ \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m - m') \right\|_1 < \sqrt{\delta} \right] \right) \leq \frac{\delta}{\sqrt{\delta}} = \sqrt{\delta} \tag{2.56}$$
which implies that
$$\Pr_{k \leftarrow K(\varepsilon)} \left[\frac{1}{2} \left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \circ \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m - m') \right\|_1 < \sqrt{\delta} \right] \geq 1 - \sqrt{\delta} \tag{2.57}$$
and so S is $\sqrt{\delta}$-tamper-evident.

We now show the second point. Note that if $\delta \geq \frac{1}{2}$, then the desired inequality on the expectation immediately holds as the trace distance between two subnormalized states can never be more than 1. If $\delta < \frac{1}{2}$, then it suffices to invoke the concentration inequality of Lemma 2.2, with an upper bound of $\beta = 1$. This yields, after a few algebraic manipulations

and recalling our assumption that S is δ-tamper-evident, that

$$\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\circ \mathrm{Id}_{\mathsf{A}}\right)\circ A\circ E_k(m-m')\right\|_1 \leq \delta + (1-\delta)\cdot\delta = 2\delta - \delta^2 \leq 2\delta \quad (2.58)$$

which is the desired result. ∎

We conclude this section by reiterating what is, morally, Gottesman's main theorem, namely that the theory of tamper-evident schemes is not vacuous. Tamper-evident schemes can be explicitly constructed. Better yet, for any message set $\mathcal{M}$ and $\delta > 0$, there exists a perfectly correct AQECM scheme which is δ-tamper-evident.

Theorem 2.22 ([Got03])**.** Let $\mathcal{M}$ be an alphabet and $\delta \in \mathbb{R}^+$ be a strictly positive real. There exists a perfectly correct δ-tamper-evident AQECM scheme with messages $\mathcal{M}$.

Gottesman explicitly demonstrates this for cases where $\mathcal{M} = \{\mathtt{0},\mathtt{1}\}^n$ for any $n \in \mathbb{N}^+$ by constructing such schemes. This result can be extended to arbitrary alphabets $\mathcal{M}'$ by adding an injective map $i : \mathcal{M} \to \{\mathtt{0},\mathtt{1}\}^{\lceil \log_2 |\mathcal{M}| \rceil}$ and any map $j : \{\mathtt{0},\mathtt{1}\}^{\lceil \log_2 |\mathcal{M}| \rceil} \to \mathcal{M}$ such that $j \circ i$ is the identity on $\mathcal{M}$. Messages $m \in \mathcal{M}$ are then pre-processed with i before being encoded and post-processed by j after being decoded. It is trivial to see that this preserves correctness and tamper evidence.

2.4 Tamper Evidence Implies Encryption

In this section, we show that any sufficiently good tamper-evident scheme is a good encryption scheme. This answers, in the positive, one of Gottesman's open questions [Got03].

We begin by giving a simple definition of encryption for QECMs and AQECMs, which states that the ciphertexts corresponding to two different messages m_0 and m_1 should be close in trace distance when averaging over all possible keys. Our definition is essentially the same as the one given by Ambainis, Mosca, Tapp, and de Wolf [AMTdW00], restricted to only considering the computational basis states and with a relaxation allowing the resulting ciphertexts to differ slightly in trace distance. This is captured in the following definition.

Definition 2.23. Let $\delta \in \mathbb{R}$. A QECM or AQECM scheme (K, E, D) is δ-encrypting if, for all messages $m_0, m_1 \in \mathcal{M}$, we have that

$$\frac{1}{2}\left\|\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} E_k\left(m_0 - m_1\right)\right\|_1 \leq \delta. \quad (2.59)$$

Note that linearity yields $\mathbb{E}_{k\leftarrow K(\varepsilon)} E_k(m_0 - m_1) = \mathbb{E}_{k\leftarrow K(\varepsilon)} E_k(m_0) - \mathbb{E}_{k\leftarrow K(\varepsilon)} E_k(m_1)$,

where the right-hand side is a bit less succinct but a bit more descriptive than what is used in Definition 2.23.

The main theorem we prove in this section, Theorem 2.30, is that any ϵ-correct δ-tamper-evident AQECM scheme is also $\sqrt{19(\delta + \sqrt{2}\epsilon)}$-encrypting. As discussed in Section 2.1.2.1, our proof follows the same ideas as the proof given by Barnum, Crépeau, Gottesman, Smith, and Tapp showing that quantum authentication schemes are also quantum encryption schemes [BCG+02] and proceeds in three steps:

1. We first show in Theorem 2.24 of Section 2.4.1 that any AQECM scheme which does not offer much security as an encryption scheme cannot be very tamper-evident.

2. Next, we show in Theorem 2.27 of Section 2.4.2 that the parallel applications of a δ-tamper-evident scheme and a δ'-tamper-evident scheme yields a $(\delta + \delta')$-tamper-evident scheme. In other words, the parallel applications of AQECM schemes yields a *linear* loss of tamper-evident security.

3. Contrasting the above point with an observation from Barnum et al. that the parallel applications of (A)QECM schemes yields an *exponential* loss of encryption security due to Lemma 2.5, we can boost the weak bound of Theorem 2.24. This yields our main result of this section, Theorem 2.30 of Section 2.4.3.

2.4.1 Bad Encryption Implies Bad Tamper Evidence

We show in this subsection that an AQECM scheme which is not a good encryption scheme cannot be a good tamper-evident scheme.

Theorem 2.24. Let $S = (K, E, D)$ be an ϵ-correct AQECM scheme. If S is *not* δ-encrypting for $\delta \geq 0$, then it is *not* $(1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon})$-tamper-evident.

Proof: We begin by identifying pairs of values (ϵ, δ) for which the claim trivially holds. We can assume that $\delta < 1$ as there is no scheme which is not δ-encrypting for $\delta \geq 1$. We can further assume that $\epsilon \geq 0$ as there does not exist an AQECM scheme which is ϵ-correct for $\epsilon < 0$. If $1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon} < 0$, then the result trivially holds as no AQECM scheme is $\tilde{\delta}$-tamper-evident if $\tilde{\delta} < 0$. Thus, we can further assume that $1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon} \geq 0$ for the remainder of the proof.

Next, we identify two messages and a tamper attack which we will use to demonstrate that S is not sufficiently tamper-evident. As S is not δ-encrypting by assumption, there are two messages $m_0, m_1 \in \mathcal{M}$ such that

$$\frac{1}{2}\left\| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_0) - \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_1) \right\|_1 > \delta. \tag{2.60}$$

By Lemma 2.3, this implies the existence of a distinguishing channel $\Phi : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}})$ satisfying

$$\langle 0| \Phi \left(\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_0) \right) |0\rangle + \langle 1| \Phi \left(\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_1) \right) |1\rangle > 1 + \delta \tag{2.61}$$

which, by linearity, also satisfies

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left(\langle 0| \Phi \left(E_k(m_0) \right) |0\rangle + \langle 1| \Phi \left(E_k(m_1) \right) |1\rangle \right) > 1 + \delta. \tag{2.62}$$

Let $\mathsf{A} = \mathbb{C}^{\{0,1\}}$ be the state space of a single qubit and $A = \mathrm{CGM}(\Phi) : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ be the coherent gentle measurement channel obtained from the distinguishing channel Φ as described in Definition 2.8. Note that A is a tamper attack against the scheme S and that by Theorem 2.10 we have that

$$\frac{1}{2} \| A(\rho) - \rho \otimes |b\rangle\langle b| \|_1 \leq \sqrt{\mathrm{Tr}\left(\rho\right) - \left(\langle b| \Phi(\rho) |b\rangle \right)^2} \tag{2.63}$$

for each $b \in \{0, 1\}$ and all $\rho \in \mathcal{D}_\bullet(\mathsf{C})$.

Before proceeding, we recall that tamper evidence is defined with respect to a probability over keys, and not an expectation. Thus, we use the concentration inequality of Lemma 2.2 to translate Equation 2.62 to this setting and obtain that

$$\begin{aligned} \Pr_{k \leftarrow K(\varepsilon)} \left[\langle 0| \Phi \left(E_k(m_0) \right) |0\rangle + \langle 1| \Phi \left(E_k(m_1) \right) |1\rangle > 1 + \delta' \right] &> \frac{(1+\delta) - (1+\delta')}{2 - (1+\delta')} \\ &= 1 - \frac{1-\delta}{1-\delta'} \end{aligned} \tag{2.64}$$

for a yet-to-be-determined value of δ' satisfying $0 \leq \delta' + 1 < 2$, as required by Lemma 2.2. Similarly, we use the same concentration inequality and the assumed ϵ-correctness of the AQECM scheme to obtain

$$\Pr_{k \leftarrow K(\varepsilon)} \left[\mathrm{Tr}\left[\overline{D}_k \circ E_k(m_b) \right] \geq 1 - \sqrt{2\epsilon} \right] \geq 1 - \sqrt{\frac{\epsilon}{2}} \tag{2.65}$$

for each $b \in \{0, 1\}$. Note that we are only checking here that D_k accepted, not that it also obtained the correct message. Combining these previous inequalities via the union bound,

we obtain that

$$\Pr_{k\leftarrow K(\varepsilon)}\left[\begin{array}{c}\langle \mathsf{0}|\,\Phi\circ E_k(m_{\mathsf{0}})\,|\mathsf{0}\rangle + \langle \mathsf{1}|\,\Phi\circ E_k(m_{\mathsf{1}})\,|\mathsf{1}\rangle > 1+\delta' \\ \wedge \\ \mathrm{Tr}\left[\overline{D}_k\circ E_k(m_{\mathsf{0}})\right]\geq 1-\sqrt{2\epsilon} \\ \wedge \\ \mathrm{Tr}\left[\overline{D}_k\circ E_k(m_{\mathsf{1}})\right]\geq 1-\sqrt{2\epsilon}\end{array}\right] > 1-\frac{1-\delta}{1-\delta'}-2\sqrt{\frac{\epsilon}{2}}. \quad (2.66)$$

Let $\mathcal{K}_{\mathrm{Good}}\subseteq\mathcal{K}$ be the set of keys satisfying the event in the previous inequality. In other words, these are the keys which satisfy two conditions: the encodings of m_{0} and m_{1} under these keys are distinguished sufficiently well by Φ and, in the absence of any adversarial attack, the encodings of m_{0} and m_{1} under these keys will be accepted with sufficiently high probability by the decryption map. Our goal now is to show that the tamper attack A succeeds well enough when a key in $\mathcal{K}_{\mathrm{Good}}$ is used. We note that, by definition,

$$\Pr_{k\leftarrow K(\varepsilon)}\left[k\in\mathcal{K}_{\mathrm{Good}}\right] > 1-\frac{1-\delta}{1-\delta'}-\sqrt{2\epsilon} \quad (2.67)$$

and so selecting δ' in such a way that the right-hand side of this inequality is at least 0 is sufficient to ensure that $\mathcal{K}_{\mathrm{Good}}$ is non-empty.

Now, for each $b\in\{\mathsf{0},\mathsf{1}\}$ and key $k\in\mathcal{K}_{\mathrm{Good}}$, define the operators

$$\rho_{k,b}=\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\mathrm{Id}_{\mathsf{A}}\right)\circ A\circ E_k(m_b) \quad \text{and} \quad \sigma_{k,b}=\mathrm{Tr}\left[\overline{D}_k\circ E_k(m_b)\right]\cdot|b\rangle\langle b|\,. \quad (2.68)$$

All of these operators are subnormalized density operators on a single qubit, i.e. elements of $\mathcal{D}_\bullet(\mathbb{C}^{\{\mathsf{0},\mathsf{1}\}})$, and so represent a possible state for the system held by an eavesdropping adversary who used the tamper attack A against the scheme S, scaled by the probability that this attack was undetected. The ρ operators correspond to the "real" scenario where the tamper attack A is actually applied. The σ operators, on the other hand, correspond to an "ideal" scenario where A is not actually applied, but where the adversary is simply given directly the bit corresponding to the message which was encoded.

We pause a moment to emphasize that if we can show that for every $k\in\mathcal{K}_{\mathrm{Good}}$ we have that

$$\frac{1}{2}\|\rho_{k,\mathsf{0}}-\rho_{k,\mathsf{1}}\| > 1-\frac{1-\delta}{1-\delta'}-\sqrt{2\epsilon} \quad (2.69)$$

then we will have shown that (K,E,D) is not $\left(1-\frac{1-\delta}{1-\delta'}-\sqrt{2\epsilon}\right)$-tamper-evident. This is now our goal.

For any key $k \in \mathcal{K}_{\text{Good}}$, we obtain via the triangle inequality that

$$\|\sigma_{k,0} - \sigma_{k,1}\|_1 \leq \|\sigma_{k,0} - \rho_{k,0}\|_1 + \|\rho_{k,0} - \rho_{k,1}\|_1 + \|\rho_{k,1} - \sigma_{k,1}\|_1, \tag{2.70}$$

which in turn implies that

$$\|\rho_{k,0} - \rho_{k,1}\|_1 \geq \|\sigma_{k,0} - \sigma_{k,1}\|_1 - \|\rho_{k,0} - \sigma_{k,0}\|_1 - \|\rho_{k,1} - \sigma_{k,1}\|_1. \tag{2.71}$$

Thus, a lower-bound on $\|\rho_{k,0} - \rho_{k,1}\|_1$ can be found by finding a lower bound on $\|\sigma_{k,0} - \sigma_{k,1}\|_1$ and an upper bound on $\sum_{b\in\{0,1\}} \|\rho_{k,b} - \sigma_{k,b}\|_1$.

By direct computation, we find that

$$\|\sigma_{k,0} - \sigma_{k,1}\|_1 = \text{Tr}\left[\overline{D}_k \circ E_k(m_0)\right] + \text{Tr}\left[\overline{D}_k \circ E_k(m_1)\right] \geq 2 - 2\sqrt{2\epsilon} \tag{2.72}$$

where the inequality follows from the assumption that $k \in \mathcal{K}_{\text{Good}}$. Next, by Equation 2.63, we have that

$$\sum_{b\in\{0,1\}} \frac{1}{2}\|A(E_k(m_b)) - E_k(m_b) \otimes |b\rangle\langle b|\|_1 \leq \sum_{b\in\{0,1\}} \sqrt{1 - \left(\langle b| \Phi(E_k(m_b)) |b\rangle\right)^2}. \tag{2.73}$$

While we do not have much knowledge on the individual values of $\sqrt{1 - \left(\langle b| \Phi(E_k(m_b)) |b\rangle\right)^2}$ for each value of $b \in \{0,1\}$, we do know that $\sum_{b\in\{0,1\}} \langle b| \Phi(E_k(m_b)) |b\rangle > 1 + \delta'$ by virtue of $k \in \mathcal{K}_{\text{Good}}$. Solving the optimization problem of maximizing the right-hand side of Equation 2.73 under this constraint, we find that

$$\begin{aligned} \sum_{b\in\{0,1\}} \frac{1}{2}\|A(E_k(m_b)) - E_k(m_b) \otimes |b\rangle\langle b|\|_1 &< 2\sqrt{1 - \left(\frac{1+\delta'}{2}\right)^2} \\ &= \sqrt{4 - (1+\delta')^2} \\ &< 2\sqrt{1-\delta'} \end{aligned} \tag{2.74}$$

where the last inequality follows by algebraic manipulation and dropping a few terms,[18] which we do to simplify later arguments. Now, as the trace distance is contractive under the map $\left(\text{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \text{Id}_{\mathsf{A}}$ and applying this map to the states in the above trace distance will allows us to recover our ρ and σ operators, we have that

$$\sum_{b\in\{0,1\}} \frac{1}{2}\|\rho_{k,b} - \sigma_{k,b}\|_1 \leq 2\sqrt{1-\delta'}. \tag{2.75}$$

[18] Specifically, $\sqrt{4 - (1+\delta')^2} = \sqrt{4 - (2 + (\delta' - 1))^2} = \sqrt{4(1-\delta') - (\delta' - 1)^2} < \sqrt{4(1-\delta')}$.

Combining all the above, we obtain that

$$\frac{1}{2}\|\rho_{k,0} - \rho_{k,1}\|_1 > 1 - 2\sqrt{1-\delta'} - \sqrt{2\epsilon}. \tag{2.76}$$

for all keys $k \in \mathcal{K}_{\text{Good}}$.

It now suffices to fix the value of δ'. We choose to take $\delta' = 1 - \left(\frac{1-\delta}{2}\right)^{2/3}$ as this implies that $2\sqrt{1-\delta'} = \frac{1-\delta}{1-\delta'} = \sqrt[3]{4(1-\delta)}$. In particular, we see that $\delta < 1$ implies that $\delta' < 1$, as required, and that $\mathcal{K}_{\text{Good}} \neq \emptyset$ as $\Pr_{k \leftarrow K(\varepsilon)}[k \in \mathcal{K}_{\text{Good}}] > 1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon} \geq 0$. Thus,

$$\Pr_{k \leftarrow K(\varepsilon)}\left[\frac{1}{2}\|\rho_{k,0} - \rho_{k,1}\|_1 > 1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon}\right] > 1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon} \tag{2.77}$$

from which we conclude that the scheme S is not $(1 - \sqrt[3]{4(1-\delta)} - \sqrt{2\epsilon})$-tamper-evident. ∎

2.4.2 On The Parallel Composition of AQECM Schemes

We study here the parallel composition of AQECM schemes and how it affects the correctness and tamper evidence of such schemes.

The parallel composition of two AQECM schemes S' and S'' is a scheme $S = S' \otimes S''$ which encodes pairs of messages (m', m'') with pairs of independent keys (k', k'') to obtain pairs of ciphertexts (ρ', ρ''). The first component of these pairs is treated with the scheme S' and the second with scheme S''. Note that, formally, we will model these pairs as tensor products, such as $\rho' \otimes \rho''$. Ciphertexts are also decoded independently, *except* for the production of the single flag qubit necessary for a well defined AQECM. The flag qubit produced by the overall scheme S will be the result of the logical-and of the flag qubits independently produced by the component schemes S' and S''. In other words, S accepts if and only if both of the schemes S' and S'' accept their respective ciphertexts.

This construction is formalized in Definition 2.25. Note that Swap channels are included to ensure that keys, plaintexts, and ciphertexts are correctly ordered and distributed to the component schemes. The V channel, for its part, takes the two flag qubits produced by the component schemes and computes their logical-and.

Definition 2.25. Let $S' = (K', E', D')$ and $S'' = (K'', E'', D'')$ be two AQECM schemes as given in Definition 2.13 where the notation for all associated alphabets and Hilbert spaces carry an additional prime $'$ or double prime $''$ to clarify with which scheme it is associated.

We define the parallel composition of these schemes, denoted $S' \otimes S''$, as the triplet of

channels (K, E, D) defined by

$$K = K' \otimes K'' \tag{2.78}$$

$$E = \left(E' \otimes E''\right) \circ \mathrm{Swap}^{(1,3,2,4)}_{\mathsf{K}',\mathsf{K}'',\mathsf{M}',\mathsf{M}''} \tag{2.79}$$

$$D = V \circ \left(D' \otimes D''\right) \circ \mathrm{Swap}^{(1,3,2,4)}_{\mathsf{K}',\mathsf{K}'',\mathsf{C}',\mathsf{C}''} \tag{2.80}$$

where V is the channel defined by

$$\rho \mapsto \sum_{b',b'' \in \{0,1\}} \left(I_{\mathsf{M}'} \otimes \left\langle b' \right| \otimes I_{\mathsf{M}''} \otimes \left\langle b'' \right|\right) \rho \left(I_{\mathsf{M}'} \otimes \left| b' \right\rangle \otimes I_{\mathsf{M}''} \otimes \left| b'' \right\rangle\right) \otimes (b' \wedge b'') \tag{2.81}$$

where $b' \wedge b'' \in \{0, 1\}$ and $b' \wedge b'' = 1 \iff b' = b'' = 1$. The channel V can be understood as measuring and discarding the flag qubits produced by D' and D'' in the computational basis and outputing a new flag qubit whose state is the logical-and of these measurement outcomes.

Note that this yields an AQECM whose keys are the elements of $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$ and whose messages are the elements of $\mathcal{M} = \mathcal{M}' \times \mathcal{M}''$. Further note that, for any key $(k', k'') \in \mathcal{K}' \times \mathcal{K}''$, we have that

$$E_{(k',k'')} = E'_{k'} \otimes E''_{k''}, \quad D_{(k',k'')} = V \circ (D'_{k'} \otimes D''_{k''}), \quad \text{and} \quad \overline{D}_{(k',k'')} = \overline{D'}_{k'} \otimes \overline{D''}_{k''}. \tag{2.82}$$

We first show how correctness is preseved under this construction.

Theorem 2.26. Let S' and S'' be two AQECM schemes, as given in Definition 2.25, which are ϵ'- and ϵ''-correct, respectively. Then, $S = S' \otimes S''$ is $(\epsilon' + \epsilon'')$-correct.

Proof: Let $m = (m', m'') \in \mathcal{M}' \times \mathcal{M}''$ be a message for S. We have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\langle m \right| \overline{D}_k \circ E_k(m) \left| m \right\rangle \tag{2.83}$$

$$= \mathop{\mathbb{E}}_{\substack{k' \leftarrow K'(\varepsilon) \\ k'' \leftarrow K''(\varepsilon)}} \left\langle m', m'' \right| \overline{D'}_{k'} \circ E'_{k'}(m') \otimes \overline{D''}_{k''} \circ E''_{k''}(m'') \left| m', m'' \right\rangle \tag{2.84}$$

$$= \mathop{\mathbb{E}}_{k' \leftarrow K'(\varepsilon)} \left\langle m' \right| \overline{D'}_{k'} \circ E'_{k'}(m') \left| m' \right\rangle \mathop{\mathbb{E}}_{k'' \leftarrow K''(\varepsilon)} \left\langle m^1 \right| \overline{D''}_{k''} \circ E_{k''}(m'') \left| m'' \right\rangle \tag{2.85}$$

$$\geq (1 - \epsilon') \cdot (1 - \epsilon'') \tag{2.86}$$

$$\geq 1 - (\epsilon' + \epsilon'') \tag{2.87}$$

which is the desired result. ∎

We now show how tamper evidence is preserved under parallel composition.

Theorem 2.27. Let S' and S'' be two AQECM schemes, as given in Definition 2.25, which are δ'- and δ''-tamper-evident, respectively. Then, $S = S' \otimes S''$ is $(\delta' + \delta'')$-tamper-evident.

Proof: Fix a tamper attack $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ against the scheme S. Now, for any key $k = (k', k'') \in \mathcal{K}' \times \mathcal{K}''$ and any message $m = (m', m'') \in \mathcal{M}' \times \mathcal{M}''$, let $\rho_{k,m} \in \mathcal{D}_\bullet(\mathsf{A})$ be the subnormalized state defined by

$$\rho_{k,m} = (\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{A}}) \circ \left(\overline{D}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m). \tag{2.88}$$

For the remainder of the proof, fix two messages $m_0 = (m'_0, m''_0)$ and $m_1 = (m'_1, m''_1)$ in $\mathcal{M}$. It suffices to show that

$$\Pr_{k \leftarrow K(\varepsilon)} \left[\frac{1}{2} \|\rho_{k,m_0} - \rho_{k,m_1}\|_1 \leq \delta_0 + \delta_1 \right] \geq 1 - (\delta_0 + \delta_1). \tag{2.89}$$

By the triangle inequality, we have for any key $k \in \mathcal{K}$ that

$$\frac{1}{2} \left\| \rho_{k,(m'_0,m''_0)} - \rho_{k,(m'_1,m''_1)} \right\|_1 \leq \underbrace{\frac{1}{2} \left\| \rho_{k,(m'_0,m''_0)} - \rho_{k,(m'_1,m''_0)} \right\|_1}_{=N'_k} + \underbrace{\frac{1}{2} \left\| \rho_{k,(m'_1,m''_0)} - \rho_{k,(m'_1,m''_1)} \right\|_1}_{=N''_k} \tag{2.90}$$

where we note the presence of the subnormalized state $\rho_{k,(m'_1,m''_0)}$, corresponding to the "hybrid" message (m'_1, m''_0), in both norms in the right-hand side of the above. It now suffices to show that N'_k is at most δ' with probability at least $1 - \delta'$ over the sampling of k and that N''_k is at most δ'' with probability at least $1 - \delta''$, over the sampling of k. Indeed, by the union bound, this would then imply that the probability over the sampling of the key k that both $N'_k \leq \delta'$ and $N''_k \leq \delta''$ is at least $1 - (\delta' + \delta'')$, which yields the desired result.

We show that $\Pr_{k \leftarrow K(\varepsilon)} [N'_k \leq \delta'] \geq 1 - \delta'$ by constructing a certain family of tamper attacks $\{A'_{k''}\}_{k'' \in \mathcal{K}''}$ against the scheme S'. The corresponding inequality for N''_k can be obtained completely analogously and will not be explicitly proven here.

For all keys $k'' \in \mathcal{K}''$ we define the tamper attack $A'_{k''} : \mathcal{L}(\mathsf{C}') \to \mathcal{L}(\mathsf{C}' \otimes \mathsf{A}')$ against the scheme S', where $\mathsf{A}' = \mathbb{C}^{\{0,1\}} \otimes \mathsf{A}$, by

$$\rho \mapsto \left(\left(\mathrm{Id}_{\mathsf{C}'} \otimes (\mathrm{Tr}_{\mathsf{M}''} \otimes \mathrm{Id}_{\mathbb{C}^{\{0,1\}}}) \circ D''_{k''} \right) \right) \circ A \circ \left(\rho \otimes E^1_{k''}(m''_0) \right). \tag{2.91}$$

In other words, $A'_{k''}$ attacks S' by first generating the state $E''_{k''}(m''_0)$, where k'' parametrizes the attack and m''_0 is determined by the fixed message m_0 we are considering in this proof, and then running the attack A against the ciphertext obtained by combining this state with the one intercepted from the honest users. The attack $A'_{k''}$ then applies the decoding

channel $D''_{k''}$ on the C'' portion of the ciphertext outputted by A, discards the corresponding message register, but *keeps* the flag qubit. For all keys $k' \in \mathcal{K}'$ and messages $m' \in \mathcal{M}'$, let $\sigma_{(k',k''),m'} \in \mathcal{D}_\bullet(\mathsf{A}')$ be the subnormalized state defined by

$$\sigma_{(k',k''),m'} = (\mathrm{Tr}_{\mathsf{M}'} \otimes \mathrm{Id}_{\mathsf{A}'}) \circ \overline{D'}_{k'} \circ A'_{k''} \circ E'_{k'}(m'). \tag{2.92}$$

Since S' is δ'-tamper-evident, we have that

$$\Pr_{k' \leftarrow K'(\varepsilon)} \left[\frac{1}{2} \left\| \sigma_{(k',k''),m'_0} - \sigma_{(k',k''),m'_1} \right\|_1 \leq \delta' \right] \geq 1 - \delta' \tag{2.93}$$

for all $k'' \in \mathcal{K}''$. Indeed, the various keys k'' simply represents different tamper attacks against the scheme S' and, by our assumption on the tamper-evident security of this scheme, the above inequality must hold for *all* attacks. Since sampling (k', k'') from $K(\varepsilon)$ is equivalent to sampling k' from $K'(\varepsilon)$ and k'' from $K''(\varepsilon)$ independently, the above implies that

$$\Pr_{(k',k'') \leftarrow K(\varepsilon)} \left[\frac{1}{2} \left\| \sigma_{(k',k''),m'_0} - \sigma_{(k',k''),m'_1} \right\|_1 \leq \delta' \right] \geq 1 - \delta'. \tag{2.94}$$

To complete the proof, it now suffices to show that

$$\frac{1}{2} \left\| \rho_{(k',k''),(m'_0,m''_0)} - \rho_{(k',k''),(m'_1,m''_0)} \right\|_1 \leq \frac{1}{2} \left\| \sigma_{(k',k''),m'_0} - \sigma_{(k',k''),m'_1} \right\|_1 \tag{2.95}$$

for all keys $(k', k'') \in \mathcal{K}$.

Recall that $\overline{D''}_{k''}$ is defined by $\rho \mapsto (I_{\mathsf{M}''} \otimes \langle 1|_{\mathsf{F}''}) D''_{k''}(\rho) (I_{\mathsf{M}''} \otimes |1\rangle_{\mathsf{F}''})$. Moreover, note that the only difference between the subnormalized states $\rho_{(k',k''),(m',m''_0)}$ and $\sigma_{(k',k''),m'}$ is that the latter is obtained by applying $(\mathrm{Tr}_{\mathsf{M}''} \otimes \mathrm{Id}_{\mathbb{C}^{\{0,1\}}}) \circ D''_{k''}$ to the C'' register produced by the channel A and the former obtained by applying $\mathrm{Tr}_{\mathsf{M}''} \circ \overline{D''}_{k''}$ to this same register. Thus, for any message $m' \in \mathcal{M}''$ and any key $(k', k'') \in \mathcal{K}$, we have that

$$\rho_{(k',k''),(m',m''_0)} = (\langle 1| \otimes I_{\mathsf{A}}) \sigma_{(k',k''),m'} (|1\rangle \otimes I_{\mathsf{A}}) \tag{2.96}$$

from which it follows, using a well-known property of the operator norm $\|\cdot\|_\infty$ (e.g. [Wat18,

Eq. 1.175]), that

$$\begin{aligned}
&\frac{1}{2}\left\|\rho_{(k',k''),(m_0',m_0'')} - \rho_{(k',k''),(m_1',m_0'')}\right\|_1 \\
&\leq \frac{1}{2}\left\|(\langle 1| \otimes I_{\mathsf{A}})\left(\sigma_{(k',k''),m_0'} - \sigma_{(k',k''),m_1'}\right)(|1\rangle \otimes I_{\mathsf{A}})\right\|_1 \\
&\leq \frac{1}{2}\|(\langle 1| \otimes I_{\mathsf{A}})\|_\infty \cdot \left\|\left(\sigma_{(k',k''),m_0'} - \sigma_{(k',k''),m_1'}\right)\right\|_1 \cdot \||1\rangle \otimes I_{\mathsf{A}}\|_\infty \\
&= \frac{1}{2}\left\|\left(\sigma_{(k',k''),m_0'} - \sigma_{(k',k''),m_1'}\right)\right\|_1,
\end{aligned} \tag{2.97}$$

since $\|\langle 1| \otimes I_{\mathsf{A}}\|_\infty = \||1\rangle \otimes I_{\mathsf{A}}\|_\infty = 1$. ∎

Before proceeding, we note that the repeated parallel composition of an AQECM scheme with itself is easily defined.

Definition 2.28. Let S be an AQECM scheme. For all $n \in \mathbb{N}^+$, we define the n-fold parallel repetition of S, denoted by $S^{\otimes n}$, by

$$S^{\otimes n} = \begin{cases} S & \text{if} \quad n = 1 \\ S^{\otimes(n-1)} \otimes S & \text{if} \quad n > 1. \end{cases} \tag{2.98}$$

Finally, the results on how correctness and tamper-evident security is preserved under parallel composition generalizes as expected to the repeated parallel composition of AQECM schemes with themselves.

Lemma 2.29. Let S be an AQECM scheme which is ϵ-correct and δ-tamper-evident. Then, for all $n \in \mathbb{N}^+$, the scheme $S^{\otimes n}$ is $(n\epsilon)$-correct and $(n\delta)$-tamper-evident.

Proof: This follows directly from the previous two results by induction. ∎

2.4.3 Good Tamper Evidence Implies Good Encryption

We can now prove the main result of this section, namely that good tamper-evident schemes are good encryption schemes.

Theorem 2.30. An ϵ-correct δ-tamper-evident AQECM scheme is $\sqrt{19(\delta + \sqrt{2\epsilon})}$-encrypting.

Proof: First, we can assume for this proof that $\delta + \sqrt{2\epsilon} < \frac{1}{19}$ as, otherwise, the result already holds since every AQECM scheme is 1-encrypting.

Let $S = (K, E, D)$ be an ϵ-correct δ-tamper-evident AQECM scheme and consider the scheme $S^{\otimes n} = (K', E', D')$, the n-fold parallel repetition of S with itself for some value $n \in \mathbb{N}^+$ to be determined later. Note that, by Lemma 2.29, $S^{\otimes n}$ is $n\epsilon$-correct.

Assume that S is not α-encrypting for some $\alpha \geq 0$. Then, there exists two messages $m_0, m_1 \in \mathcal{M}$ such that

$$\frac{1}{2}\left\| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_0) - \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_1) \right\|_1 > \alpha. \tag{2.99}$$

Now, let $m'_0 = m_0^{\times n} = (m_0, \ldots, m_0)$ and $m'_1 = m_1^{\times n} = (m_1, \ldots, m_1)$ be the n-fold repetitions of m_0 and m_1. Note that for each $b \in \{0, 1\}$ we have that

$$\mathop{\mathbb{E}}_{k' \leftarrow K'(\varepsilon)} E'_{k'}(m'_b) = \left(\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_b) \right)^{\otimes n}. \tag{2.100}$$

Then, Lemma 2.5 concerning the trace distance between copies of two states yields that

$$\begin{aligned} \frac{1}{2}\left\| \mathop{\mathbb{E}}_{k' \leftarrow K'(\varepsilon)} E'_{k'}(m'_0 - m'_1) \right\|_1 &> 1 - 2\exp\left(-\frac{n}{2} \left(\frac{1}{2} \left\| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} E_k(m_0 - m_1) \right\|_1 \right)^2 \right) \\ &> 1 - 2\exp\left(-\frac{n \cdot \alpha^2}{2} \right) \end{aligned} \tag{2.101}$$

which is to say that $S^{\otimes n}$ is not $\left(1 - 2\exp\left(-\frac{n\cdot\alpha^2}{2}\right)\right)$-encrypting. It follows, by Theorem 2.24, that $S^{\otimes n}$ is not $\left(1 - 2\exp\left(-\frac{n\cdot\alpha^2}{6}\right) - \sqrt{2n\epsilon}\right)$-tamper-evident. However, by Lemma 2.29, $S^{\otimes n}$ is $n\delta$-tamper-evident. Thus, we must have that

$$1 - 2\exp\left(-\frac{n\alpha^2}{6}\right) - \sqrt{2n\epsilon} < n\delta \tag{2.102}$$

which implies that

$$\exp\left(-\frac{n\alpha^2}{6}\right) > \frac{1 - n\delta - \sqrt{2n\epsilon}}{2} > \frac{1 - n(\delta + \sqrt{2\epsilon})}{2} \tag{2.103}$$

where the second inequality follows from the fact that $\sqrt{2n\epsilon} < n\sqrt{2\epsilon}$. We now fix the value of n to be $n = \left\lfloor \frac{1}{2(\delta+\sqrt{2\epsilon})} \right\rfloor$. Note that this quantity is well defined as at least one of δ or ϵ must be larger than 0.[19] Furthermore, our assumption that $\delta + \sqrt{2\epsilon} < \frac{1}{19}$ implies that $n \geq 1$, as required for our construction of $S^{\otimes n}$. Thus, by placing this value for n in the

[19] A perfectly correct AQECM scheme cannot be 0-tamper-evident since an adversary can always correctly guess which key was used with probability $1/|\mathcal{K}| > 0$. Against a perfectly correct scheme, an adversary who correctly guessed the key can use it to recover the plaintext undetected with certainty.

right-hand side of the above inequality, we find that $\exp\left(-\frac{n\alpha^2}{6}\right) > \frac{1}{4}$ which then implies that

$$\alpha^2 < \frac{6\ln(4)}{n} < \frac{6\ln(4)}{\frac{1}{2(\delta+\sqrt{2\epsilon})} - 1} = \frac{12\ln(4)(\delta + 2\sqrt{2\epsilon})}{1 - 2(\delta + \sqrt{2\epsilon})} < \frac{12\ln(4)(\delta + \sqrt{2\epsilon})}{\frac{17}{19}} < 19(\delta + \sqrt{2\epsilon}) \tag{2.104}$$

where the second inequality above follows from the fact that $n > \frac{1}{2(\delta+\sqrt{2\epsilon})} - 1$, the second to last from the assumption that $\delta + \sqrt{2\epsilon} < \frac{1}{19}$, and the last from direct computation. Taking the square root, we obtain that $\alpha < \sqrt{19(\delta + \sqrt{2\epsilon})}$.

We have thus shown that if S is not α-encrypting, then $\alpha < \sqrt{19(\delta + \sqrt{2\epsilon})}$. By contraposition, if $\alpha \geq \sqrt{19(\delta + \sqrt{2\epsilon})}$, then S must be α-encrypting. In particular we have that S is $\sqrt{19(\delta + \sqrt{2\epsilon})}$-encrypting. ∎

2.5 Quantum Money From Any Tamper-Evident Scheme

As discussed in Section 2.1.2.2, quantum money was first conceived by Wiesner in the founding paper of quantum cryptography [Wie83]. By leveraging the no-cloning principle, a bank is able to mint unforgeable banknotes by incorporating into them quantum systems whose specific states are unknown to the general public. In 2011, Gottesman sketched how to obtain a secure quantum money scheme from any secure tamper-evident scheme [Got11].

In this section, we formalize Gottesman's construction of a quantum money scheme from any tamper-evident scheme and we quantify and prove the level of security it achieves. We do this in Section 2.5.2 after reviewing the relevant definitions pertaining to quantum money in Section 2.5.1.

2.5.1 Defining Quantum Money

Formally, we define a quantum money scheme as a triplet of channels: a *key generation* channel K, a *minting* channel M, and a *verification* channel V. To produce a banknote, the bank first samples a classical secret key k via the key generation channel, $k \leftarrow K(\varepsilon)$, and then uses it with the minting channel M to produce a quantum state: $\rho_k = M(k)$. The resulting quantum state ρ_k is the banknote. We then suppose that the bank places this banknote in circulation, but that it eventually receives it back as part of a deposit. The bank can then, with the help of the key k, attempt to validate the banknote via the verification channel V. This channel either accepts by outputting `1` or rejects by outputting `0`. This is captured later in Definition 2.31. We say that a quantum money scheme is secure if it is

infeasible for an adversary having access to a *single* honestly produced banknote to output *two* quantum systems which would both be accepted by the bank's verification procedure. This is formalized in Definition 2.33.

Note that what we have described here is a *symmetric-* or *private-key* quantum money scheme: the key k used to generate the banknote ρ_k is the same as the one used to later validate it. This is in contrast to *public-key* quantum money schemes which have also been discussed in the literature (e.g.: [AC12]). These schemes admit a publicly known verification key which allows anyone — not only the bank — to verify if a supposed banknote is legitimate or not, but which does not aid adversaries in producing forgeries. Public-key quantum money schemes are typically studied in the context of computational assumptions and we do not consider them in this work.

The syntax we have outlined here and which we formalize in the next definition is implicit in Wiesner's original work on quantum money [Wie83]. It is also essentially the one given in Aaronson and Christiano's more modern work on this topic [AC12], up to two changes. First, we identify the public and private keys in Aaronson and Christiano's work to a be a single private key. Second, we move their definition from a computational setting, i.e. considering polynomial-time algorithms, to an information-theoretic setting, i.e. considering arbitrary quantum channels. We also note that our definition abstracts away the problem of determining which key is associated to which banknote. The standard solution, without introducing any computational assumptions, is for the banknote to be actually composed of a pair (ρ_k, s) where s is a classical serial number. This serial number is selected at random and the bank maintains a large database mapping serial numbers to the private keys used to generate and verify the banknotes. We will not consider the issue of properly associating keys to banknotes any further.

Definition 2.31. A *quantum money scheme* is a triplet of channels (K, M, V) of the form

$$K : \mathbb{C} \to \mathcal{L}(\mathsf{K}), \qquad M : \mathcal{L}(\mathsf{K}) \to \mathcal{L}(\mathsf{N}), \quad \text{and} \quad V : \mathcal{L}(\mathsf{K} \otimes \mathsf{N}) \to \mathcal{L}(\mathsf{F}) \tag{2.105}$$

for an alphabet $\mathcal{K}$ and Hilbert spaces $\mathsf{K} = \mathbb{C}^{\mathcal{K}}$, N, and $\mathsf{F} = \mathbb{C}^{\{0,1\}}$. The elements of $\mathcal{K}$ are called the *keys* of the scheme and the banknotes are quantum states on the space N.

Moreover, for every key $k \in \mathcal{K}$ we additionally define the channel $V_k : \mathcal{L}(\mathsf{N}) \to \mathcal{L}(\mathsf{F})$ by $\rho \mapsto V(k \otimes \rho)$ and the map $\overline{V}_k : \mathcal{L}(\mathsf{N}) \to \mathbb{C}$ by $\rho \mapsto \langle 1 | V_k(\rho) | 1 \rangle$.

A quantum money scheme is said to be ϵ-correct if *every* banknote which it can generate, which is to say those produced by any key with a non-zero chance of being sampled, will be accepted with probability at least $1 - \epsilon$. This is analogous to the definition given by Aaronson and Christiano [AC12].

Definition 2.32. Let $\epsilon \in \mathbb{R}$. A quantum money scheme (K, M, V) as given in Definition 2.31 is ϵ-*correct* if for all keys $k \in \mathcal{K}$ we have that

$$\langle k| K(\varepsilon) |k\rangle > 0 \implies \overline{V}_k \circ M(k) \geq 1 - \epsilon. \tag{2.106}$$

We now define our security notion. This notion quantifies an upper bound on the ability of any adversary to transform one honest banknote into two forgeries which would both be accepted, with the same key, by the bank. This notion is implicit in Wiesner's original work [Wie83] and is the one with respect to which Molina, Vidick, and Watrous prove the security of Wiesner's scheme [MVW13]. In particular, the adversarial counterfeiter is not able to query the bank's verification procedure during the course of their attack. These are called *simple counterfeiting attacks*[20] and they are the only ones we consider in this work.

Definition 2.33. Let $\delta \in \mathbb{R}$. A quantum money scheme (K, M, V) as given in Definition 2.31 is δ-*secure* if for all channels $A : \mathcal{L}(\mathsf{N}) \to \mathcal{L}(\mathsf{N} \otimes \mathsf{N})$ we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left(\overline{V}_k \otimes \overline{V}_k\right) \circ A \circ M(k) \leq \delta. \tag{2.107}$$

2.5.2 Gottesman's Construction of Quantum Money

Now that we have the necessary definitions pertaining to quantum money, we can formalize Gottesman's construction of such a scheme from any tamper-evident scheme [Got11]. We do so in Definition 2.34, after which we state the correctness and security of the resulting scheme in Theorem 2.35 and Theorem 2.39 after proving a few lemmas.

As discussed in Section 2.1.2.2, the basic idea of Gottesman's construction is to start with a tamper-evident scheme (K, E, D) and to obtain banknotes by encoding a randomly chosen message with a randomly chosen key. The bank, which knows the key-message pair, can verify a candidate banknote by decoding it according to the tamper-evident scheme and verifying that the state was accepted *and* that the correct message was obtained. In this construction, the keys of the resulting quantum money scheme are pairs composed of a key and a message from the underlying tamper-evident scheme.

A non-trivial issue which was not discussed by Gottesman but which must be considered in a formal construction occurs when the underlying tamper-evident scheme is not perfectly correct. If a tamper-evident scheme (K, E, D) is not perfectly correct, then there may exist key-message pairs (k, m) such that the decryption map D_k only accepts and correctly decodes $E_k(m)$ with arbitrarily small, perhaps even 0, probability. This follows

[20] For discussions on more involved attack models against quantum money schemes, we invite the interested reader to examine the works of Lutowski [Lut10] and of Nagaj, Sattath, Brodutch, and Unruh [NSBU16].

from the definition of correctness for AQECM schemes (Definition 2.14) which only requires each message to be accepted and correctly recovered with sufficient high probability *over the expected choice of the key*. However, correctness of a quantum money scheme (Definition 2.32) requires that banknotes produced with *any valid key* be accepted with high probability. Thus, naively instantiating Gottesman's construction with an AQECM scheme which is not perfectly correct may yield a quantum money scheme without a significant level of correctness.

To solve this issue, we require that the key generation map of the resulting quantum money scheme only samples from key-message pairs (k, m) with a sufficiently high probability that $E_k(m)$ will be correctly decoded and accepted by D_k. More precisely, we parametrize our construction by a real value $\gamma \in \mathbb{R}$ and only sample key-message pairs (k, m) for which it holds that $\langle m| \overline{D}_k \circ E_k(m) |m\rangle \geq 1 - \gamma$. Our construction must also account for the possibility that the value of γ is such that no key-message pair (k, m) satisfies this condition. In this case, we fall back to an essentially trivial scheme which will be correct, but offer no security. All of this is captured in Definition 2.34. Recall that our construction is in the information-theoretic setting and so we need not worry about the possible complexity of identifying and sampling from this subset of key-message pairs. We discuss this point in more detail at the end of this section.

Definition 2.34. Let $S = (K, E, D)$ be an AQECM scheme as given in Definition 2.13, let $\gamma \in \mathbb{R}$ be a real number, and let $\mathcal{G} \subseteq \mathcal{K} \times \mathcal{M}$ be the set of key-message pairs for S satisfying

$$\langle k| K(\varepsilon) |k\rangle > 0 \quad \text{and} \quad \langle m| \overline{D}_k \circ E_k(m) |m\rangle \geq 1 - \gamma. \tag{2.108}$$

We define the quantum money scheme $\mathrm{QM}_\gamma(S) = (K', M', V')$ in one of two ways, depending on if $\mathcal{G}$ is empty or not. In both cases, the keys will be elements of $\mathcal{K}' = \mathcal{K} \times \mathcal{M}$, implying that the Hilbert space of the key space of $\mathrm{QM}_\gamma(S)$ is $\mathsf{K}' = \mathsf{K} \otimes \mathsf{M}$, and the banknotes will be states on the Hilbert space C corresponding to the ciphertexts of S, implying that the state space of the banknotes is $\mathsf{N}' = \mathsf{C}$.

If $\mathcal{G}$ is non-empty, we define $\mathrm{QM}_\gamma(S) = (K', M', V')$ as follows:

- The key generation channel K' for the quantum money scheme $\mathrm{QM}_\gamma(S)$ is obtained by sampling a message m uniformly at random from the message space $\mathcal{M}$ of S and a key k according to the key generation procedure K of S, conditioned on the resulting pair (k, m) being in $\mathcal{G}$.

  More formally, consider a key-message pair (k, m) where k is sampled according to the AQECM's key generation channel and where m is sampled uniformly at random. Let $p_{\mathcal{G}} = \sum_{(k,m)\in\mathcal{G}} \langle k| K(\varepsilon) |k\rangle \cdot \frac{1}{|\mathcal{M}|}$ be the probability that a key-message pair generated

in this way is in $\mathcal{G}$. Note that $p_{\mathcal{G}} > 0$ as $\mathcal{G}$ is non-empty. We then define K' by

$$c \mapsto c \cdot \frac{1}{p_{\mathcal{G}}} \sum_{(k,m)\in\mathcal{G}} \frac{\langle k| K(\varepsilon) |k\rangle}{|\mathcal{M}|} |k,m\rangle\langle k,m| . \tag{2.109}$$

In particular, we see that $\langle k,m| K'(\varepsilon) |k,m\rangle = \frac{\langle k|K(\varepsilon)|k\rangle}{p_{\mathcal{G}}\cdot|\mathcal{M}|}$ for any pair $(k,m) \in \mathcal{G}$.

- On input of a key-message pair (k,m), the minting channel M' encrypts m with the key k using the encryption map E of S. In other words, $M' = E$.

- On input of a key-message pair (k,m) and a candidate banknote ρ, the verification channel V' decrypts ρ using the key k with the AQECM's decryption channel D. It then accepts if D accepts *and* produces m. Otherwise, it rejects.

  Formally, and in full generality, V' is given by

$$\begin{aligned}\rho \mapsto &\left(\sum_{m\in\mathcal{M}} \langle m| \overline{D} \left((I_{\mathsf{K}} \otimes \langle m|_{\mathsf{M}} \otimes I_{\mathsf{C}}) \rho (I_{\mathsf{K}} \otimes |m\rangle_{\mathsf{M}} \otimes I_{\mathsf{C}})\right) |m\rangle\right) (|1\rangle\langle 1| - |0\rangle\langle 0|) \\ &+ \mathrm{Tr}(\rho) |0\rangle\langle 0| .\end{aligned} \tag{2.110}$$

  This can be more easily parsed in the case that ρ is the tensor of a well formed key for the quantum money scheme and some candidate banknote: $\rho = |\tilde{k}\rangle\langle\tilde{k}| \otimes |\tilde{m}\rangle\langle\tilde{m}| \otimes \sigma$. In this case, we see that

$$\begin{aligned} V'(|\tilde{k}\rangle\langle\tilde{k}| \otimes |\tilde{m}\rangle\langle\tilde{m}| \otimes \sigma) &= \langle\tilde{m}| \overline{D}(|\tilde{k}\rangle\langle\tilde{k}| \otimes \sigma) |\tilde{m}\rangle (|1\rangle\langle 1| - |0\rangle\langle 0|) + \mathrm{Tr}(\sigma) |0\rangle\langle 0| \\ &= \langle\tilde{m}| \overline{D}_{\tilde{k}}(\sigma) |\tilde{m}\rangle |1\rangle\langle 1| + \left(\mathrm{Tr}(\sigma) - \langle\tilde{m}| \overline{D}_{\tilde{k}}(\sigma) |\tilde{m}\rangle\right) |0\rangle\langle 0| \\ &= p_{\mathrm{Accept}} \cdot |1\rangle\langle 1| + (1 - p_{\mathrm{Accept}}) \cdot |0\rangle\langle 0| \end{aligned} \tag{2.111}$$

  where $p_{\mathrm{Accept}} = \langle\tilde{m}| \overline{D}_{\tilde{k}}(\sigma) |\tilde{m}\rangle$ is the probability that the ciphertext is accepted and the message $\tilde{m}$ is obtained.[21]

If $\mathcal{G}$ is empty, we define $\mathrm{QM}_\gamma(S) = (K', M', V')$ as in the previous case, but with the following two differences:

- We take $K' : \mathbb{C} \to \mathcal{L}(\mathsf{K}')$ to be defined by

$$c \mapsto c \cdot K(\varepsilon) \otimes \frac{I_{\mathsf{M}}}{\dim \mathsf{M}}, \tag{2.112}$$

[21] Note that the $\mathrm{Tr}(\rho)$ term, which reduces to $\mathrm{Tr}(\sigma) = 1$ in our example, is necessary to ensure that V' remains trace-preserving in the case that $\mathrm{Tr}(\rho) \neq 1$.

which is to say that a key-message pair (k, m) is obtained by sampling m uniformly at random from $\mathcal{M}$ and k according to K.

- We take V' to be defined by $\rho \mapsto \mathrm{Tr}(\rho) \cdot |1\rangle\langle 1|$, i.e., V' always accepts.

We define this construction even in the case where $\mathcal{G}$ is empty so that $\mathrm{QM}_\gamma(S)$ is well defined for any AQECM S and real value γ. In particular, the map $S \mapsto \mathrm{QM}_\gamma(S)$ taking AQECM schemes to quantum money schemes is well defined for all values $\gamma \in \mathbb{R}$.

Note that our definition of $\mathrm{QM}_\gamma(S)$ when $\mathcal{G}$ is empty has no chance of offering any non-trivial security since the verification procedure always accepts. On the other hand, this yields a scheme which is perfectly correct. Thus, we obtain the following theorem concerning the correctness of $\mathrm{QM}_\gamma(S)$ and emphasize that it is independent of the correctness of S.

Theorem 2.35. Let $S = (K, E, D)$ be an AQECM scheme and let $\gamma \in \mathbb{R}_0^+$ be a non-negative real. Then, the quantum money scheme $\mathrm{QM}_\gamma(S) = (K', V', M')$ is γ-correct.

Proof: Let $\mathcal{G}$ be defined as in Definition 2.34. If $\mathcal{G}$ is empty, then V' always accepts by construction and so

$$\langle 1| V'_{k'} \circ M'(k') |1\rangle = 1 \geq 1 - \gamma \tag{2.113}$$

for any key $k' \in \mathcal{K} \otimes \mathcal{M}$. This implies that $\mathrm{QM}_\gamma(S)$ is γ-correct.

If $\mathcal{G}$ is not empty, then any key $k' = (k, m)$ for the quantum money scheme which satisfies $\langle k, m| K'(\varepsilon) |k, m\rangle > 0$ must be in $\mathcal{G}$. For such keys, we have by construction that

$$\langle 1| V'_{k'} \circ M'(k') |1\rangle = \langle m| \overline{D}_k \circ E_k(m) |m\rangle \geq 1 - \gamma. \tag{2.114}$$

Thus, $\mathrm{QM}_\gamma(S)$ is γ-correct. ∎

We now move on to demonstrating the security of $\mathrm{QM}_{\sqrt{\epsilon}}(S)$ if the underlying AQECM scheme $S = (K, E, D)$ is ϵ-correct and δ-tamper-evident. We begin by sketching the ideas involved in this proof, emphasizing that these were first presented by Gottesman [Got11].

Consider an attack A against the quantum money scheme $\mathrm{QM}_{\sqrt{\epsilon}}(S)$. For this attack to succeed, it must, on input of a single ciphertext $E_k(m)$, produce a bipartite state on two registers such that both will be accepted and both will produce a correct message when decoded by the channel D_k. Clearly, an upper bound on the probability of this occurring is the probability that the first instance of D_k accepts, but may not have recovered the correct message, and that the second instance of D_k recovers the correct message, but may not have accepted. Our proof consists of upper-bounding this probability by leveraging the fact that S is tamper-evident. Indeeed, tamper evidence means that any eavesdropping attempt — roughly corresponding to a counterfeiting attack against the resulting quantum

money scheme — will either be detected (i.e. the first instance of the D_k map will reject) with high probability or will not provide much information on the encoded message (i.e. the second instance of the D_k map will not recover m with high probability).

Our goal is now to formalize this argument. There are two technical points which need to be considered to obtain a proper proof:

1. The definition of tamper evidence, Definition 2.17, considers the ability of an adversary to distinguish, undetected, between any two specific plaintexts. To prove the security of $\mathrm{QM}_{\sqrt{\epsilon}}(S)$, we need a statement about the (in)ability of an adversary to determine, undetected, which *specific* message was sent among many possibilities.

2. Recall that the way key-message pairs (k, m) are sampled in $\mathrm{QM}_\gamma(S)$ may be slightly different from the way they would be sampled when evaluating the security of S as a tamper-evident scheme. This occurs, roughly, when $\mathcal{G} \neq \mathcal{K} \times \mathcal{M}$. We will need to account for this difference in our proof.

We give two lemmas to overcome these two points.

The first point is covered by Lemma 2.37. In this lemma, we consider an adversary characterized by two parts which act sequentially. First, there is an initial tamper attack A which, as usual, is understood as trying to obtain some information on the encoded plaintext without being detected. Second, there is a set of channels $\{A_k\}_{k\in\mathcal{K}}$ indexed by the keys of the AQECM scheme which attempt to explicitly guess the plaintext based on what the initial attack A gathered.[22] The fact that these channels are indexed by the keys models the fact that, at this point, we will assume that the adversary knows the key. The lemma bounds the ability of this two-part adversary to remain undetected *and* correctly guess the value of the message encoded by the tamper-evident scheme after they have learned the key. To prove Lemma 2.37, we will make use of another technical lemma, Lemma 2.36. This second lemma gives a bound on how well any quantum channel can distinguish (subnormalized) states from a given set. In this sense, it can be thought of as a generalization of Lemma 2.3 which considered only the case of sets given by precisely two states. Note that, unlike Lemma 2.3, we do not claim that Lemma 2.36 always gives an optimal bound.

The second point is covered by Lemma 2.38, which is a statement purely of probability theory giving an upper bound on the change in the expectation of a non-negative function when going from one random variable to another which is simply the first conditioned on some particular event occurring.

We now state and prove these lemmas.

[22] In the context of our proof of security for the quantum money scheme $\mathrm{QM}_\gamma(S)$ these will be taken to be specifically $A_k = (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D_k$, which is to say the honest decryption channel which discards the flag output. However, our lemma is more general.

Lemma 2.36. Let $\mathcal{M}$ be an alphabet and, for each $m \in \mathcal{M}$, let $\rho_m \in \mathcal{D}_\bullet(\mathsf{H})$ be a subnormalized density operator on a Hilbert space H. Further, let $\delta \in \mathbb{R}$ be a real value such that for all $m_0, m_1 \in \mathcal{M}$ we have that $\frac{1}{2}\|\rho_{m_0} - \rho_{m_1}\|_1 \leq \delta$. Then, for any random variable μ distributed on $\mathcal{M}$ and any channel $\Phi : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathbb{C}^{\mathcal{M}})$, we have that

$$\mathop{\mathbb{E}}_{m\leftarrow\mu} \langle m| \Phi(\rho_m) |m\rangle \leq \left(\max_{m\in\mathcal{M}} \Pr[\mu = m]\right) \cdot (1 - 2\delta) + 2\delta. \tag{2.115}$$

Proof: Let $p_m = \Pr[\mu = m]$ for all $m \in \mathcal{M}$ and let $m' \in \mathcal{M}$ be an element that satisfies $p_{m'} = \max_{m\in\mathcal{M}} \Pr[\mu = m]$. By singling out the m' contribution to the expectation and recalling that $\mathrm{Tr}(\rho_{m'}) = \sum_{m\in\mathcal{M}} \langle m| \rho_{m'} |m\rangle$, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{m\leftarrow\mu} \langle m| \Phi(\rho_m) |m\rangle &= p_{m'} \left\langle m'\right| \Phi(\rho_{m'}) \left|m'\right\rangle + \sum_{m\in\mathcal{M}\setminus\{m'\}} p_m \cdot \langle m| \Phi(\rho_m) |m\rangle \\ &= p_{m'} \,\mathrm{Tr}\left(\rho_{m'}\right) + \sum_{m\in\mathcal{M}\setminus\{m'\}} \left(p_m \langle m| \Phi(\rho_m) |m\rangle - p_{m'} \left\langle m'\right| \Phi(\rho_{m'}) \left|m'\right\rangle\right)\end{aligned} \tag{2.116}$$

As $p_{m'} \geq p_m$ for all $m \in \mathcal{M}$, we can replace $p_{m'}$ by p_m in each term of the summation at the cost of introducing an inequality. At the same time, we recall that $\mathrm{Tr}(\rho_{m'}) \leq 1$. Thus,

$$\begin{aligned}\mathop{\mathbb{E}}_{m\leftarrow\mu} \langle m| \Phi(\rho_m) |m\rangle &\leq p_{m'} + \sum_{\mathcal{M}\setminus\{m'\}} p_m \cdot \left(\langle m| \Phi(\rho_m) |m\rangle - \langle m| \Phi(\rho_{m'}) |m\rangle\right) \\ &\leq p_{m'} + \sum_{\mathcal{M}\setminus\{m'\}} p_m \cdot \left|\langle m| \Phi(\rho_m - \rho_{m'}) |m\rangle\right|\end{aligned} \tag{2.117}$$

By invoking a slight generalization of Lemma 2.3 which we will not formally prove,[23] we can give upper bounds to the absolute values in the summands in terms of the trace distance. Further recalling that the trace distance is contractive under the action of channels, we obtain

$$\begin{aligned}\mathop{\mathbb{E}}_{m\leftarrow\mu} \langle m| \Phi(\rho_m) |m\rangle &\leq p_{m'} + \sum_{m\in\mathcal{M}\setminus m'} p_m \cdot \|\rho_m - \rho_{m'}\|_1 \\ &\leq p_{m'} + (1 - p_{m'}) \cdot 2\delta\end{aligned} \tag{2.118}$$

where we invoke our assumption of the pair-wise trace distance between the subnormalized states we consider and that $\sum_{m\in\mathcal{M}\setminus\{m'\}} p_m = 1 - p_{m'}$. Rearranging the terms yields the desired result. ∎

[23] Recall that Lemma 2.3 implies that $|\langle 0| \Phi(\rho - \sigma) |0\rangle| \leq \|\rho - \sigma\|_1$ for any two subnormalized density operators $\rho, \sigma \in \mathcal{D}_\bullet(\mathsf{H})$ and any channel $\Phi : \mathcal{L}(\mathsf{H}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}})$. The slight generalizing here is changing the codomain of Φ to $\mathcal{L}(\mathbb{C}^{\mathcal{M}})$ and arguing that the inequality does not depend on the element we pick in $\mathcal{M}$.

Note that if $\mathrm{Tr}(\rho_m) = \mathrm{Tr}(\rho_{m'})$ for all $m \in \mathcal{M}$, then we can obtain a better bound. Indeed, under this assumption Lemma 2.3 yields $|\langle m| \Phi(\rho_m - \rho_{m'}) |m\rangle| \leq \frac{1}{2}\|\rho_m - \rho_{m'}\|_1$ instead of $|\langle m| \Phi(\rho_m - \rho_{m'}) |m\rangle| \leq \|\rho_m - \rho_{m'}\|_1$, implying a final bound of $p_{m'} + (1 - p_{m'})\delta$. Moreover, this matches the bound of Lemma 2.3 if $\mathcal{M} = \{\mathtt{0}, \mathtt{1}\}$, μ is uniformly random, and both traces are 1.

We can now state and prove Lemma 2.37.

Lemma 2.37. Let (K, E, D) be a δ-tamper-evident AQECM as given in Definition 2.13 for $\delta \leq 1$ and let μ be a random variable on the messages of this scheme. For any tamper attack $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ against this scheme and any set $\{A'_k : \mathcal{L}(\mathsf{A}) \to \mathcal{L}(\mathsf{M})\}_{k \in \mathcal{K}}$ of channels indexed by the keys of the scheme, we have that

$$\mathop{\mathbb{E}}_{\substack{k \leftarrow K(\varepsilon) \\ m \leftarrow \mu}} \langle m| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \otimes A'_k \right) \circ A \circ E_k(m) |m\rangle \leq \left(\max_{m \in \mathcal{M}} \Pr\left[\mu = m\right] \right) (1 - 4\delta) + 4\delta. \quad (2.119)$$

Proof: For any key k and message m of the AQECM scheme (K, E, D), define the two operators

$$\rho_{k,m} = \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m) \quad \text{and} \quad \sigma_{k,m} = A'_k(\rho_{k,m}). \quad (2.120)$$

More precisely, $\rho_{k,m} \in \mathcal{D}_\bullet(\mathsf{A})$ is the (subnormalized) state held by the adversary after the verifier has verified the ciphertext but before they learn the key, and $\sigma_{k,m} \in \mathcal{D}_\bullet(\mathsf{M})$ can be understood as the adversary's subsequent guess of m after having been given the key k. Note that the $\rho_{k,m}$ operators are precisely those in the left-hand side of Equation 2.119.

Now, by linearity and Lemma 2.36 with $\Phi = \mathrm{Id}_{\mathsf{M}}$, it suffices for us to show that $\frac{1}{2}\left\|\mathbb{E}_{k \leftarrow K(\varepsilon)} \sigma_{k,m} - \mathbb{E}_{k \leftarrow K(\varepsilon)} \sigma_{k,m'}\right\| \leq 2\delta$ for all $m, m' \in \mathcal{M}$ to obtain the desired result. To obtain this bound, we observe that the triangle inequality allows us to take the expectation out of the norm and the fact that the trace distance is contractive under the action of channels allows us to pass from the σ operators to the ρ operators. This yields that

$$\begin{aligned} \frac{1}{2}\left\| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \sigma_{k,m} - \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \sigma_{k,m'} \right\| &\leq \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2}\left\| \sigma_{k,m} - \sigma_{k,m'} \right\| \\ &\leq \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2}\left\| \rho_{k,m} - \rho_{k,m'} \right\|. \end{aligned} \quad (2.121)$$

It now suffices to show that $\mathbb{E}_{k \leftarrow K(\varepsilon)} \frac{1}{2}\left\|\rho_{k,m} - \rho_{k,m'}\right\|_1 \leq 2\delta$. However, this follows directly from the fact that S is assumed to be δ-tamper-evident and Lemma 2.21 which states that δ-tamper-evidence implies a 2δ upper-bound on this expectation. ∎

We now state and prove Lemma 2.38 which will help us overcome the second technical hurdle we had identified. As mentioned, this is a result purely of probability theory.

Lemma 2.38. Let $\mathcal{A}$ be an alphabet, A be a random variable distributed on $\mathcal{A}$, $\mathcal{B} \subseteq \mathcal{A}$ be a susbset such that $\Pr[A \in \mathcal{B}] > 0$, and $f : \mathcal{A} \to \mathbb{R}_0^+$ be any map. Finally, let B be the random variable distributed on $\mathcal{B}$ defined by $\Pr[B = b] = \Pr[A = b]/\Pr[A \in \mathcal{B}]$, i.e. B is simply A conditioned on $A \in \mathcal{B}$. Then

$$\left| \mathop{\mathbb{E}}_{a \leftarrow A} f(a) - \mathop{\mathbb{E}}_{a \leftarrow B} f(a) \right| \leq \max_{a \in \mathcal{A}} f(a) \cdot \Pr[A \notin \mathcal{B}]. \tag{2.122}$$

Proof: By definition of the expectation and our random variables, we find

$$\begin{aligned} \left| \mathop{\mathbb{E}}_{a \leftarrow A} f(a) - \mathop{\mathbb{E}}_{a \leftarrow B} f(a) \right| &= \left| \sum_{a \in \mathcal{A}} \Pr[A = a] \cdot f(a) - \sum_{a \in \mathcal{B}} \frac{\Pr[A = a]}{\Pr[A \in \mathcal{B}]} \cdot f(a) \right| \\ &= \left| \sum_{a \in \mathcal{A} \setminus \mathcal{B}} \Pr[A = a] \cdot f(a) - \sum_{a \in \mathcal{B}} \left(\frac{\Pr[A = a]}{\Pr[A \in \mathcal{B}]} - \Pr[A = a] \right) \cdot f(a) \right| \\ &= \left| \sum_{a \in \mathcal{A} \setminus \mathcal{B}} \Pr[A = a] \cdot f(a) - \sum_{a \in \mathcal{B}} (1 - \Pr[A \in \mathcal{B}]) \frac{\Pr[A = a]}{\Pr[A \in \mathcal{B}]} \cdot f(a) \right|. \end{aligned} \tag{2.123}$$

Since the codomain of f is $\mathbb{R}_0^+$, the last of these absolute values is the absolute value of the difference between two non-negative real values. Hence, the absolute value of the difference is upper-bounded by the maximum of these two values. Noting that

$$\begin{aligned} \sum_{a \in \mathcal{A} \setminus \mathcal{B}} \Pr[A = a] \cdot f(a) &\leq \sum_{a \in \mathcal{A} \setminus \mathcal{B}} \Pr[A = a] \cdot \max_{a \in \mathcal{A}} f(a) \\ &= \Pr[A \notin \mathcal{B}] \cdot \max_{a \in \mathcal{A}} f(a) \end{aligned} \tag{2.124}$$

and

$$\begin{aligned} \sum_{a \in \mathcal{B}} (1 - \Pr[A \in \mathcal{B}]) \frac{\Pr[A = a]}{\Pr[A \in \mathcal{B}]} \cdot f(a) &\leq \Pr[A \notin \mathcal{B}] \frac{\sum_{a \in \mathcal{B}} \Pr[A = a]}{\Pr[A \in \mathcal{B}]} \cdot \max_{a \in \mathcal{A}} f(a) \\ &= \Pr[A \notin \mathcal{B}] \cdot \max_{a \in \mathcal{A}} f(a) \end{aligned} \tag{2.125}$$

yields the desired result. ∎

With Lemma 2.37 and Lemma 2.38 in hand, we can now state and prove the security of the quantum money scheme $\mathrm{QM}_{\sqrt{\epsilon}}(S)$.

Theorem 2.39. Let $S = (K, E, D)$ be an ϵ-correct δ-tamper-evident AQECM scheme with messages $\mathcal{M}$. Then, the quantum money scheme $\mathrm{QM}_{\sqrt{\epsilon}}(S)$ is $\left(|\mathcal{M}|^{-1} + 4\delta + \sqrt{\epsilon}\right)$-secure.

Proof: First, we assume that $\epsilon < 1$. Otherwise, the result trivially holds since any quantum money scheme is δ'-secure for any $\delta' \geq 1$.

Now, note that the set

$$\mathcal{G} = \left\{(k, m) \in \mathcal{K} \times \mathcal{M} \ : \ \langle k| K(\varepsilon) |k\rangle > 0 \wedge \langle m| \overline{D}_k \circ E_k(m) |m\rangle \geq 1 - \sqrt{\epsilon}\right\} \tag{2.126}$$

used in the construction of $\mathrm{QM}_{\sqrt{\epsilon}}(S)$ is non-empty. Indeed, as the AQECM scheme S is assumed to be ϵ-correct, we have that

$$\mathop{\mathbb{E}}_{m \leftarrow \mathcal{M}} \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle m| \overline{D}_k \circ E(m) |m\rangle \geq 1 - \epsilon \tag{2.127}$$

which, by the concentration inequality of Lemma 2.2, implies that

$$\Pr_{\substack{k \leftarrow K(\varepsilon) \\ m \leftarrow \mathcal{M}}} [(k, m) \in \mathcal{G}] = \Pr_{\substack{k \leftarrow K(\varepsilon) \\ m \leftarrow \mathcal{M}}} \left[\langle m| \overline{D}_k \circ E(m) |m\rangle \geq 1 - \sqrt{\epsilon}\right] \geq 1 - \sqrt{\epsilon} > 0. \tag{2.128}$$

It follows that the quantum money scheme $\mathrm{QM}_{\sqrt{\epsilon}}(S) = (K', M', V')$ is constructed according to the non-empty $\mathcal{G}$ case of Definition 2.34. Recalling this definition, we see that for any counterfeiting attack $A' : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{C})$ and any $(k, m) \in \mathcal{K} \times \mathcal{M} = \mathcal{K}'$, we have that

$$\begin{aligned} \left(\overline{V'}_{(k,m)} \otimes \overline{V'}_{(k,m)}\right) \circ A' \circ M'(k \otimes m) &= \langle m, m| \left(\overline{D}_k \otimes \overline{D}_k\right) \circ A' \circ E_k(m) |m, m\rangle \\ &\leq \langle m| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \tilde{D}_k\right) \circ A' \circ E_k(m) |m\rangle \end{aligned} \tag{2.129}$$

where $\tilde{D}_k = (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D_k$ is the decoding channel which discards, without verifying, the accept/reject flag. At a high-level this inequality is obtained by noting that the probability that both instances of the map D_k accept *and* decode the correct message is no greater than the probability that the first accepts, possibly with the wrong message, and that the second decodes the correct message, possibly while rejecting. Mathematically, it follows from the definition of the trace and the fact that for any positive semidefinite operator ρ on the appropriate spaces, such as $\rho = (D_k \otimes D_k) \circ A \circ E_k(m)$, it holds that

$$\begin{aligned} \langle m, \mathtt{1}, m, \mathtt{1}| \rho |m, \mathtt{1}, m, \mathtt{1}\rangle &\leq \sum_{m' \in \mathcal{M}} \sum_{b' \in \{\mathtt{0},\mathtt{1}\}} \left\langle m', \mathtt{1}, m, b'\right| \rho \left|m', \mathtt{1}, m, b'\right\rangle \\ &= \langle \mathtt{1}, m| \left(\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{F}} \otimes \mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}\right)(\rho) |\mathtt{1}, m\rangle . \end{aligned} \tag{2.130}$$

By Lemma 2.37 and the fact that S is δ-tamper-evident, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{k\leftarrow K(\varepsilon)\\ m\leftarrow\mathcal{M}}} \langle m| \left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\tilde{D}_k\right)\circ A\circ E_k(m)\,|m\rangle &\leq \frac{1}{|\mathcal{M}|}(1-4\delta)+4\delta \\ &\leq \frac{1}{|\mathcal{M}|}+4\delta\end{aligned} \tag{2.131}$$

where, for simplicity, we neglect the $4\delta|\mathcal{M}|^{-1}$ term for the remainder of the proof.

We must now account for the possible deviation of the key generation procedures K' of $\mathrm{QM}_{\sqrt{\epsilon}}(S)$ from simply sampling m uniformly at random and k from $K(\varepsilon)$. For every key-message pair $(k,m)\in\mathcal{K}'=\mathcal{K}\times\mathcal{M}$, we have that

$$\langle k,m|\,K'(\varepsilon)\,|k,m\rangle = \begin{cases}\frac{\langle k,m|K(\varepsilon)|k,m\rangle}{\sum_{(k,m)\in\mathcal{G}}\langle k,m|K(\varepsilon)|k,m\rangle} & \text{if}\quad (k,m)\in\mathcal{G}\\ 0 & \text{else.}\end{cases} \tag{2.132}$$

Hence, by Lemma 2.38, we have that

$$\begin{aligned}&\mathop{\mathbb{E}}_{(k,m)\leftarrow K'(\varepsilon)} \langle m| \left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\tilde{D}_k\right)\circ A\circ E_k(m)\,|m\rangle \\ &\leq \mathop{\mathbb{E}}_{\substack{k\leftarrow K(\varepsilon)\\ m\leftarrow\mathcal{M}}} \langle m| \left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\tilde{D}_k\right)\circ A\circ E_k(m)\,|m\rangle + \left(1-\Pr_{\substack{k\leftarrow K(\varepsilon)\\ m\leftarrow\mathcal{M}}}[(k,m)\in\mathcal{G}]\right) \\ &\leq \frac{1}{\mathcal{M}}+4\delta+\left(1-\Pr_{\substack{k\leftarrow K(\varepsilon)\\ m\leftarrow\mathcal{M}}}[(k,m)\in\mathcal{G}]\right) \\ &\leq \frac{1}{\mathcal{M}}+4\delta+\sqrt{\epsilon}\end{aligned} \tag{2.133}$$

where the last inequality is obtained by Equation 2.128. This is the desired result. ∎

On avoiding sampling from $\mathcal{G}$. As we have previously mentioned, we do not consider here the complexity of sampling key-message pairs from $\mathcal{G}$. If the complexity of this sampling would be problematic, we can modify our construction of $\mathrm{QM}_{\gamma}(S)$ to avoid it. However, this comes at the cost of only achieving a weaker notion of correctness. Instead of achieving the notion of correctness we gave in Definition 2.32, namely that

$$\langle k|\,K(\varepsilon)\,|k\rangle > 0 \implies \overline{V}_k\circ M(k)\geq 1-\epsilon, \tag{2.134}$$

which is to say that *all* banknotes produced by the bank are accepted with high probability, we will only achieve

$$\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \overline{V}_k \circ M(k) \geq 1-\epsilon, \tag{2.135}$$

which is to say that banknotes are accepted, *on average*, with high probability. In particular, it may be possible for the bank to produce faulty banknotes which will not be accepted with any significant probability.

We denote our modified construction of a quantum money scheme from an AQECM scheme S by $\mathrm{QM}_{\mathrm{weak}}(S)$. This is exactly as the $\mathcal{G} \neq \varnothing$ case of $\mathrm{QM}_{\gamma}(S)$ in Definition 2.34, except that key-message pairs (k, m) are always generated by sampling k according to the key generation channel S and sampling m uniformly at random from the messages of S. If S is an ϵ-correct δ-tamper-evident AQECM scheme, then it is straightforward to see that the quantum money scheme $\mathrm{QM}_{\mathrm{weak}}(S)$ is ϵ-correct according with respect to the weaker definition of correctness given in Equation 2.135 and that it is $\left(|\mathcal{M}|^{-1} + 4\delta\right)$-secure by applying the proof of Theorem 2.39 but stopping at Equation 2.131.

2.6 Tamper Evidence, Revocation, and Certified Deletion

In this section, we explore the relation between tamper evidence and revocation schemes. A high-level overview of this section can be found in Section 2.1.2.3, our initial discussion in the introduction.

In Section 2.6.1 we review the syntax of a revocation scheme and we define the security of such schemes in Section 2.6.2. This includes our novel definition, the existing game-based definition, and proofs of the near equivalence of these two definitions. We give our construction of tamper-evident schemes from revocation schemes in Section 2.6.3.2 and our construction of revocation schemes from tamper-evident schemes in Section 2.6.3.1. Finally, in Section 2.6.4 we highlight that our results yield as an easy corollary that revocation implies encryption.

2.6.1 The Syntax of Revocation and Certified Deletion Schemes

Our first task in defining revocation and certified-deletion is to lay out the syntax for schemes aiming to achieve this security notion. We formalize this syntax in Definition 2.40 as *quantum encryptions of classical messages with revocation* (QECMR) schemes, or simply *revocation schemes*, and note that it is essentially the same definition as the one given for *certified deletion encryption* schemes by Broadbent and Islam [BI20], modulo the fact that we move the definition from a computational setting to an information-theoretic setting.

A QECMR (K, E, D, R, V) is a QECM (K, E, D) equipped with two additional channels R and V. The first, R, is the *revocation* channel and is unkeyed. This is the channel which the receiver of the original message should use if the message is to be revoked or forfeited. It processes a ciphertext into a state which we will call a *revocation token*. The second channel, V, is the *verification* channel and it is keyed. This channel processes a revocation token with the help of a key and produces a single qubit indicating whether it accepts or rejects the revocation.

Definition 2.40. A *quantum encoding of classical messages with revocation* (QECMR) is a tuple (K, E, D, R, V) of channels such that (K, E, D) is a QECM as given in Definition 2.11 and V and R are of the form

$$R : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{R}) \quad \text{and} \quad V : \mathcal{L}(\mathsf{K} \otimes \mathsf{R}) \to \mathcal{L}(\mathsf{F}) \tag{2.136}$$

for an alphabet $\mathcal{R}$ and Hilbert spaces $\mathsf{R} = \mathbb{C}^{\mathcal{R}}$ and $\mathsf{F} = \mathbb{C}^{\{0,1\}}$. For all keys $k \in \mathcal{K}$, we also define the map $\overline{V}_k : \mathcal{L}(\mathsf{R}) \to \mathbb{C}$ by $\rho \mapsto \langle 1 | V(k \otimes \rho) | 1 \rangle$.

The correctness of a QECMR is quantified by a single value $\epsilon \in \mathbb{R}$ which is an upper bound on both (1) the probability over the key generation process that any message is not correctly recovered after decryption and (2) the probability over the key generation process that any well formed revocation token is not accepted.[24] This is given in Definition 2.41.

Definition 2.41. Let $\epsilon \in \mathbb{R}$. A QECMR as given in Definition 2.40 is ϵ-correct if the QECM (K, E, D) is ϵ-correct and, for all messages $m \in \mathcal{M}$, we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \overline{V}_k \circ R \circ E_k(m) \geq 1 - \epsilon. \tag{2.137}$$

Certified deletion schemes as a subset of revocation schemes. We briefly comment on the distinction between a revocation scheme and a certified deletion scheme, but emphasize that this distinction is inconsequential for our work. Note that the previous definition does not require the revocation channel R to produce a classical output. While this property was not formally encoded in Broadbent and Islam's original definition [BI20], it was explicitly emphasized in the surrounding discussion.

A natural way to enforce the requirement that the revocation map produces a classical output would be to require that $R = \Delta_{\mathcal{R}} \circ R$, which is to say that the output of R never changes if it is immediately measured in the computational basis. This is *not* what we will do here. Instead, we make a similar requirement on the verification map V.

[24] One could quantify correctness for a QECMR with a *pair* of values, one bounding the probability of bad decryption and the other the probability of bad revocation. We do not need this level of granularity here.

Definition 2.42. A QECMR scheme as given in Definition 2.40 is a *certified deletion scheme* if $V = V \circ (\mathrm{Id}_{\mathsf{K}} \otimes \Delta_{\mathcal{R}})$.

We impose this condition on the *verification* map instead of the *revocation* map because, of the two, only the verification map appears in the upcoming security definitions. Hence, the condition "revocation tokens in a certified deletion scheme must be classical" needs to be encoded in the verification map for it to be relevant in the discussion of security. However, for honest users, there is no distinction between these two options. Indeed, if the verification map always begins with a computational basis measurement, the revocation map may as well output a classical state. Similarly, in this case, adversaries attempting to fool the verification map will gain no advantage in submitting a non-classical state. Mathematically, this is due to the fact that the computational basis measurement channel $\Delta_{\mathcal{R}}$ is idempotent, yielding, for any $k \in \mathcal{K}$, that $(V_k \circ \Delta_{\mathcal{R}}) \circ (\Delta_{\mathcal{R}} \circ R) = (V_k \circ \Delta_{\mathcal{R}}) \circ R = V_k \circ (\Delta_{\mathcal{R}} \circ R)$.

2.6.2 Defining Revocation Security

We give two definitions of revocation security for QECMR schemes and show that they are conceptually equivalent. The first, Definition 2.43, is a novel formulation which is well suited for the information-theoretic regime and closely resembles Gottesman's definition of tamper evidence. The second, Definition 2.46, is essentially the one given by Broadbent and Islam [BI20].

Our new definition inspired by tamper evidence. We begin by defining a *revocation* attack against an QECMR scheme. This is a channel which, on input of a ciphertext, aims to produce a revocation token which will be accepted while also keeping some information on the ciphertext.

Definition 2.43. A *revocation attack* against a QECMR scheme as given in Definition 2.40 is a channel of the form $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{R} \otimes \mathsf{A})$ for some Hilbert space A.

This is quite similar to the definition of a tamper attack $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$. The only difference is that the ciphertext space in the codomain is replaced with the revocation token space. Conceptually, there is also a close similarity: in both cases the adversary attempts to have an honest party accept a state while trying to obtain or keep information.

We now give our novel definition for revocation security, inspired by the definition of tamper evidence given in Definition 2.17. The only differences are that we substitute the decryption map of an AQECM with the verification map of a QECMR and that we bound the *expectation* of the trace distance over the choice of keys instead of requiring that it is small with high probability over the choice of the keys.

Definition 2.44. Let $\delta \in \mathbb{R}$ and let (K, E, D, R, V) be a QECMR scheme as given in Definition 2.40. This scheme has δ-revocation security if for any two messages $m, m' \in \mathcal{M}$ and any revocation attack A against it we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m - m') \right\|_1 \leq \delta. \tag{2.138}$$

The existing game-based definition of revocation security. We now recall the game-based definition of security for revocation, which is stated in Definition 2.46. This will formalize the idea of the following game played between an adversary and a referee:

1. The adversary prepares a tripartite state over the space $\mathsf{M} \otimes \mathsf{M} \otimes \mathsf{A}$. This should be understood as the adversary preparing two candidate messages m_0 and m_1 on the space $\mathsf{M} \otimes \mathsf{M}$ as well as keeping a memory state on the space A. The adversary then sends the $\mathsf{M} \otimes \mathsf{M}$ subsystem to the referee.

2. The referee samples a key $k \leftarrow K(\varepsilon)$, samples uniformly at random a bit $b \leftarrow \{0, 1\}$, and measures the received states on the $\mathsf{M} \otimes \mathsf{M}$ space in the computational basis to obtain a result (m_0, m_1). Then, they return $E_k(m_b)$ to the adversary.

3. Acting on their memory state and the $E_k(m_b)$ state received from the referee, the adversary produces a bipartite state on $\mathsf{R} \otimes \mathsf{A}$ and sends the R subsystem to the referee. This should be understood as the adversary attempting to convince the referee that the ciphertext has been revoked while keeping some information.

4. The referee verifies the state returned by the adversary by applying V_k and measuring the resulting qubit in the computational basis. Call the result v.

5. The referee divulges the key k to the adversary, after which the adversary produces a guess $g \in \{0, 1\}$ for the value of b.

Conceptually, we understand the adversary to win if and only if they correctly guess the value of b *and* the referee accepted the revocation. In other words, the adversary wins if and only if $v = 1$ and $g = b$. We could then define the security of this scheme as the advantage of the adversary in winning this game beyond the trivially achievable probability of $\frac{1}{2}$.

Instead, we take a slightly different approach and simply require that the adversary's output g does not depend much on b when $v = 1$. Formally, the scheme achieves δ-game-revocation security if the absolute value between $\mathbb{E}_{k \leftarrow K(\varepsilon)} \Pr[v = 1 \wedge g = 1 | b = 0]$ and $\mathbb{E}_{k \leftarrow K(\varepsilon)} \Pr[v = 1 \wedge g = 1 | b = 1]$, where the probability is taken over one execution of this game with the sampled key k, is at most δ. This is represented in Table 2.1.

	Ref. Sampled $b = 0$		Ref. Sampled $b = 1$	
	Ref. Rej. ($v = 0$)	Ref. Acc. ($v = 1$)	Ref. Rej. ($v = 0$)	Ref. Acc. ($v = 1$)
Adv. Guessed $g = 0$				
Adv. Guessed $g = 1$		p_0		p_1

Table 2.1: Eight possible scenarios in a revocation game. For both values of $b' \in \{0, 1\}$, we let $p_{b'} = \Pr[g = 1 \wedge v = 1 | b = b']$ and identify these events in the table.

We formalize this game and security statement in the following two definitions. In order to differentiate from our previously defined notions of revocation attacks and revocation security, we will use the terms *game-revocation attack* and *game-revocation security* here.

Definition 2.45. Let (K, E, D, V, R) be a QECMR as given in Definition 2.40. A *game-revocation attack* against this scheme is a triplet of channel (A^0, A^1, A^2) of the form

$$\begin{aligned} A^0 &: \mathcal{L}(\mathbb{C}) \to \mathcal{L}(\mathsf{M} \otimes \mathsf{M} \otimes \mathsf{A}) \\ A^1 &: \mathcal{L}(\mathsf{C} \otimes \mathsf{A}) \to \mathcal{L}(\mathsf{R} \otimes \mathsf{A}) \\ A^2 &: \mathcal{L}(\mathsf{K} \otimes \mathsf{A}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}}) \end{aligned} \tag{2.139}$$

for some Hilbert space A.

For every key $k \in \mathcal{K}$, we also define the map $A^2_k : \mathcal{L}(\mathsf{A}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}})$ by $\rho \mapsto A^2(k \otimes \rho)$.

Note that A^0, A^1, and A^2 model, respectively, the actions taken by the adversary in steps 1, 3, and 5 of the game we described.

Definition 2.46. Let $S = (K, E, D, V, R)$ be a QECMR scheme as given in Definition 2.40. For all keys $k \in \mathcal{K}$, we define the channels

$$E^0_k = E_k \circ (\Delta_{\mathcal{M}} \otimes \mathrm{Tr}_{\mathsf{M}}) \quad \text{and} \quad E^1_k = E_k \circ (\mathrm{Tr}_{\mathsf{M}} \otimes \Delta_{\mathcal{M}}) \tag{2.140}$$

which is to say that for both $b \in \{0, 1\}$, E^b_k measures its input in the computational basis to obtain an outcome (m_0, m_1) and then outputs $E_k(m_b)$.

The scheme S has *δ-game-revocation security* if for all game-revocation attacks (A^0, A^1, A^2) against it we have that

$$\left| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle 1 | \, A^2_k \circ \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A^1 \circ \left(\left(E^0_k - E^1_k\right) \otimes \mathrm{Id}_A\right) \circ A^0(\varepsilon) \, | 1 \rangle \right| \leq \delta. \tag{2.141}$$

This is essentially the definition of certified deletion given by Broadbent and Islam [BI20] and of revocation given by Unruh [Unr15]. The main difference is that we do not

state this definition in an asymptotic regime; instead of requiring the adversaries to have an advantage at most negligible in a security parameter, we say that the scheme achieves δ-security if their advantage is at most δ. We also differ from Broadbent and Islam as we require the adversary to initially submit *two* messages. Their definition only considered cases where the message space was the set of bit strings of a fixed length and fixed the second message which the referee could use to be the all-zero bit string. In our setting, where we do not impose such restrictions on the message space, it is more natural to let the adversary submit two messages than to have a definition which depends on the referee's alternative message.

Another game-based definition. A slightly different game-based definition of revocation security has also been given in the work of Hiroka, Morimae, Nishimaki, and Yamakawa [HMNY21] on certified deletion. In their definition, the adversary does not need the referee to accept the revocation token to win the game. However, the adversary is given the key in step 5 if and only if the referee accepts this token. If the referee does not accept, then the adversary must generate their guess g without knowledge of the key k. Evidently, a QECMR scheme achieving a good level of revocation security with respect to the game of Hiroka *et al.* must also be a good encryption scheme. Indeed, if the scheme was not a good encryption scheme, then an adversary could distinguish the two ciphertexts without needing the key, and hence would be able to win the game without even trying to obtain the key. Thus, the Hiroka *et al.* definition has the advantages of covering two security notions, encryption and revocation, in a single security game and may better reflect certain use cases. However, it does have certain theoretical drawbacks.

One such drawback is that analysing the revocation security of a QECMR scheme with respect to this definition is a bit more involved than with respect to Definition 2.46. Indeed, we must now consider the adversary's guessing strategy in two very distinct cases: in the case where they do get the key, and in the case where they do not.

A second drawback is that this definition ties together two security notions, encryption and revocation, which *a priori* are distinct. For example, not every encryption scheme allows revocation. This grouping of encryption and revocation hinders a more fine-grained study of these security notions. For example, is it possible to achieve revocation without encryption? With respect to their definition, the answer is trivially "no". On the other hand, this is a non-trivial question with respect to Definition 2.46, even if we show later in Section 2.6.4 that the answer remains "no". We believe that it is theoretically more interesting to know that revocation security implies encryption by necessity, and not simply by definition.

The near equivalence of these two definitions. As we have introduced in Definition 2.44 a new formulation of revocation security, we have a responsibility to show how it relates to the existing definition. We show that there is at most a factor of two difference between the revocation security and game-revocation security achieved by any given QECMR scheme. More precisely, we show in Theorem 2.47 that δ-revocation implies 2δ-game-revocation and in Theorem 2.48 that δ-game-revocation implies δ-revocation. In other words, if a scheme has δ_{r}-revocation and δ_{gr}-game-revocation security, then

$$\delta_{\mathrm{r}} \leq \delta_{\mathrm{gr}} \leq 2\delta_{\mathrm{r}} \quad \text{or, equivalently,} \quad \frac{1}{2}\delta_{\mathrm{gr}} \leq \delta_{\mathrm{r}} \leq \delta_{\mathrm{gr}}. \tag{2.142}$$

This factor of two between upper and lower bounds may be a proof artefact. However, it seems to be closely related to the fact that the inequality $|\langle 1| \Phi(\rho - \sigma) |1\rangle| \leq \frac{1}{2}\|\rho - \sigma\|_1$ for a channel Φ may not hold if ρ or σ are subnormalized operators. In the discussion surrounding Lemma 2.3, we noted that the correct inequality in this case is $|\langle 1| \Phi(\rho - \sigma) |1\rangle| \leq \|\rho - \sigma\|_1$, namely that the upper bound must be multiplied by a factor of two to hold.

Theorem 2.47. A QECMR with δ-revocation security has 2δ-game-revocation security.

Proof: Let (K, E, D, V, R) be a QECMR as given in Definition 2.40 and let (A^0, A^1, A^2) be a game-revocation attack against this scheme as given in Definition 2.45. The first step of this proof is to eliminate the roles of the A^0 and A^2 channels and identify the remaining A^1 channel with a revocation attack as considered in Definition 2.44.

Recall that the quantity to bound is

$$\left| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle 1| A_k^2 \circ (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \circ \left((E_k^0 - E_k^1) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A^0(\varepsilon) |1\rangle \right|. \tag{2.143}$$

We begin by eliminating A^2. By the triangle inequality and Lemma 2.3, which tells us that $|\langle 1| \Phi(A - B) |1\rangle| \leq \|A - B\|_1$ if Φ is a channel and $A - B$ is Hermitian, we have that

$$\begin{aligned} &\left| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle 1| A_k^2 \circ (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \circ \left((E_k^0 - E_k^1) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A^0(\varepsilon) |1\rangle \right| \\ &\leq \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left| \langle 1| A_k^2 \circ (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \circ \left((E_k^0 - E_k^1) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A^0(\varepsilon) |1\rangle \right| \\ &\leq \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\| (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \circ \left((E_k^0 - E_k^1) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A^0(\varepsilon) \right\|_1. \end{aligned} \tag{2.144}$$

Next, we eliminate A^0. By definition, we have that

$$\left((E_k^0 - E_k^1) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A^0 = E_k \circ ((\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{M}}) - (\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{M}})) \circ (\Delta_{\mathcal{M}} \otimes \Delta_{\mathcal{M}}) \circ A^0 \tag{2.145}$$

and we can write

$$(\Delta_{\mathcal{M}} \otimes \Delta_{\mathcal{M}}) \circ A^0(\varepsilon) = \sum_{m,m' \in \mathcal{M}} p_{m,m'} \cdot m \otimes m' \otimes \sigma_{m,m'} \tag{2.146}$$

where the $p_{m,m'}$ terms are non-negative reals which sum to 1 and $\sigma_{m,m'} \in \mathcal{D}(\mathsf{A})$ is the state of the adversarial memory register A if (m, m') is the result of the measurement. Thus,

$$\left(\left(E_k^0 - E_k^1\right) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A^0(\varepsilon) = \sum_{m,m' \in \mathcal{M}} p_{m,m'} \cdot E_k(m - m') \otimes \sigma_{m,m'} \tag{2.147}$$

and so

$$\begin{aligned} &\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\| (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \circ \left(\left(E_k^0 - E_k^1\right) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A^0(\varepsilon) \right\|_1 \\ &= \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\| \sum_{m,m' \in \mathcal{M}} p_{m,m'} (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \left(\left(E_k(m - m')\right) \otimes \sigma_{m,m'}\right) \right\|_1 . \end{aligned} \tag{2.148}$$

Using the triangle inequality, we can upper bound the values of the previous equation by

$$\begin{aligned} &\sum_{m,m' \in \mathcal{M}} p_{m,m'} \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\| (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A^1 \left(\left(E_k(m - m')\right) \otimes \sigma_{m,m'}\right) \right\|_1 \\ &= \sum_{m,m' \in \mathcal{M}} p_{m,m'} \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\| (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A'_{m,m'} \circ E_k(m - m') \right\|_1 \end{aligned} \tag{2.149}$$

where, for every $m, m' \in \mathcal{M}$, we define the channel $A'_{m,m'}$ by $\rho \mapsto A^1(\rho \otimes \sigma_{m,m'})$. Note that the channel $A'_{m,m'}$ is a revocation attack. As we assume that the QECMR scheme has δ-revocation security, we have by definition that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A'_{m,m'} \circ E_k(m - m') \right\|_1 \leq \delta \tag{2.150}$$

for all $m, m' \in \mathcal{M}$ and so

$$\sum_{m,m'} p_{m,m'} \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \left\| (\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}) \circ A'_{m,m'} \circ E_k(m - m') \right\|_1 \leq \sum_{m,m'} p_{m,m'} 2\delta = 2\delta \tag{2.151}$$

which, with Equation 2.144, yields the desired result. ∎

We now move on to show that δ-game-revocation security implies δ-revocation security. We do this by transforming an revocation attack A and pair of messages (m, m') into a game-revocation attack (A^0, A^1, A^2). Essentially, A^0 simply generates the state $m \otimes m'$ as the

candidate messages for the referee, A^1 is taken to be A, and A^2 is an optimal distinguishing channel for the two possible states held by the adversary. For technical reasons, namely to saturate the first bound in Lemma 2.3, we also need to consider a closely related game-revocation attack $(A^0, A^1, A^2_{\text{Flip}})$ which simply flips the guess of A^2.

Theorem 2.48. A QECMR with δ-game-revocation security has δ-revocation security.

Proof: Let (K, E, D, V, R) be a QECMR as given in Definition 2.40. For any revocation attack $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{R} \otimes \mathsf{A})$ and any two messages $m, m' \in \mathcal{M}$, we define the game-revocation attack (A^0, A^1, A^2) as follows:

- The channel $A^0 : \mathbb{C} \to \mathcal{L}(\mathsf{C} \otimes \mathsf{C} \otimes \mathsf{A})$ is defined by $c \mapsto c\left(m \otimes m' \otimes \frac{I_{\mathsf{A}}}{\dim \mathsf{A}}\right)$.

  The final term in this tensor product is there merely because the syntax requires A^0 to include the A space in its output. It will be promptly deleted and replaced by A^1.

- The channel $A^1 : \mathcal{L}(\mathsf{C} \otimes \mathsf{A}) \to \mathcal{L}(\mathsf{R} \otimes \mathsf{A})$ is defined by $A^1 = A \circ (\mathrm{Id}_{\mathsf{C}} \otimes \mathrm{Tr}_{\mathsf{A}})$, which is to say that it simply applies the revocation attack A.

- The channel $A^2 : \mathcal{L}(\mathsf{K} \otimes \mathsf{A}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}})$ is any channel which satisfies

$$\begin{aligned} \sum_{b \in \{0,1\}} \left| \langle b | A^2 \left(\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} k \otimes \overline{V}_k \circ A \circ E_k(m - m') \right) |b\rangle \right| \\ = \left\| \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} k \otimes \overline{V}_k \circ A \circ E_k(m - m') \right\|_1 \end{aligned} \tag{2.152}$$

  which is to say that A^2 is an optimal channel to distinguish between the possible outputs of A^1 when given the key k. Such a channel exists by Lemma 2.3.

We also define $A^2_{\text{Flip}} : \mathcal{L}(\mathsf{K} \otimes \mathsf{C}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}})$ by $\rho \mapsto (|1\rangle\langle 0| + |0\rangle\langle 1|)A^2(\rho)(|1\rangle\langle 0| + |0\rangle\langle 1|)$, which is to say that A^2_{Flip} is simply A^2 followed by a bit flip operation. In particular, it holds that $\langle 1| A^2_{\text{Flip}}(\rho) |1\rangle = \langle 0| A^2(\rho) |0\rangle$ for any linear operator ρ. Note that $(A^0, A^1, A^2_{\text{Flip}})$ is also a game-revocation attack. We examine the performance of both these attacks.

By the definition of A^0 and A^1, we have that

$$A^1 \circ \left((E^0_k - E^1_k) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A^0(\varepsilon) = A \circ E_k(m - m') \tag{2.153}$$

and so, summing over both attacks, we see that

$$\begin{aligned}
&\sum_{C\in\{A^2,A^2_{\mathrm{Flip}}\}} \left| \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \langle 1| C_k \circ \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A^1 \circ \left((E^0_k - E^1_k) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A^0(\varepsilon) |1\rangle \right| \\
&= \sum_{C\in\{A^2,A^2_{\mathrm{Flip}}\}} \left| \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \langle 1| C_k \circ \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m-m') |1\rangle \right| \\
&= \sum_{C\in\{A^2,A^2_{\mathrm{Flip}}\}} \left| \langle 1| C \left(\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} k \otimes \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m-m') \right) |1\rangle \right| \\
&= \sum_{b\in\{0,1\}} \left| \langle b| A^2 \left(\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} k \otimes \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m-m') \right) |b\rangle \right| \\
&= \left\| \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} k \otimes \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m-m') \right\|_1 \\
&= \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m-m') \right\|_1
\end{aligned} \tag{2.154}$$

where the first three equalities follow from the definitions of the various maps, the fourth from the defining property of A^2, and the fifth from Lemma 2.6 which states a technical property of the Schatten 1-norm.

As the scheme is assumed to be δ-game-revocation secure, both of the initial absolute values is upper-bounded by δ. Hence, $\mathbb{E}_{k\leftarrow K(\varepsilon)} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m-m') \right\|_1 \leq 2\delta$ which yields the desired result after dividing both sides by two. ∎

2.6.3 From Revocation to Tamper Evidence and Back

We show here how to construct a revocation scheme from a tamper-evident scheme (Section 2.6.3.1) and vice-versa (Section 2.6.3.2). The intuition for these constructions can be conveyed by recalling and directly comparing the inequalities defining the two relevant security guarantees.

As stated in Definition 2.17, an AQECM scheme is δ-tamper-evident if and only if for all channels $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ and all messages $m, m' \in \mathcal{M}$ we have that

$$\Pr_{k\leftarrow K(\varepsilon)} \left[\frac{1}{2} \left\| \left((\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m-m') \right\|_1 \leq \delta \right] \geq 1 - \delta. \tag{2.155}$$

As stated in Definition 2.44, a QECMR scheme has δ-revocation security if and only if for

all channels $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{R} \otimes \mathsf{A})$ and all messages $m, m' \in \mathcal{M}$ we have that

$$\mathop{\mathbb{E}}_{k \leftarrow \kappa} \frac{1}{2} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m - m') \right\|_1 \leq \delta. \tag{2.156}$$

The striking similarity between these two security guarantees is that they are both statements on the trace distance between subnormalized states held by an adversary, conditioned on an honest receiver accepting what the adversary gave them.

We acknowledge one difference, namely that revocation security is expressed as a bound on the *expected* trace distance while tamper evidence states that with *high probability over the choice of keys* the trace distance is small. However, using the concentration inequality given in Lemma 2.2, we can shift between both types of statements with ease.

2.6.3.1 Revocation From Tamper Evidence

We show here how to construct a QECMR which is ϵ-correct and has 2δ-revocation security from any ϵ-correct δ-tamper-evident AQECM.

Our construction, which we denote $S \mapsto \mathrm{Rev}(S)$, is quite simple. It suffices to note that in an AQECM scheme the decoding and verification of a ciphertext are done by the same channel, but that in a QECMR these tasks are accomplished by different channels. Thus, we can obtain the decoding and revocation maps of the QECMR by simply using the decoding map of the AQECM and keeping only the appropriate output. Finally, the revocation map is the identity, which is to say that the original recipient simply returns the ciphertext unmodified during the revocation protocol. This is formalized in Definition 2.49.

Definition 2.49. Let $S = (K, E, D)$ be an AQECM scheme as given in Definition 2.13. We define $\mathrm{Rev}(S) = (K', E', D', R', V')$ by $K' = K$, $E' = E$, $R = \mathrm{Id}_{\mathsf{C}}$, as well as

$$D' = (\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D, \quad \text{and} \quad V' = (\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{F}}) \circ D. \tag{2.157}$$

Note that $\mathrm{Rev}(S)$ is a QECMR with the same keys and messages as S.

We show in the following theorem that $\mathrm{Rev}(S)$ inherits the correctness of S.

Theorem 2.50. If S is an ϵ-correct AQECM, then $\mathrm{Rev}(S)$ is an ϵ-correct QECMR.

Proof: Let S be an AQECM as given in Definition 2.13 and let $\mathrm{Rev}(S)$ be a QECMR as

given in Definition 2.49. For any message $m \in \mathcal{M}$, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{k\leftarrow K'(\varepsilon)} \langle m| D'_k \circ E'_k(m) |m\rangle &= \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \langle m| (\mathrm{Id}_{\mathsf{M}} \circ \mathrm{Tr}_{\mathsf{F}}) \circ D_k \circ E_k(m) |m\rangle \\ &\geq \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \langle m| \overline{D}_k \circ E_k(m) |m\rangle \\ &\geq 1-\epsilon \end{aligned} \tag{2.158}$$

where the equality follows from the definition of $\mathrm{Rev}(S)$, the first inequality by replacing the trace with the map $\rho \mapsto \langle 1| \rho |1\rangle$, and the second by the assumption that S is ϵ-correct. Similarly, for any message $m \in \mathcal{M}$, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{k\leftarrow K'(\varepsilon)} \overline{V'}_k \circ R \circ E'_k(m) &= \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \circ E_k(m) \\ &\geq \mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \langle m| \overline{D}_k \circ E_k(m) |m\rangle \\ &\geq 1-\epsilon. \end{aligned} \tag{2.159}$$

Thus, we can conclude that $\mathrm{Rev}(S)$ is an ϵ-correct QECMR scheme. ∎

Finally, we show that if S is δ-tamper evident, then $\mathrm{Rev}(S)$ has 2δ-revocation security. The factor of two loss in security is incurred by passing from a concentration inequality in the definition of tamper evidence to an expectation in the definition of revocation security, as illustrated in Lemma 2.21.

Theorem 2.51. $\mathrm{Rev}(S)$ has 2δ-revocation security if S is a δ-tamper-evident AQECM.

Proof: We can assume that $0 \leq \delta < \frac{1}{2}$ as the result result is trivially true if $\delta \geq \frac{1}{2}$ and there are no δ-tamper-evident schemes if $\delta < 0$.

By definition of $\mathrm{Rev}(S)$, specifically that $R' = \mathrm{Id}_{\mathsf{C}}$, revocation tokens are simply ciphertexts from S. This implies that the Hilbert space R is equal to C. Thus, any revocation attack $A : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{R} \otimes \mathsf{A})$ against $\mathrm{Rev}(S)$ is also a tamper attack against S.

Since S is δ-tamper-evident, for any two messages $m, m' \in \mathcal{M}$ it holds by Lemma 2.21 that

$$\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k \right) \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E_k(m-m') \right\|_1 \leq 2\delta - \delta^2 \tag{2.160}$$

which implies, after substituting in the channels of $\mathrm{Rev}(S)$, that

$$\mathop{\mathbb{E}}_{k\leftarrow K'(\varepsilon)} \frac{1}{2} \left\| \left(\overline{V'}_k \otimes \mathrm{Id}_{\mathsf{A}} \right) \circ A \circ E'_k(m-m') \right\|_1 \leq 2\delta - \delta^2 \tag{2.161}$$

which is sufficient to obtain that $\mathrm{Rev}(S)$ has 2δ-revocation security. ∎

2.6.3.2 Tamper Evidence from Revocation

We show here how to construct an AQECM which is $2\sqrt[4]{\epsilon}$-correct and $\sqrt{\delta}$-tamper evident from any QECMR which is ϵ-correct and has δ-revocation security. We denote the construction $S \mapsto \mathrm{TE}(S)$.

The main idea of this construction is for the recipient in the tamper-evident setting to try to detect meaningful eavesdropping by running the complete revocation protocol of the original revocation scheme, namely $V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R)$, on their side upon reception of the ciphertext. They accept the ciphertext if and only if this revocation protocol accepts. This leads to an obvious problem: how can they then recover the plaintext if they have run the revocation protocol?

The solution is for the recipient to actually use the *coherent gentle measurement* version of $V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R)$, namely $\mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R))$ as described in Definition 2.8. The resulting channel produces as output a state on the ciphertext space as well as a single qubit flag where the resulting ciphertext state should be quite close to the original one, provided that it was very likely for the revocation protocol to accept. The recipient can then subsequently recover the plaintext by applying the decryption map of the QECMR scheme to this ciphertext state.

This construction is formalized in the following definition.

Definition 2.52. Let $S = (K, E, D, V, R)$ be a QECMR scheme as given in Definition 2.40. We define the AQECM $\mathrm{TE}(S) = (K', E', D')$ by

$$K' = K, \qquad E' = E, \quad \text{and} \quad D' = (D \otimes \mathrm{Id}_{\mathsf{F}}) \circ \mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R)). \tag{2.162}$$

We now show the correctness of $\mathrm{TE}(S)$.

Theorem 2.53. Let S be an ϵ-correct QECMR. Then, $\mathrm{TE}(S)$ is a $2\sqrt[4]{\epsilon}$-correct AQECM

Proof: Let $S = (K, E, D, V, R)$ be an ϵ-correct QECMR as given in Definition 2.40. We assume that $0 \leq \epsilon < \frac{1}{16}$ as there is no QECMR which is ϵ'-correct if $\epsilon' < 0$ and the result is trivially true if $\epsilon \geq \frac{1}{16}$. Let $\mathrm{TE}(S) = (K', E', D')$ be as given in Definition 2.52.

Recall that correctness for a QECMR considers the correctness of decoding the proper message and the correctness of accepting an honest revocation separately. In particular, there is *a priori* no guarantee that a message m which correctly encodes and decodes with the key k will have its revocation with the same key accepted with high probability. However, correctness for an AQECM considers both the correct decoding and the acceptance simultaneously.

The fact that the QECMR S is ϵ-correct means that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle 1 | V_k \circ R \circ E_k(m) | 1 \rangle \geq 1 - \epsilon \quad \text{and} \quad \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \langle m | D_k \circ E_k(m) | m \rangle \geq 1 - \epsilon \tag{2.163}$$

for any message $m \in \mathcal{M}$. With the concentration inequality of Lemma 2.2 and the union bound, this implies that for any $m \in \mathcal{M}$ it holds that

$$\Pr_{k \leftarrow K(\varepsilon)} \left[\begin{array}{c} \langle 1| V_k \circ R \circ E_k(m) |1\rangle \geq 1 - \sqrt{\epsilon} \\ \wedge \\ \langle m| D_k \circ E_k(m) |m\rangle \geq 1 - \sqrt{\epsilon} \end{array} \right] \geq 1 - 2\sqrt{\epsilon}. \tag{2.164}$$

Let $\mathcal{G}_m \subseteq \mathcal{K}$ be the set of keys satisfying the conditions in this probability statement. Note that $1 - 2\sqrt{\epsilon} > 0$ as $\epsilon < 1/4$ and so $\mathcal{G}_k \neq \varnothing$. Note that this set depends, in general, on m.

By definition of E' and D', we have for any message $m \in \mathcal{M}$ and key $k \in \mathcal{K}$ that

$$\langle m| \overline{D'}_k \circ E'_k(m) |m\rangle = \left(\langle m| \otimes \langle 1| \right) (D \otimes \mathrm{Id}_{\mathsf{F}}) \circ \mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R)) \left(k \otimes E_k(m) \right) \left(|m\rangle \otimes |1\rangle \right). \tag{2.165}$$

We begin by examining $\mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R))(k \otimes E_k(m))$ under the assumption that $k \in \mathcal{G}_m$. Under this assumption, we have that

$$\langle 1| \left(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R) \right)(k \otimes E_k(m)) |1\rangle = \langle 1| V_k \circ R \circ E_k(m) |1\rangle \geq 1 - \sqrt{\epsilon} \tag{2.166}$$

and so, by the properties of the CGM map established in Theorem 2.10, it follows that

$$\frac{1}{2} \| \mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R))(k \otimes E_k(m)) - k \otimes E_k(m) \otimes |1\rangle\langle 1| \|_1 \leq \sqrt{1 - (1 - \sqrt{\epsilon})^2} = \sqrt{2\sqrt{\epsilon} - \epsilon}. \tag{2.167}$$

We can substitute $\mathrm{CGM}(V \circ (\mathrm{Id}_K \otimes R))(k \otimes E_k(m))$ with the state $k \otimes E_k(m) \otimes |1\rangle\langle 1|$ in the rigth-hand side of Equation 2.165 and its value will change by at most $\sqrt{2\sqrt{\epsilon} - \epsilon}$.[25] Thus,

$$\begin{aligned} & \left(\langle m| \otimes \langle 1| \right) (D \otimes \mathrm{Id}_{\mathsf{F}}) \circ \mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R)) \left(k \otimes E_k(m) \right) \left(|m\rangle \otimes |1\rangle \right) \\ & \geq \left(\langle m| \otimes \langle 1| \right) (D \otimes \mathrm{Id}_{\mathsf{F}}) \left(k \otimes E_k(m) \otimes |1\rangle\langle 1| \right) \left(|m\rangle \otimes |1\rangle \right) - \sqrt{2\sqrt{\epsilon} - \epsilon} \\ & = \langle m| D(k \otimes E_k(m)) |m\rangle - \sqrt{2\sqrt{\epsilon} - \epsilon} \\ & \geq 1 - \sqrt{\epsilon} - \sqrt{2\sqrt{\epsilon} - \epsilon} \end{aligned} \tag{2.168}$$

where the final inequality is obtained by the assumption that $\langle m| D_k \circ E_k(m) |m\rangle \geq 1 - \sqrt{\epsilon}$ since $k \in \mathcal{G}_m$.

[25] That the change in the inner product following such a substitution can be controlled by the trace distance is well known. It can be recovered as a corollary to the second remark after Lemma 2.3 and considering, in this case, the channel $\Phi = \mathrm{Ms}(\mu)$ where $\mu = \{0, 1\} \to \mathrm{Pos}(\mathsf{M} \otimes \mathbb{C}^{\{0,1\}})$ is the measurement defined by $\mu(1) = |m\rangle\langle m| \otimes |1\rangle\langle 1|$ and $\mu(0) = I_{\mathsf{M} \otimes \mathbb{C}^{\{0,1\}}} - \mu(0)$.

Finally, we have to account for the cases where $k \notin \mathcal{G}_m$. We have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{k\leftarrow K'(\varepsilon)} \langle m| \overline{D'}_k \circ E'_k(m) |m\rangle &\geq \Pr_{k\leftarrow K(\varepsilon)}[k \in \mathcal{G}_m] \cdot \left(1 - \sqrt{\epsilon} - \sqrt{2\sqrt{\epsilon} - \epsilon}\right) \\
&\geq (1 - 2\sqrt{\epsilon}) \left(1 - \sqrt{\epsilon} - \sqrt{2\sqrt{\epsilon} - \epsilon}\right) \\
&= 1 - \sqrt[4]{\epsilon} \underbrace{\left((1 - 2\sqrt{\epsilon})\sqrt{2 - \sqrt{\epsilon}} - 2\epsilon^{3/4} + 3\sqrt[4]{\epsilon}\right)}_{=g(\epsilon)} \\
&\geq 1 - 2\sqrt[4]{\epsilon}
\end{aligned} \tag{2.169}$$

where the factor of 2 results from maximizing $g(\epsilon)$ over all values $0 \leq \epsilon \leq \frac{1}{16}$. Noting that this holds for all $m \in \mathcal{M}$ completes the proof. ∎

We now show that if S has δ-revocation security, then $\mathrm{TE}(S)$ is $\sqrt{\delta}$-tamper evident. We begin with a technical lemma making explicit a relation between the maps of these schemes.

Lemma 2.54. Let $S = (K, E, D, V, R)$ be a QECMR scheme as given in Definition 2.40 and let $S' = \mathrm{TE}(S) = (K', E', D')$ be an AQECM as given in Definition 2.52. Then, for any key $k \in \mathcal{K}$, we have that

$$\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_k = \overline{V}_k \circ R. \tag{2.170}$$

Proof: Let $\rho \in \mathcal{L}(\mathsf{C})$ be a linear operator. Then, by definition of D', we have that

$$\begin{aligned}
\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_k(\rho) &= \mathrm{Tr}_{\mathsf{M}} \left((I_{\mathsf{M}} \otimes \langle 1|) D'(k \otimes \rho) (I_{\mathsf{M}} \otimes |1\rangle) \right) \\
&= \mathrm{Tr}_{\mathsf{M}} \left((I_{\mathsf{M}} \otimes \langle 1|) (D \otimes \mathrm{Id}_{\mathsf{F}}) \circ \mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R))(k \otimes \rho) (I_{\mathsf{M}} \otimes |1\rangle) \right)
\end{aligned} \tag{2.171}$$

and if we let $C_{\langle 1|} : \mathcal{L}(\mathsf{F}) \to \mathbb{C}$ be defined by $\sigma \mapsto \langle 1| \sigma |1\rangle$, we find that

$$(\mathrm{Tr}_{\mathsf{M}} \otimes C_{\langle 1|}) \circ (D \otimes \mathrm{Id}_{\mathsf{F}}) = (\mathrm{Tr}_{\mathsf{M}} \circ D) \otimes C_{\langle 1|} = \mathrm{Tr}_{\mathsf{K}\otimes\mathsf{C}} \otimes C_{\langle 1|} = C_{\langle 1|} \circ (\mathrm{Tr}_{\mathsf{K}\otimes\mathsf{C}} \otimes \mathrm{Id}_{\mathsf{F}})$$

which allows us to continue the previous equation as

$$= \langle 1| (\mathrm{Tr}_{\mathsf{K}\otimes\mathsf{C}} \otimes \mathrm{Id}_{\mathsf{F}}) \circ \mathrm{CGM}(V \circ (\mathrm{Id}_{\mathsf{K}} \otimes R))(k \otimes \rho) |1\rangle \tag{2.172}$$

which, by properties of the CGM map exhibited in Theorem 2.10, allows us to further continue as

$$\begin{aligned} &= \langle 1| V \circ (\mathrm{Id}_k \otimes R)\,(k \otimes \rho)\,|1\rangle \\ &= \langle 1| V_k \circ R(\rho)\,|1\rangle \end{aligned} \tag{2.173}$$

which is precisely $\overline{V}_k \circ R(\rho)$. Since this equality holds for all linear operators ρ, we obtain the desired result that the maps are equal. ∎

With this lemma in hand, we now proceed to the theorem proper.

Theorem 2.55. If S is a δ-revocation secure QECMR, then $\mathrm{TE}(S)$ is $\sqrt{\delta}$-tamper evident.

Proof: Let $S = (K, E, D, V, R)$ be a QECMR scheme as given in Definition 2.40 which has δ-revocation security and let $\mathrm{TE}(S) = (K', E', D')$ be as given in Definition 2.52.

Let $A' : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{A})$ be a tamper attack against $\mathrm{TE}(S)$. Then, $A = (R \otimes \mathrm{Id}_{\mathsf{A}}) \circ A'$ is a revocation attack against S. Since S has δ-revocation security, for any two messages $m, m' \in \mathcal{M}$ we have that

$$\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m - m') \right\|_1 \leq \delta \tag{2.174}$$

which, as $\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_k = \overline{V}_k \circ R$, implies that

$$\begin{aligned} &\mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left((\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_k) \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A' \circ E_k(m - m') \right\|_1 \\ &= \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ (R \otimes \mathrm{Id}_{\mathsf{A}}) \circ A' \circ E_k(m - m') \right\|_1 \\ &= \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \frac{1}{2} \left\| \left(\overline{V}_k \otimes \mathrm{Id}_{\mathsf{A}}\right) \circ A \circ E_k(m - m') \right\|_1 \\ &\leq \delta. \end{aligned} \tag{2.175}$$

Thus, $\mathrm{TE}(S)$ is $\sqrt{\delta}$-tamper evident by application of Lemma 2.21. ∎

2.6.4 Revocation Implies Encryption

An interesting and direct corollary to the work we have done so far in this chapter is that every revocation scheme must be an encryption scheme. This follows from three facts: First, as discussed in Section 2.4, every tamper-evident scheme is an encryption scheme. Second, as discussed in Section 2.6.3.2, every revocation scheme can be transformed into a

tamper-evident scheme via the mapping $S \mapsto \mathrm{TE}(S)$. And, third, this mapping leaves the key generation and encryption channel unchanged. These observations yield the following.

Corollary 2.56. Let $S = (K, E, D, V, R)$ be an ϵ-correct QECMR with δ-revocation security. Then, (K, E, D) is a $\sqrt{19(\sqrt{\delta} + 2\sqrt[8]{\epsilon})}$-encrypting QECM.

Proof: By Theorems 2.53 and 2.55, we have that $\mathrm{TE}(S) = (K', E', D')$ is a $2\sqrt[4]{\epsilon}$-correct $\sqrt{\delta}$-tamper-evident AQECM. By Theorem 2.30, which tells us that tamper evidence implies encryption, $\mathrm{TE}(S)$ is $\sqrt{19(\sqrt{\delta} + 2\sqrt[8]{\epsilon})}$-encrypting. Recall that the property of being encrypting, given in Definition 2.23, depends only on the key generation and the encoding channels. Moreover, since $\mathrm{TE}(S)$ and (K, E, D) have the same key generation and encoding channels by construction (see Definition 2.52), it follows that (K, E, D) is a $\sqrt{19(\sqrt{\delta} + 2\sqrt[8]{\epsilon})}$-encrypting QECM. ∎

2.7 What Tamper Evidence Is Not

We have so far been concerned with "positive" implications of tamper evidence. We complete this chapter by changing our perspective and collecting in this section a few shortcomings, or non-implications, of tamper evidence. We have four results in this category:

- In Section 2.7.1, we show that tamper evidence does not imply uncloneable encryption. In other words, a single ciphertext can be split into two shares, both of which cause the honest decryption procedure to yield the correct message, but neither accepting.
- In Section 2.7.2, we show that tamper evidence may permit an adversary to split a single ciphertext into two states which are both accepted by the honest decryption procedure, but neither yielding the correct message. In other words, tamper evidence does not imply a unique accepting instance.
- In Section 2.7.3.1, we show that tamper evidence does not imply authentication.
- In Section 2.7.3.2, we show that tamper evidence does not imply the encryption of arbitrary quantum states.

In all cases, we obtain optimal separations. We can pick any $\delta > 0$ and instantiate the result for a perfectly correct δ-tamper-evident AQECM where the other security notion is, in some sense, perfectly broken.

The last three results are obtained from the same generic construction using a simple group-based secret sharing scheme described in Section 2.7.2. Even further, we obtain the

last two with precisely the same AQECM scheme described in Section 2.7.3: a classical one-time pad where only the key is encoded with a tamper-evident scheme.

2.7.1 Tamper Evidence Does Not Imply Uncloneable Encryption

Uncloneable encryption was first formalized by Broadbent and Lord [BL20] following ideas and questions initially raised during Gottesman's work on tamper-evident schemes [Got03]. We eschew stating a formal definition of uncloneable encryption in this work. It suffices to recall the general principle that a symmetric key encryption scheme is uncloneable if it is infeasible for an initial adversary ignorant of the key used to encrypt a message to split the resulting ciphertext between two subsequent non-communicating adversaries such that both can recover information on the plaintext, even if *they* know the key. In general, the adversaries attempting to recover information on the plaintext are not restricted to using the decryption procedure prescribed by the scheme.

The exact relation between tamper evidence and uncloneable encryption has remained unclear since Gotesman first highlighted both of these notions [Got03]. In one direction, it is easy to see that uncloneability does not imply tamper evidence as we can equip any uncloneable encryption scheme with a decryption procedure which *always* accepts. Such a scheme cannot offer a non-trivial level of security as a tamper-evident scheme. However, this modification preserves any notion of security as an uncloneable encryption scheme as this security guarantee is independent of the decryption process.

Here, we complete the picture by showing that there is no implication in the other direction either. We do so by explicitly constructing tamper-evident schemes which do not offer any non-trivial security as uncloneable encryption schemes. Specifically, we define a construction taking any tamper-evident scheme S to a new scheme denoted Double(S) which is the parallel repetition of S, namely $S \otimes S$, encoding the same message *twice* with independent keys.

Note that Double(S) differs from $S \otimes S$ in two respects: it adds to the encryption map a mechanism to copy classical plaintexts before encoding them and adds a mechanism to select which of the two recovered messages should ultimately be produced after decoding both ciphertexts. This "message selection" mechanism takes the form of an extra qubit added to the ciphertexts that the decryption map will measure to determine which message to output. By showing that these modifications do not negatively change the correctness or security of Double(S) compared to $S \otimes S$, we obtain that Double(S) is 2ϵ-correct and 2δ-tamper-evident if S is ϵ-correct and δ-tamper-evident.

Clearly, Double(S) offers no non-trivial security as an uncloneable encryption scheme. Indeed, by splitting the two encryptions between two different parties, both will be able to

recover the plaintext once they learn the key. Even further, due to the way we construct the "message selection" mechanism of the decryption channel, both parties will be able to use the honest decryption map to recover the message, something that is stronger than what is needed to break the uncloneable encryption security guarantee.

We formalize this discussion in what follows.

Definition 2.57. Let $S = (K, E, D)$ be an AQECM scheme as given in Definition 2.13 and let $S \otimes S = (K', E', D')$ be the parallel repetition of this scheme with itself, as given in Definition 2.25. Finally, let $V \in \mathcal{U}(\mathsf{M}, \mathsf{M} \otimes \mathsf{M})$ be the isometry defined by $V = \sum_{m \in \mathcal{M}} |m, m\rangle\langle m|$. Note that V perfectly copies all elements of $m \in \mathcal{M}$, which is to say that $VmV^\dagger = m \otimes m$.

We define $\text{Double}(S) = (K'', E'', D'')$ to be the triplet given by the following channels:

- The key generation channel K'' is identical to the key generation channel of $S \otimes S$, i.e. $K'' = K' = K \otimes K$.
- The encoding channel $E'' : \mathcal{L}(\mathsf{K} \otimes \mathsf{K} \otimes \mathsf{M}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}} \otimes \mathsf{C} \otimes \mathsf{C})$ is given by

$$\rho \mapsto |0\rangle\langle 0| \otimes E' \left((I_{\mathsf{K}\otimes\mathsf{K}} \otimes V) \, \rho \, (I_{\mathsf{K}\otimes\mathsf{K}} \otimes V)^\dagger \right) \tag{2.176}$$

  which is to say that E'' first copies the plaintext, assuming it is in the computational basis, via the isometry V and then simply applies the encoding channel E' from $S \otimes S$, followed by prepending a single qubit in the 0 state.

- Let $\Psi : \mathcal{L}(\mathbb{C}^{\{0,1\}} \otimes \mathsf{M} \otimes \mathsf{M}) \to \mathcal{L}(\mathsf{M})$ be a channel which measures, in the computational basis, the first qubit it is given and then either discards the second M register if the measurement outcome is 0 or discards the first M register if the outcome is 1. In other words, the first qubit indicates which plaintext register is kept. Formally, this is given by the map

$$\begin{aligned} \rho \mapsto &(\text{Id}_{\mathsf{M}} \otimes \text{Tr}_{\mathsf{M}}) \left((\langle 0| \otimes I_{\mathsf{M}\otimes\mathsf{M}}) \, \rho \, (|0\rangle \otimes I_{\mathsf{M}\otimes\mathsf{M}}) \right) \\ &+ (\text{Tr}_{\mathsf{M}} \otimes \text{Id}_{\mathsf{M}}) \left((\langle 1| \otimes I_{\mathsf{M}\otimes\mathsf{M}}) \, \rho \, (|1\rangle \otimes I_{\mathsf{M}\otimes\mathsf{M}}) \right). \end{aligned} \tag{2.177}$$

  The decoding channel $D'' : \mathcal{L}(\mathsf{K} \otimes \mathsf{K} \otimes \mathbb{C}^{\{0,1\}} \otimes \mathsf{C} \otimes \mathsf{C}) \to \mathcal{L}(\mathsf{M} \otimes \mathsf{F})$ is then given by

$$D'' = (\Psi \otimes \text{Id}_{\mathsf{F}}) \circ \left(\text{Id}_{\mathbb{C}^{\{0,1\}}} \otimes D' \right) \circ \left(\text{Swap}^{(2,3,1)}_{\mathsf{K},\mathsf{K},\mathbb{C}^{\{0,1\}}} \otimes \text{Id}_{\mathsf{C}\otimes\mathsf{C}} \right) \tag{2.178}$$

  which is to say that D'' begins by applying D', the decoding map from $S \otimes S$, on all but the first qubit, then checks the first qubit via a computational basis measurement to determine which message register M to keep and which to discard, all the while keeping the flag qubit produced by D' untouched. The swap channel is merely there to

correctly route the key material. Finally, note that $\overline{D''}_{(k_0,k_1)} = \Psi \circ \left(\mathrm{Id}_{\mathbb{C}^{\{0,1\}}} \otimes \overline{D'}_{(k_0,k_1)}\right)$ for all keys $(k_0, k_1) \in \mathcal{K} \times \mathcal{K}$.

We now prove that the mapping $S \mapsto \mathrm{Double}(S)$ preserves the correctness and security of the scheme, up to the factor of 2 introduced by passing via the $S \otimes S$ scheme.

For correctness, this is essentially trivial. The probability of accepting and correctly recovering both messages, i.e. the $S \otimes S$ correctness criterion, cannot be less than the probability of accepting both messages and correctly recovering the first, which is precisely the $\mathrm{Double}(S)$ correctness criterion.

Theorem 2.58. If S is an ϵ-correct AQECM scheme, then $\mathrm{Double}(S)$ is 2ϵ-correct.

Proof: This follows essentially immediately from the fact that $S \otimes S$ is 2ϵ-correct, as shown in Theorem 2.26. We go through the details for completeness.

Let S, $S \otimes S$, and $\mathrm{Double}(S)$ be defined and denoted as in Definition 2.57. By definition, for every key $(k_0, k_1) \in \mathcal{K} \times \mathcal{K}$ and every message $m \in \mathcal{M}$ we have that

$$E''_{(k_0,k_1)}(m) = |0\rangle\langle 0| \otimes E'_{(k_0,k_1)}(m \otimes m). \tag{2.179}$$

It then follows that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(k_0,k_1)\leftarrow K''(\varepsilon)} \langle m| \overline{D''}_{(k_0,k_1)} \circ E''_{(k_0,k_1)}(m) |m\rangle \\
&= \mathop{\mathbb{E}}_{(k_0,k_1)\leftarrow K'(\varepsilon)} \langle m| \left(\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{M}}\right) \circ \overline{D'}_{(k_0,k_1)} \circ E'_{k_0,k_1}(m \otimes m) |m\rangle \\
&\geq \mathop{\mathbb{E}}_{(k_0,k_1)\leftarrow K'(\varepsilon)} \langle m, m| \overline{D'}_{(k_0,k_1)} \circ E'_{(k_0,k1)}(m \otimes m) |m, m\rangle \\
&\geq 1 - 2\epsilon
\end{aligned} \tag{2.180}$$

where the first inequality holds as the event of measuring both plaintexts to be m cannot be more likely than the event of measuring the first plaintext to be m and the second inequality follows from the fact that $S \otimes S$ is 2ϵ-correct. Thus, $\mathrm{Double}(S)$ is 2ϵ-correct. ∎

For security, we note that the choice to accept or reject the ciphertext is independent of the extra qubit used for the message selection mechanism and that this qubit does not depend on the original message or key. Informally, this means it does not provide any useful information to the adversary. Formally, this means that it can be simulated by the adversary.

Theorem 2.59. If S is a δ-tamper-evident AQECM, then $\mathrm{Double}(S)$ is 2δ-tamper-evident.

Proof: This follows quickly from the fact that $S \otimes S$ is 2δ-tamper-evident, as shown in Theorem 2.26. We demonstrate that any attack against Double(S) implies an attack against $S \otimes S$ with the same performance by showing that an adversary can simulate all differences between these two schemes.

Let S, $S \otimes S$, and Double(S) be defined and denoted as in Definition 2.57. Next, let $A'' : \mathcal{L}(\mathbb{C}^{\{0,1\}} \otimes \mathsf{C} \otimes \mathsf{C}) \to \mathcal{L}(\mathbb{C}^{\{0,1\}} \otimes \mathsf{C} \otimes \mathsf{C} \otimes \mathsf{A})$ be a tamper attack against Double(S). Consider now the tamper attack $A' : \mathcal{L}(\mathsf{C} \otimes \mathsf{C}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{C} \otimes \mathsf{A})$ against $S \otimes S$ given by

$$\rho \mapsto (\mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \mathrm{Id}_{\mathsf{C}\otimes\mathsf{C}\otimes\mathsf{A}}) \circ A'' \left(|0\rangle\langle 0| \otimes \rho\right) \tag{2.181}$$

In essence, A' simulates the attack A'' by first providing it with the extra qubit it expects, and then removing it before returning a ciphertext to the decoding map D' of $S' = S \otimes S$. So, for every message $m \in \mathcal{M}$ and key $(k_0, k_1) \in \mathcal{K} \times \mathcal{K}$, we have by definition that

$$A' \circ E'_{(k_0,k_1)}(m \otimes m) = (\mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \mathrm{Id}_{\mathsf{C}\otimes\mathsf{C}\otimes\mathsf{A}}) \circ A'' \circ E''_{(k_0,k_1)}(m). \tag{2.182}$$

Next, we claim that

$$\mathrm{Tr}_{\mathsf{M}} \circ \overline{D''}_{(k_0,k_1)} = \mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}} \circ \overline{D'}_{(k_0,k_1)} \circ (\mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \mathrm{Id}_{\mathsf{C}\otimes\mathsf{C}}) \,. \tag{2.183}$$

Intuitively, this is simply stating that discarding the single plaintext ultimately produced by $\overline{D''}_{(k_0,k_1)}$ is equivalent to discarding both plaintexts produced by $\overline{D'}_{(k_0,k_1)}$, ignoring for the moment the extra qubit in the ciphertext. Formally, the trace-preserving property of channels implies that $\mathrm{Tr}_{\mathsf{M}} \circ \Psi = \mathrm{Tr}_{\mathbb{C}^{\{0,1\}}\otimes\mathsf{M}\otimes\mathsf{M}} = \mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}}$. We can then write

$$\begin{aligned}
\mathrm{Tr}_{\mathsf{M}} \circ \overline{D''}_{(k_0,k_1)} &= \mathrm{Tr}_{\mathsf{M}} \circ \Psi \circ \left(\mathrm{Id}_{\mathbb{C}^{\{0,1\}}} \otimes \overline{D'}_{(k_0,k_1)}\right) \\
&= \mathrm{Tr}_{\mathbb{C}^{\{0,1\}}\otimes\mathsf{M}\otimes\mathsf{M}} \left(\mathrm{Id}_{\mathbb{C}^{\{0,1\}}} \otimes \overline{D'}_{(k_0,k_1)}\right) \\
&= \mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \left(\mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}} \circ \overline{D'}_{(k_0,k_1)}\right) \\
&= \left(\mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}} \circ \overline{D'}_{(k_0,k_1)}\right) \circ (\mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \mathrm{Id}_{\mathsf{C}\otimes\mathsf{C}})
\end{aligned} \tag{2.184}$$

which is the desired claim.

This, and a final remainder that $K'' = K'$, allows us to write that for any two messages

$m, m' \in \mathcal{M}$ it holds that

$$\begin{aligned}
&\Pr_{(k_0,k_1)\leftarrow K''(\varepsilon)}\left[\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D''}_{(k_0,k_1)}\right)\otimes \mathrm{Id}_{\mathsf{A}}\right)\circ A''\circ E''_{(k_0,k_1)}(m-m')\right\|_1 \leq 2\delta\right] \\
&= \Pr_{(k_0,k_1)\leftarrow K'(\varepsilon)}\left[\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}}\circ\overline{D'}_{(k_0,k_1)}\right)\otimes \mathrm{Id}_{\mathsf{A}}\right)\circ A'\circ E'_{(k_0,k_1)}(m\otimes m-m'\otimes m')\right\|_1 \leq 2\delta\right] \\
&\geq 1-2\delta
\end{aligned} \tag{2.185}$$

where the last inequality follows from the fact that $S \otimes S$ is 2δ-tamper evident. Thus, the scheme Double(S) is also 2δ-tamper evident. ∎

As mentioned, the end result of this construction is that it yields an optimal separation between tamper-evident schemes and uncloneable encryption schemes. This is the content of the next theorem.

Theorem 2.60. For any strictly positive $\delta \in \mathbb{R}^+$ and any alphabet $\mathcal{M}$, there exists a perfectly correct δ-tamper-evident scheme $S' = (K', E', D')$ with messages $\mathcal{M}$ and ciphertext space C' as well as a channel $\tilde{A} : \mathcal{L}(\mathsf{C}' \otimes \mathsf{C}') \to \mathcal{L}(\mathsf{C}' \otimes \mathsf{C}')$ such that for any message $m \in \mathcal{M}$ and any key k' with $\langle k'| K'(\varepsilon) |k'\rangle > 0$, we have that

$$\langle m, m| \left(\mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}} \otimes \mathrm{Id}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}\right) \circ \left(D'_{k'} \circ D'_{k'}\right) \circ \tilde{A} \circ E'_k(m) |m, m\rangle = 1. \tag{2.186}$$

Proof: By Theorem 2.22, there exists a perfectly correct $\frac{\delta}{2}$-tamper-evident AQECM scheme S with messages $\mathcal{M}$. Then, $S' = \text{Double}(S)$ is a perfectly correct δ-tamper-evident scheme. It then suffices to consider the channel $\tilde{A}$ given by

$$\rho \mapsto \mathrm{Swap}^{(1,4,3,5,2,6)}_{\mathbb{C}^{\{0,1\}},\mathbb{C}^{\{0,1\}},\mathsf{C},\mathsf{C},\mathsf{C},\mathsf{C}}\left(|0\rangle\langle 0| \otimes |1\rangle\langle 1| \otimes \frac{\mathrm{Id}_{\mathsf{C}}}{\dim \mathsf{C}} \otimes \frac{\mathrm{Id}_{\mathsf{C}}}{\dim \mathsf{C}} \otimes \left(\mathrm{Tr}_{\mathbb{C}^{\{0,1\}}} \otimes \mathrm{Id}_{\mathsf{C}\otimes\mathsf{C}}\right)(\rho)\right) \tag{2.187}$$

so that for any key $k' = (k_0, k_1)$ which has a non-zero probability of being produced by K' and any message m we have

$$\tilde{A} \circ E'_{(k_0,k_1)}(m) = \left(|0\rangle\langle 0| \otimes E_{k_0}(m) \otimes \frac{\mathrm{Id}_{\mathsf{C}}}{\dim \mathsf{C}}\right) \otimes \left(|1\rangle\langle 1| \otimes \frac{\mathrm{Id}_{\mathsf{C}}}{\dim \mathsf{C}} \otimes E_{k_1}(m)\right). \tag{2.188}$$

Both of these are valid ciphertexts for the scheme Double(S) and it is clear that the decryption map $D'_{(k_0,k_1)}$ of Double(S) will recover the correct plaintext m in both cases due to how the plaintext selection qubit is fixed and the perfect correctness of the scheme S. ∎

2.7.2 Tamper Evidence with a Simple Secret Sharing Scheme

We describe in this section a method to construct a tamper-evident scheme by combining two tamper-evident schemes with a secret sharing scheme (e.g.: [KL15, Sec. 13.3]). This construction will be useful to obtain various separations.

We conjecture that our construction works with *any* information theoretic 2-out-of-2, or even any k-out-of-n secret sharing scheme, but we restrict ourselves to considering the following simple group-based secret sharing scheme similar to the one-time pad. Consider an alphabet $\mathcal{M}$ equipped with a binary operation $* : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ such that $(\mathcal{M}, *)$ forms a group. The secret sharing scheme we use maps any element $m \in \mathcal{M}$ to the couple $(p, p*m)$ for a uniformly random $p \in \mathcal{M}$. Clearly, neither p nor $p * m$, independently, depends on m when p is sampled uniformly at random. This holds as $p \mapsto p * m$ is a permutation for all $m \in \mathcal{M}$. Hence, neither pieces of information alone is sufficient to obtain any information on m. But, knowledge of both p and $p * m$ is clearly sufficient to recover m.

We combine this secret sharing scheme with any two AQECM schemes $S = (K, E, D)$ and $S' = (K', E', D')$ which share the same messages $\mathcal{M}$ to obtain a new AQECM we will denote $S * S'$. On input of a message m and a pair of keys $(k, k') \in \mathcal{K} \times \mathcal{K}'$, the encryption procedure of this new scheme will sample a uniformly random $p \in \mathcal{M}$ and output $E_k(p) \otimes E'_{k'}(p * m)$. The decryption procedure of $S * S'$ will proceed as expected: both ciphertexts will be decrypted using the decryption maps D and D', respectively, and the appropriate group action will be applied to recover the message m. However, the decryption map of $S * S'$ will reject if and only if *both* ciphertexts were rejected by D. Otherwise, i.e. if at least one of D or D' accepts, the decryption map of $S * S'$ will accept.

Assume that S and S' exhibit sufficiently good correctness and security as tamper-evident schemes. Correctness of $S * S'$ follows naturally from the correctness of S and S'. The security of $S * S'$ as a tamper-evident scheme essentially follows from the fact that if no tampering is detected on at least one of the two ciphertexts produced by this scheme, then an eavesdropper is highly unlikely to have learned information on both shares of the secret sharing scheme and hence is unlikely to have learned any information on the message.

We formalize this discussion in the following. In particular, we will require that the group operation be computed coherently. This will be a useful property later.

Definition 2.61. Let $S = (K, E, D)$ and $S' = (K', E', D')$ be two AQECM schemes as given in Definition 2.13 which share the same message space $\mathcal{M}$. Let this message space be equipped with a group operation denoted $*$ and let $U_* \in \mathcal{U}(\mathsf{M} \otimes \mathsf{M})$ be the unitary operator which coherently computes this operation, which is to say that $U(|a\rangle \otimes |b\rangle) = |a\rangle \otimes |a * b\rangle$ for all $a, b \in \mathcal{M}$.[26] We define the AQECM scheme $S * S' = (K^*, E^*, D^*)$ as follows:

[26] Such a group operation can be found for any alphabet $\mathcal{M}$ by considering any bijection $\phi : \mathcal{M} \to \mathbb{Z}/|\mathcal{M}|\mathbb{Z}$,

- The key generation channel $K^* : \mathbb{C} \to \mathcal{L}(\mathsf{K} \otimes \mathsf{K}')$ is simply the parallel application of the key generation channels of S and S', i.e. $K^* = K \otimes K'$.

- The encryption channel $E^* : \mathcal{L}(\mathsf{K} \otimes \mathsf{K}' \otimes \mathsf{M}) \to \mathcal{L}(\mathsf{C} \otimes \mathsf{C}')$ is given by

$$\rho \mapsto \mathop{\mathbb{E}}_{p \leftarrow \mathcal{M}} \left(E \otimes E'\right) \circ \mathrm{Swap}^{(1,3,2,4)}_{\mathsf{K},\mathsf{K}',\mathsf{M},\mathsf{M}} \left(\left(I_{\mathsf{K}\otimes\mathsf{K}'} \otimes U_*\right) (p \otimes \rho) \left(I_{\mathsf{K}\otimes\mathsf{K}'} \otimes U_*\right)^\dagger \right). \quad (2.189)$$

  In other words, E^* samples uniformly at random an element $p \in \mathcal{M}$, coherently computes the result of the group action between the supplied message and p and encrypts the resulting state on the message spaces with $E \otimes E'$ after having properly routed the keys. For a state of the expected form $\rho = k \otimes k' \otimes m$, the result of this channel is

$$E^*_{(k,k')}(m) = \mathop{\mathbb{E}}_{p \leftarrow \mathcal{M}} E_k(p) \otimes E'_{k'}(p * m). \quad (2.190)$$

- The decryption channel $D^* : \mathcal{L}\left(\mathsf{K} \otimes \mathsf{K}' \otimes \mathsf{C} \otimes \mathsf{C}'\right) \to \mathcal{L}(\mathsf{M} \otimes \mathsf{F})$ is given by

$$\rho \mapsto W \circ \left(D \otimes D'\right) \circ \mathrm{Swap}^{(1,3,2,4)}_{\mathsf{K},\mathsf{K}',\mathsf{C},\mathsf{C}'}(\rho) \quad (2.191)$$

  where $W : \mathcal{L}(\mathsf{M} \otimes \mathsf{F} \otimes \mathsf{M} \otimes \mathsf{F}) \to \mathcal{L}(\mathsf{M} \otimes \mathsf{F})$ is a channel which (1) measures both flag qubits produced by D and D' and outputs the logical "or" of the result, (2) coherently undoes the group operation on the message spaces, and (3) discards the final unnecessary message space supposed to hold the p element sampled by the encryption channel. Formally, this is given by

$$\rho \mapsto (\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{M}\otimes\mathsf{F}}) \left(\sum_{b,b' \in \{0,1\}} \left(U_*^\dagger \otimes \left|b \vee b'\right\rangle\!\left\langle b, b'\right|\right) \mathrm{Swap}^{(1,3,2,4)}_{\mathsf{M},\mathsf{F},\mathsf{M},\mathsf{F}}(\rho) \left(U_* \otimes \left|b, b'\right\rangle\!\left\langle b \vee b'\right|\right) \right). \quad (2.192)$$

The following theorem states that if S and S' are ϵ-correct and ϵ'-correct, then $S * S'$ is $(\epsilon+\epsilon')$-correct. This follows from the fact that the probability that the decryption procedure of $S * S'$ both produces the correct message and accepts is at least the probability that both instances of the decryption procedure from S accept and produce the correct messages. As the keys are independently sampled, this is at least $(1-\epsilon)(1-\epsilon') \geq 1-(\epsilon+\epsilon')$. Note that this is identical to the correctness of $S \otimes S'$. In particular, we are disregarding the possibility that only one decryption procedure accepts, or that these decryption maps both output wrong messages which, by coincidence, still lead to the correct message being recovered after application of the group operation.

where $\mathbb{Z}/|\mathcal{M}|\mathbb{Z}$ is the additive group of integers modulo $|\mathcal{M}|$. Simply define $m * m' = \phi^{-1}(\phi(m) + \phi(m'))$.

Theorem 2.62. Let S and S' be ϵ- and ϵ'-correct AQECMs, respectively, with the same message space $\mathcal{M}$ which admits a group operation denoted $*$. Then, $S*S'$ is $(\epsilon+\epsilon')$-correct.

Proof: For all keys $(k,k') \in \mathcal{K}\times\mathcal{K}'$ and all messages $m \in \mathcal{M}$ we have by definition that

$$\begin{aligned}\langle m|\,\overline{D^*}_{(k,k')}\circ E^*_{(k,k')}(m)\,|m\rangle &= \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\langle m|\,\overline{D^*}_{(k,k')}\left(E_k(p)\otimes E_{k'}(p*m)\right)|m\rangle\\ &\geq \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\langle p,p*m|\,\overline{D}_k\circ E_k(p)\otimes\overline{D'}_{k'}\circ E'_{k'}(p*m)\,|p,p*m\rangle\\ &= \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\langle p|\,\overline{D}_k\circ E_k(p)\,|p\rangle\cdot\langle p*m|\,\overline{D'}_{k'}\circ E'_{k'}(p*m)\,|p*m\rangle\end{aligned} \tag{2.193}$$

Now, recalling that $K^* = K\otimes K'$ so that both elements of the key pair are generated independently, we have that

$$\begin{aligned}&\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)}\langle m|\,\overline{D^*}_{(k_0,k_1)}\circ E^*_{(k_0,k_1)}(m)\,|m\rangle\\ &\geq \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\left(\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)}\langle p|\,\overline{D}_k\circ E_k(p)\,|p\rangle\right)\left(\mathop{\mathbb{E}}_{k'\leftarrow K'(\varepsilon)}\langle p*m|\,\overline{D'}_{k'}\circ E'_{k'}(p*m)\,|p*m\rangle\right)\\ &\geq \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}(1-\epsilon)(1-\epsilon')\\ &\geq 1-(\epsilon+\epsilon')\end{aligned} \tag{2.194}$$

which is the desired result. ∎

We now move on to proving the security of $S*S'$. We begin with a technical lemma providing us with a useful decomposition of the map $\mathrm{Tr}_{\mathsf{M}}\circ\overline{D^*}_{(k,k')}$ obtained by application of the inclusion-exclusion principal. Note that this channel computes the probability that at least one of the two underlying ciphertexts is accepted. This is equal to the probability that the first is accepted, plus the probability that the second is accepted, minus the probability that both were accepted.

Lemma 2.63. In the notation of Definition 2.61, we have for every $(k,k')\in\mathcal{K}\times\mathcal{K}'$ that

$$\mathrm{Tr}_{\mathsf{M}}\circ\overline{D'}_{(k,k')} = \left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\mathrm{Tr}_{\mathsf{C}} + \mathrm{Tr}_{\mathsf{C}}\otimes\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_{k'}\right) - \left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_{k'}\right). \tag{2.195}$$

Proof: By definition, for every key pair $(k,k')\in\mathcal{K}\times\mathcal{K}'$ we have that $\overline{D^*}_{(k,k')}(\rho)$ is given by

$$(\mathrm{Tr}_{\mathsf{M}}\otimes\mathrm{Id}_{\mathsf{M}})\left(\sum_{\substack{b,b'\in\{0,1\}\\(b,b')\neq(0,0)}}\left(U_*^\dagger\otimes\langle b,b'|\right)\mathrm{Swap}^{(1,3,2,4)}_{\mathsf{M},\mathsf{F},\mathsf{M},\mathsf{F}}\circ(D_k\otimes D'_{k'})(\rho)\left(U_*\otimes|b,b'\rangle\right)\right) \tag{2.196}$$

It follows that $\mathrm{Tr}_{\mathsf{M}} \circ \overline{D^*}_{(k,k')}(\rho)$ is given by

$$\mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}} \left(\sum_{\substack{b,b'\in\{0,1\} \\ (b,b')\neq(0,0)}} \left(U_*^\dagger \otimes \left\langle b,b'\right|\right) \mathrm{Swap}^{(1,3,2,4)}_{\mathsf{M},\mathsf{F},\mathsf{M},\mathsf{F}} \circ (D_k \otimes D'_{k'})(\rho) \left(U_* \otimes \left|b,b'\right\rangle\right) \right). \tag{2.197}$$

Tracing out the message spaces after undoing the group operation via conjugation by $U_*^\dagger$ is equivalent to tracing them out before this conjugation. We can even take this trace prior to the Swap operation. Hence, the previous expression is equivalent to

$$\sum_{\substack{b,b'\in\{0,1\} \\ (b,b')\neq(0,0)}} \left\langle b,b'\right| \left(((\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{F}}) \circ D_k) \otimes \left((\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{F}}) \circ D'_{k'}\right)\right)(\rho) \left|b,b'\right\rangle \tag{2.198}$$

which is, in turn, equivalent to

$$\left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \mathrm{Tr}_{\mathsf{C}'} + \mathrm{Tr}_{\mathsf{C}} \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) - \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right)\right)(\rho) \tag{2.199}$$

where the first summand counts the cases of $(b_0, b_1) \in \{(1,0),(1,1)\}$, the second the cases of $(b_0,b_1) \in \{(0,1),(1,1)\}$, and the third accounts for the fact that the case of $(b_0,b_1) = (1,1)$ would otherwise be double counted. Note that the Tr_{C} term arises as a result of the fact that $\sum_{b\in\{0,1\}} \langle b| (\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Id}_{\mathsf{F}}) D_k(\rho) |b\rangle = (\mathrm{Tr}_{\mathsf{M}} \otimes \mathrm{Tr}_{\mathsf{F}}) \circ D_k(\rho) = \mathrm{Tr}_{\mathsf{C}}(\rho)$ where the first equality is obtained by a characterization of the trace and the second by the fact that D_k is trace preserving. The $\mathrm{Tr}_{\mathsf{C}'}$ arises similarly. ∎

With the previous lemma in hand, we can now prove the security of $S * S'$.

Theorem 2.64. Let S and S' be δ- and δ'-tamper-evident AQECMs, respectively, with the same message space $\mathcal{M}$ which admits a group operation denoted $*$. Then, the AQECM scheme $S * S'$ is $2\sqrt{\delta+\delta'}$-tamper evident.

Proof: Fix two messages $m, \tilde{m} \in \mathcal{M}$ and a tamper attack $A^* : \mathcal{L}(\mathsf{C}\otimes\mathsf{C}') \to \mathcal{L}(\mathsf{C}\otimes\mathsf{C}'\otimes\mathsf{A}^*)$ against the AQECM scheme $S * S' = (K^*, E^*, D^*)$. For any two elements $p, m \in \mathcal{M}$ and any key $(k,k') \in \mathcal{K}\times\mathcal{K}' = \mathcal{K}^*$, we define the states

$$\rho^{p,m}_{k,k'} = A^* \left(E_k(p) \otimes E'_{k'}(p * m)\right) \tag{2.200}$$

and

$$\rho^m_{k,k'} = A^* \circ E^*_{(k,k')}(m). \tag{2.201}$$

Note that $\rho^{p,m}_{k,k'}$ is the state after the adversary's attack — but before the application of the decryption map — when the message m is encrypted with the key (k,k') and the secret sharing scheme picked p as the additional element. Further note that $\rho^{m}_{k,k'} = \mathbb{E}_{p\leftarrow\mathcal{M}}\,\rho^{p,m}_{k,k'}$ is the same state, but averaged over all choices of p.

Recall that Lemma 2.21 essentially restates the property of being tamper evident from a concentration inequality on

$$\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D^*}_{(k,k')}\right)\otimes\mathrm{Id}_{\mathsf{A}^*}\right)\left(\rho^{m}_{k,k'}-\rho^{\tilde{m}}_{k,k'}\right)\right\|_1 \tag{2.202}$$

over the sampling of the key (k,k') to an expectation of this trace distance over the same sampling. This is done via Markov's inequality (Theorem 2.1). Our goal for the remainder of the proof will be to give an upper bound for the expected value of this quantity over sampling of the key (k,k'). We note that working with the expectations will simplify our accounting of the random sampling of the p value by the encryption channel E^*.

Using the result and notation of Lemma 2.63, linearity, and the fact that norms are subadditive, for any key $(k,k')\in\mathcal{K}^*$ and any two messages $m,\tilde{m}\in\mathcal{M}$, we have that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)}\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D^*}_{(k,k')}\right)\otimes\mathrm{Id}_{\mathsf{A}^*}\right)\left(\rho^{m}_{k,k'}-\rho^{\tilde{m}}_{k,k'}\right)\right\|_1\\
&\leq \mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)}\mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D^*}_{(k,k')}\right)\otimes\mathrm{Id}_{\mathsf{A}^*}\right)\left(\rho^{p,m}_{k,k'}-\rho^{p,\tilde{m}}_{k,k'}\right)\right\|_1\\
&\leq \mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)}\mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_{k}\right)\otimes\mathrm{Tr}_{\mathsf{C}'}\otimes\mathrm{Id}_{\mathsf{A}}\right)\left(\rho^{p,m}_{k,k'}-\rho^{p,\tilde{m}}_{k,k'}\right)\right\|_1\\
&\quad+\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)}\mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\frac{1}{2}\left\|\left(\mathrm{Tr}_{\mathsf{C}}\otimes\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D'}_{k'}\right)\otimes\mathrm{Id}_{\mathsf{A}}\right)\left(\rho^{p,m}_{k,k'}-\rho^{p,\tilde{m}}_{k,k'}\right)\right\|_1\\
&\quad+\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)}\mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_{k}\right)\otimes\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D'}_{k'}\right)\otimes\mathrm{Id}_{\mathsf{A}}\right)\left(\rho^{p,m}_{k,k'}-\rho^{p,\tilde{m}}_{k,k'}\right)\right\|_1.
\end{aligned}\tag{2.203}$$

Looking at the last of these inequalities, we will give an upperbound on the first two summands by using the fact that S and S' are δ- and δ'-tamper-evident, respectively, and we will upperbound the third summand by using the fact that $S\otimes S'$ is $(\delta+\delta')$-tamper-evident by Theorem 2.27.

We begin with the third summand. Let $S''=S\otimes S'=(K'',E'',D'')$ and note the following facts. First, the channel A^* is also a tamper attack against S'', second we have that $(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k)\otimes(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D'}_{k'})=\mathrm{Tr}_{\mathsf{M}\otimes\mathsf{M}}\circ\overline{D''}_{(k,k')}$, and third by construction we have that $K^*=K''$. These, with the fact that S'' is $(\delta+\delta')$-tamper evident, imply via Lemma 2.21

that

$$\mathop{\mathbb{E}}_{(k,k)\leftarrow K^*(\varepsilon)} \frac{1}{2}\left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \left(\rho_{k,k'}^{p,m} - \rho_{k,k'}^{p,\tilde{m}}\right) \right\|_1 \leq 2(\delta+\delta') - (\delta+\delta')^2 \tag{2.204}$$

for all $p \in \mathcal{M}$. As this inequality holds for all $p \in \mathcal{M}$, we can also bound the expectation over the sampling of p:

$$\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)} \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}} \frac{1}{2}\left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \left(\rho_{k,k'}^{p,m} - \rho_{k,k'}^{p,\tilde{m}}\right) \right\|_1 \leq 2(\delta+\delta') - (\delta+\delta')^2. \tag{2.205}$$

We now move on to the first two summands. We will only explicitly treat the second as the bound for first second is obtained completely analogously. For every $p \in \mathcal{M}$ and $k \in \mathcal{K}$, let $A'_{k,p} : \mathcal{L}(\mathsf{C}') \to \mathcal{L}(\mathsf{C}' \otimes \mathsf{A}^*)$ be defined by

$$\rho \mapsto \left(\mathrm{Tr}_{\mathsf{C}} \otimes \mathrm{Id}_{\mathsf{C}'\otimes\mathsf{A}^*}\right) \circ A^* \left(E_k(p) \otimes \rho\right). \tag{2.206}$$

Note that $A'_{k,p}$ is a tamper attack against the underlying AQECM scheme S' which is assumed to be δ'-tamper-evident. Thus, for every $m, \tilde{m} \in \mathcal{M}$ Lemma 2.21 implies that

$$\mathop{\mathbb{E}}_{k'\leftarrow K'(\varepsilon)} \frac{1}{2}\left\| \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \circ A'_{k,p} \circ E'_{k'}((m * p) - (\tilde{m} * p)) \right\|_1 \leq 2\delta' - \delta'^2. \tag{2.207}$$

On the other hand, by definition of $A'_{k,p}$, we have that

$$\begin{aligned}
&\left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \circ A'_{k,p} \left(E'_{k'}((p * m) - (p * \tilde{m}))\right) \\
&= \left(\left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \circ \left(\mathrm{Tr}_{\mathsf{C}} \otimes \mathrm{Id}_{\mathsf{C}'\otimes\mathsf{A}^*}\right) \circ A^* \left(E_k(p) \otimes E'_{k'}(p * m) - E_k(p) \otimes E'_{k'}(p * \tilde{m})\right) \\
&= \left(\mathrm{Tr}_{\mathsf{C}} \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \left(\rho_{k,k'}^{m,p} - \rho_{k,k'}^{\tilde{m},p}\right).
\end{aligned} \tag{2.208}$$

Thus,

$$\mathop{\mathbb{E}}_{k'\leftarrow K'(\varepsilon)} \frac{1}{2}\left\| \left(\mathrm{Tr}_{\mathsf{C}} \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D'}_{k'}\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \left(\rho_{k,k'}^{m,p} - \rho_{k,k'}^{\tilde{m},p}\right) \right\|_1 \leq 2\delta' - \delta'^2. \tag{2.209}$$

As this holds for all k and all p, we can also bound the expectation over the sampling of both these values:

$$\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)} \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}} \frac{1}{2}\left\| \left(\mathrm{Tr}_{\mathsf{C}} \otimes \left(\mathrm{Tr}_{\mathsf{M}} \circ \overline{D}_k\right) \otimes \mathrm{Id}_{\mathsf{A}^*}\right) \left(\rho_{k,k'}^{m,p} - \rho_{k,k'}^{\tilde{m},p}\right) \right\|_1 \leq 2\delta' - \delta'^2. \tag{2.210}$$

Noting that $\mathbb{E}_{p\leftarrow\mathcal{M}} E_k(p) \otimes E'_{k'}(p * m) = \mathbb{E}_{p\leftarrow\mathcal{M}} E_k(p * m^{-1}) \otimes E'_{k'}(p)$ where m^{-1} is the

inverse of m for $*$, a similar exercise for the first summand will yield

$$\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)} \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}} \frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D}_k\right)\otimes\mathrm{Tr}_{\mathsf{C}'}\otimes\mathrm{Id}_{\mathsf{A}^*}\right)\left(\rho^{m,p}_{k,k'}-\rho^{\tilde{m},p}_{k,k'}\right)\right\|_1 \leq 2\delta-\delta^2. \tag{2.211}$$

We conclude that

$$\begin{aligned}\mathop{\mathbb{E}}_{(k,k')\leftarrow K^*(\varepsilon)} \frac{1}{2}\left\|\left(\left(\mathrm{Tr}_{\mathsf{M}}\circ\overline{D^*}_{(k,k')}\right)\otimes\mathrm{Id}_{\mathsf{A}^*}\right)\left(\rho^{m}_{k,k'}-\rho^{\tilde{m}}_{k,k'}\right)\right\|_1 &\leq 4(\delta+\delta')-(\delta+\delta')^2-\delta^2-\delta'^2\\ &\leq 4(\delta+\delta')\end{aligned} \tag{2.212}$$

where, for simplicity, we drop the higher order terms. A final invocation of Lemma 2.21 with this bound yields that S^* is $2\sqrt{\delta+\delta'}$-tamper evident. ∎

In the following theorem, we precisely state that ciphertexts produced by $S * S'$ can be split into two parts, both of which will be accepted by the honest decryption procedure. We merely sketch the proof as the idea is quite simple: leave one unmodified encoded share of the secret sharing scheme in each part.

Theorem 2.65. For every $\delta > 0$ and every alphabet $\mathcal{M}$, there exists a perfectly correct δ-tamper-evident AQECM scheme with messages $\mathcal{M}$ as well as a channel $\tilde{A}$ satisfying

$$\mathop{\mathbb{E}}_{k\leftarrow K(\varepsilon)} \mathrm{Tr}\left[\left(\overline{D}_k\otimes\overline{D}_k\right)\circ\tilde{A}\circ E_k(m)\right] = 1 \tag{2.213}$$

for all messages $m\in\mathcal{M}$.

Proof: By Theorem 2.22, there exists a perfectly correct $\frac{1}{2}\delta^2$-tamper-evident AQECM scheme S' with messages $\mathcal{M}$. Consider now $S = S'*S'$ which is a perfectly correct δ-tamper-evident AQECM with messages $\mathcal{M}$. It then suffices to take the channel $\tilde{A}:\mathcal{L}(\mathsf{C})\to\mathcal{L}(\mathsf{C}\otimes\mathsf{C})$ defined by

$$\rho\mapsto \mathrm{Swap}^{(1,4,2,3)}_{\mathsf{C},\mathsf{C}',\mathsf{C}',\mathsf{C}'}\left(\rho\otimes\frac{I_{\mathsf{C}'}}{\dim\mathsf{C}'}\otimes\frac{I_{\mathsf{C}'}}{\dim\mathsf{C}'}\right) \tag{2.214}$$

where C' is the ciphertext space of the underlying AQECM scheme S'. Thus, for any messages m and key $k=(k_0,k_1)$, we have that

$$\begin{aligned}\tilde{A}\circ E_k(m) &= \tilde{A}\left(\mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}} E'_{k_0}(p)\otimes E'_{k_1}(p*m)\right)\\ &= \mathop{\mathbb{E}}_{p\leftarrow\mathcal{M}}\left(E'_{k_0}(p)\otimes\frac{I_{\mathsf{C}'}}{\dim\mathsf{C}}\right)\otimes\left(\frac{I_{\mathsf{C}'}}{\dim\mathsf{C}'}\otimes E'_{k_1}(p*m)\right).\end{aligned} \tag{2.215}$$

The fact that the underlying scheme S' is perfectly correct implies that both instances of

the decryption map in $\overline{D}_k$ will accept with certainty for any $k = (k_0, k_1)$, m and p. ∎

2.7.3 One-Time Pad with a Tamper-Evident Encoded Key

The goal of this subsection is to formally study an AQECM scheme obtained by combining a classical one-time pad with a tamper-evident scheme $S = (K, E, D)$. The idea is to encode a message m with a random one-time pad p and then encrypt p with the tamper-evident scheme and make the resulting state part of the ciphertext. For a key k and a message m taken from an alphabet $\mathcal{M}$ equipped with a group action $*$, the ciphertexts produced by this scheme are of the form

$$\mathop{\mathbb{E}}_{p \leftarrow \mathcal{M}} E_k(p) \otimes (p * m). \tag{2.216}$$

Decryption proceeds as expected. With the key k, recover p from the first component using the decryption map D_k from the AQECM scheme S, after which m can be recovered from the second component $p * m$. Accept if and only if D_k accepted.

As we will see, such a scheme will be a useful example to demonstrate that tamper evidence does *not* imply certain properties.

Note that this scheme appears quite similar to the ones constructed in Section 2.7.2. In fact, one could describe this scheme as simply $S * T$ where T is a trivial AQECM scheme on the same message space $\mathcal{M}$. We formally give such a trivial scheme in the next definition.

Definition 2.66. Let $\mathcal{M}$ be an alphabet. The *always rejecting trivial AQECM scheme for* $\mathcal{M}$, denoted $\mathrm{Triv}^{\mathcal{M},0}$, is an AQECM scheme with messages $\mathcal{M}$, a single key $\mathcal{K} = \{\varepsilon\}$, ciphertext space $\mathsf{C} = \mathsf{M}$, and defined by the following three channels:

- The key generation channel $K^{\mathcal{M},0}$ is the identity on $\mathbb{C}^{\{\varepsilon\}}$, which is isomorphic to $\mathbb{C}$.
- The encryption channel $E^{\mathcal{M},0}$ is the identity on $\mathcal{L}(\mathsf{M}) = \mathcal{L}(\mathsf{C})$.
- The decryption channel $D^{\mathcal{M},0} : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{M} \otimes \mathsf{F})$ is defined by $\rho \mapsto \rho \otimes |0\rangle\langle 0|$.

We neglect the key space in the domains of $E^{\mathcal{M},0}$ and $D^{\mathcal{M},0}$ as $\mathbb{C} \otimes \mathsf{M} = \mathsf{M} = \mathsf{C} = \mathbb{C} \otimes \mathsf{C}$ and will omit the ε subscript for simplicity when discussing the keyed version of these channels.

By construction, $S^* = S * \mathrm{Triv}^{\mathcal{M},0}$ is precisely the scheme we have described and that we wish to study here.

Note that by always rejecting, the AQECM scheme $\mathrm{Triv}^{\mathcal{M},0}$ is 0-tamper-evident. However, this comes at the cost of not achieving any non-trivial level of correctness as correctness requires accepting certain ciphertexts.

Lemma 2.67. For any alphabet $\mathcal{M}$, the scheme $\text{Triv}^{\mathcal{M},0}$ is 0-tamper evident.

Proof: As $D^{\mathcal{M},0}$ always rejects, the map $\overline{D^{\mathcal{M},0}} : \mathcal{L}(\mathsf{C}) \to \mathcal{L}(\mathsf{M})$ takes any linear operator to the zero operator $\mathbf{0}_{\mathsf{M}} \in \mathcal{L}(\mathsf{M})$. Moreover, for any other linear operator $L \in \mathcal{L}(\mathsf{A})$, it holds that $\mathbf{0}_{\mathsf{M}} \otimes L = \mathbf{0}_{\mathsf{M}} \otimes \mathbf{0}_{\mathsf{A}}$. Hence, for any tamper attack A and any messages m and $\tilde{m}$,

$$\frac{1}{2}\left\| \left(\left(\text{Tr}_{\mathsf{M}} \circ \overline{D^{\mathcal{M},0}} \right) \otimes \text{Id}_{\mathsf{A}} \right) \circ A \circ E(m - \tilde{m}) \right\|_1 = \frac{1}{2} \|\mathbf{0}_{\mathsf{A}} - \mathbf{0}_{\mathsf{A}}\|_1 = 0 \tag{2.217}$$

which yields the desired result. ∎

By Theorem 2.64 and the previous lemma, if S is δ-tamper evident, then S^* is $2\sqrt{\delta}$-tamper evident.[27] However, since $\text{Triv}^{\mathcal{M},0}$ is 1-correct, we cannot use Theorem 2.62 to show that the scheme S^* exhibits non-trivial correctness. Instead, we will directly prove that S^* is as correct as S. This is the content of the following theorem.

Theorem 2.68. Let $S = (K, E, D)$ be an ϵ-correct δ-tamper-evident scheme with messages $\mathcal{M}$ which admit a group operation denoted $*$. Then, $S * \text{Triv}^{\mathcal{M},0} = (K^*, E^*, D^*)$ is $2\sqrt{\delta}$-tamper-evident and ϵ-correct.

Proof: That $S * \text{Triv}^{\mathcal{M},0}$ is $2\sqrt{\delta}$-tamper evident follows immediately from Lemma 2.67 and Theorem 2.64. The generic correctness result of Theorem 2.62 is not sufficient in this case as it only yields $(1+\epsilon)$-correctness. However, direct computation yields

$$\begin{aligned}
\mathop{\mathbb{E}}_{k \leftarrow K^*(\varepsilon)} \langle m | \, \overline{D_k^*} \circ E_k^*(m) \, | m \rangle &= \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \mathop{\mathbb{E}}_{p \leftarrow \mathcal{M}} \langle m | \, \overline{D_k^*}(E_k(p) \otimes (p * m)) \, | m \rangle \\
&= \mathop{\mathbb{E}}_{k \leftarrow K(\varepsilon)} \mathop{\mathbb{E}}_{p \leftarrow \mathcal{M}} \langle p | \, \overline{D_k} \circ E_k(p) \, | p \rangle \\
&\geq \mathop{\mathbb{E}}_{p \leftarrow \mathcal{M}} (1 - \epsilon) \\
&\geq 1 - \epsilon
\end{aligned} \tag{2.218}$$

where the first equality follows by definition of K^*, E^*, and $\text{Triv}^{\mathcal{M},0}$, while the second equality from the observation that D_k^* will accept and produce the correct plaintext if and only if D accepts the first ciphertext and outputs p. ∎

For the purposes of our next two examples, we make this construction even more precise. Consider a perfectly correct δ-tamper-evident AQECM scheme S whose messages are a single bit, which is to say that $\mathcal{M} = \{0, 1\}$. Let $S^{\oplus} = S \oplus \text{Triv}^{\mathcal{M},0} = (K^{\oplus}, E^{\oplus}, D^{\oplus})$ be the scheme obtained from the construction of Definition 2.61 where the group operation

[27] We conjecture that S^* is actually δ-tamper evident.

is taken to be the exclusive-or $\oplus$. By Theorem 2.68, $S^{\oplus}$ is a perfectly correct $2\sqrt{\delta}$-tamper-evident AQECM scheme.

We now use $S^{\oplus}$ to show seperations between tamper evidence and two other notions: authentication and the encryption of quantum states. Note that by Theorem 2.22, we can instantiate $S^{\oplus}$ with a perfectly correct δ'-tamper-evident scheme S for any $\delta' > 0$. Hence, we can assume that $S^{\oplus}$ is perfectly correct and δ-tamper evident for any $\delta > 0$.

2.7.3.1 Tamper Evidence Does Not Imply Authentication or Robustness

The notion of robustness for AQECM schemes was originally stated in Broadbent and Islam's work on certified deletion [BI20], but it can be understood as a restriction of Barnum et al.'s notion of quantum authentication [BCG+02] from any arbitrary quantum states to only classical messages. In a nutshell, an AQECM scheme is robust if it is infeasible for an adversary to change a ciphertext in such a way that it will still be accepted by the decryption map, but that a different message will be recovered. We do not state a formal definition of this property.

As Gottesman noted, every quantum authentication scheme is itself a tamper-evident AQECM scheme [Got03]. Trivially, these are also robust AQECM schemes. It is also clear that there exist robust AQECMs which are not tamper evident. For example, simply consider any classical message authentication code which is information-theoretically secure (e.g.: [KL15, Sec. 4.6]). As the encodings produced by these message authentication codes are classical, it is infeasible to detect an adversary making and keeping a copy of them.

The above naturally leads to the following question: Does tamper evidence imply robustness? In the following, we sketch a proof that the answer is "no". There are tamper-evident schemes which are not robust.

Consider the AQECM scheme $S^{\oplus}$ as previously described in this section. Further consider the channel $\tilde{A} : \mathcal{L}(\mathsf{C} \otimes \mathbb{C}^{\{0,1\}}) \to \mathcal{L}(\mathsf{C} \otimes \mathbb{C}^{\{0,1\}})$ defined by

$$\rho \mapsto (I_{\mathsf{C}} \otimes X)\, \rho\, (I_{\mathsf{C}} \otimes X) \tag{2.219}$$

where $X = |0\rangle\langle 1| + |1\rangle\langle 0|$ is the bit-flip unitary operator. For any plaintext $b \in \{0, 1\}$, we see that

$$\begin{aligned} \tilde{A} \circ E_k^{\oplus}(m) &= \tilde{A}\left(\mathop{\mathbb{E}}_{p \leftarrow \{0,1\}} E_k(p) \otimes (p \oplus b) \right) \\ &= \mathop{\mathbb{E}}_{p \leftarrow \{0,1\}} E_k(p) \otimes (p \oplus b \oplus 1) \\ &= E_k^{\oplus}(b \oplus 1). \end{aligned} \tag{2.220}$$

As we assume that S is perfectly correct, $S^{\oplus}$ is also perfectly correct. Hence, for any key k

and message $b \in \{0,1\}$, we have

$$\langle b \oplus 1| \overline{D^{\oplus}}_k \circ \tilde{A} \circ E_k^{\oplus}(b) |b \oplus 1\rangle = 1 \tag{2.221}$$

which is to say that an adversary using $\tilde{A}$ can change the encoded plaintext undetected and with certainty. It follows $S^{\oplus}$ cannot offer any non-trivial level of robustness.

2.7.3.2 Tamper Evidence Does Not Imply the Encryption of Quantum States

While AQECM schemes generally only guarantee correctness and various notions of security for *classical* messages, their encryption and decryption maps remain quantum channels. Hence, we can still inquire about their behaviours with respect to arbitrary quantum states.

We claim without further proof that $S^{\oplus}$ is also perfectly correct with respect to any *quantum state*, in the sense that $D_k^{\oplus} \circ E_k^{\oplus}(\rho) = \rho \otimes |1\rangle\langle 1|$ for all keys k and quantum states $\rho \in \mathbb{C}^{\{0,1\}}$. This follows directly from the details of the construction of $S^{\oplus}$ from Definition 2.61, in particular from the fact that the group operation is computed coherently.

On the other hand, recall that $S^{\oplus}$ is $7\sqrt[4]{\delta}$-encrypting for classical messages due to Theorem 2.30 and the fact that it is $2\sqrt{\delta}$-tamper evident. However, we claim that it offers no non-trivial level of security as an encryption scheme for arbitrary quantum states. This results from the fact that the classical one-time-pad fails to change certain quantum states.

More precisely, going through the details of the construction in Definition 2.61, we see that the unitary $U_{\oplus}$ which coherently computes the group operation is, here, simply the controlled-not linear operator CNOT which maps $|b, b'\rangle$ to $|b, b \oplus b'\rangle$ for any $b, b' \in \{0,1\}$. Thus, for any key k and state ρ, we have that

$$E_k^{\oplus}(\rho) = \mathop{\mathbb{E}}_{p \leftarrow \{0,1\}} (E_k \otimes \mathrm{Id}_{\mathbb{C}^{\{0,1\}}}) \,\mathrm{CNOT}\,(p \otimes \rho)\,\mathrm{CNOT}^{\dagger} = \mathop{\mathbb{E}}_{p \leftarrow \{0,1\}} E_k(p) \otimes X^p \rho X^p \tag{2.222}$$

where $X = |0\rangle\langle 1| + |1\rangle\langle 0|$ is once again the bit-flip unitary operator.

Consider now the states $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Note that both resulting density operators $|+\rangle\langle +|$ and $|-\rangle\langle -|$ are invariant under conjugation by X. Hence, for any two keys k and k', we have that

$$\begin{aligned} \frac{1}{2}\left\| E_k^{\oplus}(|+\rangle\langle +|) - E_{k'}^{\oplus}(|-\rangle\langle -|) \right\|_1 &= \frac{1}{2}\left\| \mathop{\mathbb{E}}_{p \leftarrow \{0,1\}} E_k(p) \otimes |+\rangle\langle +| - \mathop{\mathbb{E}}_{p \leftarrow \{0,1\}} E_{k'}(p) \otimes |-\rangle\langle -| \right\|_1 \\ &\geq \frac{1}{2}\| |+\rangle\langle +| - |-\rangle\langle -| \|_1 \\ &= 1 \end{aligned} \tag{2.223}$$

where the inequality follows by tracing out the ciphertext from S and the last equality from

Lemma 2.4 and $\langle +|-\rangle = 0$. In other words, the encodings of $|+\rangle$ and $|-\rangle$ under $E_k^{\oplus}$ for any keys remain perfectly distinguishable. Hence, this AQECM scheme does not offer any non-trivial level of security as an encryption scheme for arbitrary quantum states.

Chapter 3

Secure Software Leasing*

Chapter Abstract

Quantum cryptography is known for enabling functionalities that are unattainable using classical information alone. Recently, *Secure Software Leasing (SSL)* has emerged as one of these areas of interest. Given a target circuit C from a circuit class, SSL produces an encoding of C that enables a recipient to evaluate C, and also enables the originator of the software to *verify* that the software has been *returned* — meaning that the recipient has relinquished the possibility of any further use of the software. Clearly, such a functionality is unachievable using classical information alone, since it is impossible to prevent a user from keeping a copy of the software. Recent results have shown the achievability of SSL using quantum information for a class of functions called *compute-and-compare* (these are a generalization of the well-known *point functions*). These prior works, however all make use of setup or computational assumptions. Here, we show that SSL is achievable for compute-and-compare circuits *without any assumptions*.

Our technique involves the study of *quantum copy-protection*, which is a notion related to SSL, but where the encoding procedure inherently *prevents* a would-be quantum software pirate from *splitting* a single copy of an encoding for C into two parts, each of which enables a user to evaluate C. We show that point functions can be copy-protected *without any assumptions*, for a novel security definition involving one honest and one malicious evaluator; this is achieved by showing that from any quantum message authentication code, we can derive such an *honest-malicious* copy-protection scheme. We then show that a generic honest-malicious copy-protection scheme implies SSL; by prior work, this yields SSL for compute-and-compare functions.

*This chapter reflects work which was completed in collaboration with Anne Broadbent, Stacey Jeffery, Supartha Podder, and Aarthi Sundaram and which has been published in the *Proceedings of the 19th International Conference on the Theory of Cryptography* (TCC 2021) [BJL+21a]. It has been edited from its published version for inclusion in this thesis, including by the incorporation of technical details found in the full version [BJL+21b]. doi:10.1007/978-3-030-90459-3_4.

Chapter Contents

3.1 Introduction

One of the defining features of quantum information is the *no-cloning* principle, according to which it is not possible, in general, to take an arbitrary quantum state and produce two copies of it [Par70, WZ82, Die82]. This principle is credited for many of the feats of quantum information in cryptography, including quantum key distribution (QKD) [BB84] and quantum money [Wie83]. (For a survey on quantum cryptography, see [BS16]).

The quantum no-cloning principle tells us that, in a certain sense, quantum information behaves more like a *physical* object than a digital one: there are situations where quantum information can be distributed and used, but it cannot be duplicated. One such example is quantum money [Wie83], in which a quantum system is used to encode a very basic type of information — the ability to verify authenticity. However, we can envisage quantum encodings that achieve richer levels of applicability. We thus define a hierarchy of "uncloneable" objects, where the basic notion provides only authenticity, and the topmost notion provides *functionality*.

The uncloneability hierarchy includes:

- **Authenticity.** In the first and most basic level, the uncloneability property can be used to *verify* authenticity.
- **Information.** Next, *information* is made uncloneable, meaning that there is some underlying data that can be decoded, but there are limitations on the possibility of copying this data while it is encoded.
- **Functionality.** At the top level of the hierarchy, a *functionality* is made uncloneable, meaning that there are limitations on how many users can simultaneously evaluate the functionality.

For both the case of *information* and *functionality*, a type of *verification* is possible but optional: this verification is a way to confirm that a message or functionality is returned; after such verification is confirmed, further reading/use of the encoded information is impossible.

We emphasize that none of the concepts in the hierarchy are possible in a conventional digital world, since classical information can be copied. Thus the hierarchy is best understood intuitively at the level of a physical analogy where, for example, authenticity is verified by physical objects and functionalities are distributed in *hardware* devices.

Achieving the hierarchy. We summarize below the known results on achievability of the hierarchy.

1. The *authenticity* level of the hierarchy is the most well-understood, and it includes quantum money [Wie83], quantum coins [MS10], and publicly-verifiable quantum money [AC12].

2. Next, the *information* level includes *tamper-evident* encryption [Got03] and *uncloneable encryption* [BL20].

   We comment here on a technique of Gottesman [Got03] that is relevant to our work. In [Got03], it is shown that tamper-evident encryption can be achieved using the primitive of *Quantum Message Authentication (QMA)* [BCG+02] — in other words, the *verification* of quantum authentication not only gives a guarantee that the underlying plaintext is intact, but *also* that no adversary can gain information on the plaintext, *even if the key is revealed.* Uncloneable encryption is a notion that is complementary to tamper-evident encryption, and it focuses on *preventing* duplication of an underlying plaintext. In [BL20], it is shown to be achievable in the Quantum Random Oracle Model (QROM).

3. Finally, the *functionality* level of the hierarchy was first discussed in terms of *quantum copy-protection* by Aaronson [Aar09]: here, a quantum encoding allows the evaluation of a function on a chosen input, but in a way that the number of *simultaneous* evaluations is limited. In [Aar09], copy-protection for a class of functions is shown to exist assuming a quantum oracle; this was improved (for a more restricted family of circuits) to a *classical* oracle in [ALL+21]. Further work in [CMP24] improved the assumption to the QROM.[2]

   A related concept, also at the functionality level of the hierarchy, was recently put forward: *Secure Software Leasing (SSL)*, where a quantum encoding allows evaluation of a circuit, while also enabling the originator to verify that the software is *returned* (meaning that it can no longer be used to compute the function). SSL was first studied by Ananth and La Placa [AL21][3] where it was shown that SSL could be achieved for *searchable compute-and-compare circuits*[4]; in order to achieve their result (which is with respect to an *honest* evaluation), they make use of strong cryptographic assumptions: quantum-secure subspace obfuscators, a common reference string, and the difficulty of the Learning With Errors (LWE) problem. Further work [CMP24]

[2] This is an improvement, as a QROM does not depend on the circuit to be computed.

[3] Two notions are actually introduced in [AL21]: *finite-term* and *infinite-term* SSL. In this work, SSL refers to finite-term SSL.

[4] A circuit class $\mathcal{C}$ is a *compute-and-compare* circuit class if for every circuit in $\mathcal{C}$, there is an associated circuit C and string α such that on input x, the circuit outputs 1 if and only if $C(x) = \alpha$. *Searchability* refers to the fact that there is an efficient algorithm that, on input $C \in \mathcal{C}$, outputs an x such that $C(x) = \alpha$. From this point on, *searchability* is an implicit assumption throughout this work.

improved the result on achievability for the same class of circuits, this time against *malicious evaluations*, and in the QROM. Very recently, [KNY21] showed the achievability of SSL, based on LWE, against honest evaluators, and for classes of functions beyond *evasive* functions.[5]

3.1.1 Summary of Contributions

Due to their foundational role in the study of uncloneability as well as for potential applications, SSL and copy-protection are emerging as important elements of quantum cryptography. In this work, we solve two important open problems related to SSL and quantum copy-protection.

Secure Software Leasing. We show how to construct an SSL scheme for compute-and-compare circuits against a malicious evaluator. Ours is the first scheme that makes no assumptions — there are no setup assumptions, such as a QROM or a common reference string, and no computational assumptions, such as one-way functions or the LWE assumption. We thus show for the first time that SSL is achievable, unconditionally. A compromise we make in order to achieve this is the use of a natural but weaker notion of correctness *with respect to a distribution*. We note that general SSL was shown to be impossible [AL21], and that [Aar09] mentions how *learnable* functions cannot be copy protected. It is thus natural that we focus our efforts on achieving SSL for compute-and-compare circuits, which is a family of functions that is not learnable.

In more detail, we follow the security notion of [CMP24], which postulates a game between a challenger, and a pirate Pete. Upon sampling a circuit from a given distribution, the challenger encodes the circuit and sends it to Pete. Pete then produces a register that he returns to the challenger who performs a *verification*; upon successful verification, we continue the game (otherwise, we abort), by presenting to Pete a challenge input $x \in \{0,1\}^n$ (chosen according to a given distribution). The scheme is ϵ-*secure* if we can bound the probability that the game does not abort and that Pete correctly evaluates the circuit on the challenge input x to be within ϵ of his trivial guessing probability. Here, trivially guessing means that Pete answers the challenge by seeing only x, i.e., disregarding all other information obtained by interacting with the challenger. Thus, security is defined relative to the distribution on the circuits and on the challenges. For SSL, η-correctness is defined with respect to an input distribution, and means that, up to some error term η, the honest evaluation on an encoded circuit produces the correct outcome, *in expectation*.[6]

[5] Informally, *evasive* functions are the class of functions such that it is hard to find an accepting input, given only black-box access to a functions. Note that compute-and-compare functions are evasive.

[6] This notion is weaker than the more common notion of correctness that holds for *all* inputs. However, in

We show how to achieve SSL with respect to the uniform distribution on point functions, and the challenge distribution which samples uniformly from the distribution where the correct response is 0 or 1 with equal probability and which we denote $T_p^{(1/2)}$. Our technique is a reduction from SSL to *honest-malicious* copy-protection, as well as a new construction for quantum honest-malicious copy-protection (with respect to essentially the same distributions as stated above). Prior work noted, informally, that copy-protection implies SSL [AL21]. Here, we formally show that our new and weaker (and thus easier-to-achieve) notion of copy-protection, discussed in the next paragraph, implies SSL. Our work focuses on achieving SSL for point functions; by applying our result with [CMP24], this implies SSL for compute-and-compare circuits.

Honest-Malicious Copy-Protection. We define a new security model for copy-protection: *honest-malicious* copy-protection.[7] Here, we consider a game between a challenger, a pirate (Pete), and two evaluators. Importantly, the first evaluator, Bob, is *honest* (meaning that he will execute the legitimate evaluation procedure) and the second evaluator, Charlie, is *malicious*. In the setting of copy-protection, we want to bound the probability that, after each receiving a quantum register from Pete, who takes as input a single copy protected program, the two evaluators (who cannot communicate), are *both* able to correctly evaluate the encoded circuit. Following [CMP24], this is formalized by a game, parameterized by a distribution on the input circuits and corresponding challenge distributions on pairs of n-bit strings. A challenger samples a circuit, encodes it using the copy-protection scheme and sends the encoding to Pete who creates the two registers. Then a challenge pair (x_1, x_2) is sampled from the corresponding challenge distribution; Bob receives x_1 while Charlie receives x_2. They *win* if they each produce the correct output of the original circuit evaluated on x_1 and x_2, respectively. An honest-malicious copy-protection scheme is ϵ-*secure* for the given distributions if the probability that the evaluators win the game is within ϵ of the success probability of the trivial strategy that is achievable when Bob gets the full encoding and Charlie guesses to the best of his ability without interacting with Pete. As in the case of SSL, η-correctness for copy-protection is defined with respect to an input distribution, and means that, up to some constant η, the honest evaluation on an encoded circuit produces the correct outcome, *in expectation*.[8]

Section 3.4, we give evidence that achieving this stronger notion of correctness may be possible, by showing that for the standard notion of copy-protection (against two malicious evaluators), correctness in expectation implies worst-case correctness, which would then imply worst-case correctness for SSL.

[7] This is a stronger notion of security than *infinite term SSL* as defined in [AL21], which is a form of copy-protection where both evaluators are honest, and is achieved in [AL21] under strong assumptions.

[8] Similarly to the discussion in Footnote 6, this is a weaker notion than the more common notion of correctness for *all* inputs.

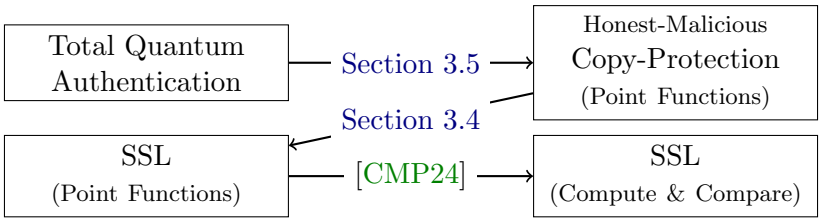


Figure 3.1: Relations between various notions considered in this chapter.

We establish the relevance of honest-malicious copy-protection by showing that, for general functions, honest-malicious copy-protection implies SSL.

In order to complete our main result, we show how to achieve honest-malicious copy-protection for point functions, where the challenge distribution is $(T_p^{(1/2)} \times T_p^{(1/2)})$, and correctness is also with respect to $T_p^{(1/2)}$. To the best of our knowledge, this is the first unconditional copy-protection scheme; via the previously discussed reduction, it yields the first SSL scheme without assumptions. See Figure 3.1 for a pictorial representation of the sequence of results. Our idea is to generically use a *quantum message authentication scheme (QAS)* that satisfies the *total authentication* security property [GYZ17]. Briefly, a QAS is a private-key scheme with an encoding and decoding procedure such that the probability that the decoding accepts *and* the output of the decoding in *not* the original message is small. Security of a *total* QAS is defined in terms of the existence of a *simulator* that reproduces the auxiliary register that an adversary has after attacking an encoded system, *whenever* the verification accepts. An important feature of a total QAS is that essentially no information about the key is leaked if the receiver accepts the authenticated state.

The main insight for the construction of honest-malicious copy-protection for point functions from a total QAS is to associate the key to the QAS with the point p in the point function. A copy protected program is thus an encoding of an arbitrary (but fixed) state $|\psi\rangle$ via a total QAS, using p as the key. Given p', the evaluation of the point function encoding is the QAS verification *with the key* p'. We thus get correctness in the case $p' = p$ from the correctness of the QAS; correctness in expectation for $p' \neq p$ follows with a bit more work. Importantly, the *total* security property of the QAS gives us a handle on the auxiliary register that an adversary holds, *in the case that the verification accepts*. Since Bob is honest, his evaluation corresponds to the QAS verification map; in the case that Bob gets the challenge $x_1 = p$, we use the properties of the total QAS to reason about Charlie's register, and we are able to show that Charlie's register cannot have much of a dependence on p, which is to say that Charlie's outcome is necessarily independent of p. This is sufficient to conclude that Bob and Charlie cannot win the copy-protection game for a uniform point

with probability much better than the trivial strategy in which Charlie makes an educated guess, given the challenge x_2. We note that total authentication is known to be satisfied by a scheme based on 2-designs [AM17], as well as by the *strong trap code* [DS18]. Putting all of the above together, we obtain our main result, which is an explicit SSL scheme for point functions $P_p : \{0,1\}^n \to \{0,1\}$ which is $O(2^{-\frac{n}{16}})$-correct (on average) and $O(2^{-\frac{n}{32}})$-secure, under uniform sampling of p and where the challenge distribution is $T_p^{(1/2)}$.[9] We note the similarity between our approach for achieving honest-malicious copy-protection and the approach in [Got03] to achieve tamper-evident encryption based on quantum authentication schemes. We also mention a similarity with the blueprint in [CMP24], which also produces a copy protected program starting from a private-key encryption scheme (in this case, the one of [BL20]), associates a point with the key, and uses a type of verification of the integrity of the plaintext after decryption as the evaluation method.

Too good to be true? We emphasize that our results require no assumptions at all, which is to say that the result is in the standard model (as opposed to, say, the quantum random oracle model), and does not rely on any assumption on the computational power of the adversary. That either copy-protection or SSL should be achievable in this model is very counter-intuitive, hence we explain here how we circumvent related impossibility results. In short, our work strikes a delicate balance between correctness and security, in order to achieve the best of both worlds.

Prior work [Aar09] defines quantum copy-protection assuming the adversary is given *multiple identical* copies of the same copy protected state. Under this model, it is possible to show how an unbounded adversary can distinguish between the copy protected programs for different functions [Aar09], which makes unconditionally secure copy-protection impossible. In our scenario, we allow only a *single* copy of the program state, hence this reasoning is not applicable.

Next, consider a scheme for either copy-protection or SSL which is *perfectly correct*, meaning that the outcome of the evaluation procedure is deterministic on a well formed input. Clearly, such a scheme cannot be secure against computationally unbounded adversaries since, *in principle*, there is a sequence of measurements that an unbounded adversary can perform (via purification and rewinding), in order to perfectly obtain the truth table of the function. We conclude that perfectly correct schemes cannot satisfy our notion of unconditional security for copy-protection.

We note that our scheme is, by design, not perfectly correct. This can be seen by reasoning about the properties of the QAS: in any QAS, it is necessary that, for a fixed

[9] This is achieved by instantiating the copy-protection scheme from Section 3.5 with a total quantum authentication scheme given by Lemma 3.4 and using it in the SSL construction of Section 3.4.3.

encoding with key k, there are a number of keys on which the verification accepts. The reason why this is true is similar to the argument above regarding perfect correctness: if this were not true, then the QAS (which is defined with respect to unbounded adversaries) would not be secure, since an adversary could in principle find k by trying all keys (coherently, so as to not disturb the quantum state) until one accepts. Somewhat paradoxically, it is this imperfection in the correctness that thus allows the unconditional security. Another way to understand the situation is that the honest evaluation in our copy-protection (or SSL) scheme will unavoidably slightly damage the quantum encoding (even if performed coherently). In a brute-force attack, these errors necessarily accumulate to the point of rendering the program useless, and therefore the brute-force attack fails.

3.1.2 Open Problems

Our work leaves open a number of interesting avenues for future work. For instance:

1. Could we show the more standard notion of correctness of our scheme, that is, correctness with respect to *any* distribution?
2. Is unconditional SSL achievable for a richer class of functions?
3. Can our results on copy-protection be extended to hold against *two* malicious evaluators?

In Section 3.4, we show that 1 and 3 are related, by establishing that a point function copy-protection scheme that is secure against two malicious evaluators and satisfies average correctness can be turned into a scheme that also satisfies the more standard notion of correctness.

3.1.3 Acknowledgements

We would like to thank Christian Majenz and Martti Karvonen for related discussions. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0083, Canada's NFRF and NSERC, an Ontario ERA, and the University of Ottawa's Research Chairs program. SJ is a CIFAR Fellow in the Quantum Information Science program.

3.1.4 Outline

The remainder of this chapter is structured as follows. In Section 3.2, we give background information on notation, basic notions, and quantum message authentication. In Section 3.3,

we define correctness and security for quantum copy-protection and SSL. In Section 3.4, we show the connection between malicious-malicious security and standard correctness, as well as links between honest-malicious copy-protection and SSL. Finally, our main technical construction of honest-malicious copy-protection from any total QAS is given in Section 3.5.

3.2 Preliminaries

3.2.1 Notation

All Hilbert spaces in this work are complex and of finite dimension. We will usually denote a Hilbert space using a sans-serif font such as S or H. We will often omit the tensor symbol when taking the tensor product of two Hilbert spaces, i.e.: $\mathsf{A} \otimes \mathsf{B} = \mathsf{AB}$. We use the Dirac notation [Dir39] throughout, which is to say that $|\psi\rangle \in \mathsf{H}$ denotes a vector and $\langle\psi| : \mathsf{H} \to \mathbb{C}$ denotes the corresponding linear map in the dual space. Finally, Hilbert spaces may be referred to as "registers", acknowledging that they sometimes model physical objects which may be sent, kept, discarded, etc., by participants in quantum information processing tasks.

The set of linear operators, unitary operators, and density operators on a Hilbert space H are denoted by $\mathcal{L}(\mathsf{H})$, $\mathcal{U}(\mathsf{H})$, and $\mathcal{D}(\mathsf{H})$ respectively. A linear operator may be accompanied by a subscript indicating the Hilbert space on which it acts. This will be useful for bookkeeping and to omit superfluous identities. For example, if $L_{\mathsf{A}} \in \mathcal{L}(\mathsf{A})$ and $|\psi\rangle_{\mathsf{AB}} \in \mathsf{AB}$, then $L_{\mathsf{A}} |\psi\rangle_{\mathsf{AB}} = (L_{\mathsf{A}} \otimes I_{\mathsf{B}}) |\psi\rangle_{\mathsf{AB}}$. Similarly, we will often omit superfluous identities when considering tensor products of CPTP maps.

We recall [Wat18] that the trace norm, or Schatten 1-norm, of a linear operator is given by $\|X\|_1 = \max_{U \in \mathcal{U}(\mathsf{A})} |\langle U|X\rangle|$ where $\langle U|X\rangle = \mathrm{Tr}\left[U^\dagger X\right]$. The trace distance between two linear operators is then given by $\Delta(X,Y) = \frac{1}{2}\|X - Y\|_1$. If $\Delta(X,Y) \leq \epsilon$, we write $X \approx_\epsilon Y$. We also give a technical lemma pertaining to the trace distance between bipartite states of a particular form.[10]

Lemma 3.1. Let A and B be Hilbert spaces and $\{|\psi_j\rangle\}_{j\in J} \subseteq \mathsf{A}$ be a set of orthonormal vectors. Then, for any sets of linear operators $\{X_j\}_{j\in J}$ and $\{Y_j\}_{j\in J}$ on B, we have that

$$\Delta\left(\sum_{j\in J} |\psi_j\rangle\langle\psi_j| \otimes X_j, \sum_{j\in J} |\psi_j\rangle\langle\psi_j| \otimes Y_j\right) = \sum_{j\in J} \Delta\left(X_j, Y_j\right). \tag{3.1}$$

For a distribution D on a set S, we will use the notation $x \leftarrow D$ to denote that x is sampled from D, and $D(x)$ to denote the probability that a given $x \in S$ is sampled. If S is

[10] This lemma is equivalent to Lemma 2.6 from the previous chapter. A proof is provided there.

finite, for a given map $f : S \to \mathbb{C}$ we write

$$\mathop{\mathbb{E}}_{x \leftarrow D} f(x) = \sum_{x \in S} D(x) f(x). \tag{3.2}$$

If no distribution is specified or clear from context, we assume a uniform distribution, which is to say that $\mathbb{E}_x f(x) = |S|^{-1} \cdot \sum_{x \in S} f(x)$.

Two classes of functions will be of particular interest in this work: point functions and compute-and-compare functions. For any $p \in \{0,1\}^n$, the map $P_p : \{0,1\}^n \to \{0,1\}$ satisfying $P_p(x) = 1 \iff x = p$ is called the point function for p. For any function $f : \{0,1\}^n \to \{0,1\}^m$ and bit string $y \in \{0,1\}^m$, the map $\mathsf{CC}^f_y : \{0,1\}^n \to \{0,1\}$ which satisfies $\mathsf{CC}^f_y(x) = 1 \iff f(x) = y$ is the compute-and-compare function of f and y. We note that point functions can be seen as compute-and-compare functions where f is the identity.

An n-bit Boolean circuit is a circuit taking as input an element of $\{0,1\}^n$ and producing as output a single bit. Throughout this work, we will denote a family of Boolean circuits on n bits as $\mathcal{C}$.

3.2.2 Quantum Authentication

We recall the definition of a total quantum authentication scheme [GYZ17] and highlight a few properties of such schemes.

Definition 3.2. An authentication scheme $\mathsf{QAS} = \{(\mathsf{QAS.Auth}_k, \mathsf{QAS.Ver}_k)\}_{k \in \mathcal{K}}$ for the Hilbert space M is a pair of keyed CPTP maps

$$\mathsf{QAS.Auth}_k : \mathcal{L}(\mathsf{M}) \to \mathcal{L}(\mathsf{Y}) \quad \text{and} \quad \mathsf{QAS.Ver}_k : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{MF}) \tag{3.3}$$

where F admits $\{|\mathrm{Acc}\rangle, |\mathrm{Rej}\rangle\}$ as an orthonormal basis. Moreover, these maps are such that for all states $\rho \in \mathcal{D}(\mathsf{M})$ and all keys $k \in \mathcal{K}$ we have that

$$\mathsf{QAS.Ver}_k \circ \mathsf{QAS.Auth}_k(\rho) = \rho \otimes |\mathrm{Acc}\rangle\langle \mathrm{Acc}| . \tag{3.4}$$

We assume throughout this work that the keys for an authentication scheme are generated uniformly at random.

To facilitate our analysis, we will make the same simplifying assumptions as in [GYZ17] on any quantum authentication scheme considered in this work:

1. We assume that $\mathsf{QAS.Auth}_k$ can be modeled by an isometry. Specifically, we assume

that

$$\mathsf{QAS.Auth}_k(\rho) = A_k \rho A_k^\dagger \tag{3.5}$$

for some isometry $A_k \in \mathcal{L}(\mathsf{M}, \mathsf{Y})$.

2. For all keys $k \in \mathcal{K}$, we have that $A_k A_k^\dagger$ is the projector onto the image of A_k. In other words, this operator projects onto valid authenticated states for the key k. We then assume that $\mathsf{QAS.Ver}_k$ is given by the map

$$\rho \mapsto A_k^\dagger \rho A_k \otimes |\mathrm{Acc}\rangle\langle\mathrm{Acc}| + \mathrm{Tr}\left[\left(I - A_k A_k^\dagger\right)\rho\right] \cdot \frac{I}{\dim(\mathsf{M})} \otimes |\mathrm{Rej}\rangle\langle\mathrm{Rej}| \,. \tag{3.6}$$

In other words, $\mathsf{QAS.Ver}_k$ verifies if the state is a valid encoded state. If it is, then it inverts the authentication procedure and adds an "accept" flag. If it is not, then it outputs the maximally mixed state and adds a "reject" flag.

Finally, we will also define the map $\mathsf{QAS.Ver}'_k : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{M})$ by

$$\rho \mapsto (I_\mathsf{M} \otimes \langle\mathrm{Acc}|_\mathsf{F})\, \mathsf{QAS.Ver}_k(\rho)\, (I_\mathsf{M} \otimes |\mathrm{Acc}\rangle_\mathsf{F}) = A_k^\dagger \rho A_k. \tag{3.7}$$

Essentially, this map outputs a subnormalized state corresponding to the state of the message register M conditioned on the verification procedure accepting the state. In particular, note that the probability that the verification procedure accepts the state ρ when using the key k is given by $\mathrm{Tr}\left(\mathsf{QAS.Ver}'_k(\rho)\right)$.

Definition 3.2 does not make any type of security guarantee on an authentication scheme. It only specifies a syntax, Equation 3.3, and a correctness guarantee, Equation 3.4. The following definition describes the security guarantee of an ϵ-total quantum authentication scheme. Note that this security definition differs from some early notions of security for quantum authentication schemes [BCG+02, DNS12].

Definition 3.3. An authentication scheme QAS is an ϵ-total authentication scheme if for all CPTP maps $\Phi : \mathcal{L}(\mathsf{YZ}) \to \mathcal{L}(\mathsf{YZ})$ there exists a completely positive trace non-increasing map $\Psi : \mathcal{L}(\mathsf{Z}) \to \mathcal{L}(\mathsf{Z})$ such that

$$\mathop{\mathbb{E}}_{k\in\mathcal{K}} |k\rangle\langle k| \otimes \mathsf{QAS.Ver}'_k \circ \Phi \circ \mathsf{QAS.Auth}_k(\rho) \approx_\epsilon \mathop{\mathbb{E}}_{k\in\mathcal{K}} |k\rangle\langle k| \otimes \mathsf{QAS.Ver}'_k \circ \Psi \circ \mathsf{QAS.Auth}_k(\rho) \tag{3.8}$$

for any state $\rho \in \mathcal{D}(\mathsf{MZ})$.

In short, a scheme is an ϵ-total authentication scheme if, conditioned on $\mathsf{QAS.Ver}$ accepting, then the adversarial action $\Phi : \mathcal{L}(\mathsf{YZ}) \to (\mathsf{YZ})$ which acted on the state space of the authenticated system Y and some additional non-authenticated register Z could have been simulated by an action Ψ only acting on the non-authenticated register.

A key difference between the "total" security definition given in [GYZ17] and previous security definitions for authentication schemes is the explicit $|k\rangle\langle k|$ state which appears in Equation 3.8. This key register will be used, with the help of Lemma 3.1, in some of our technical arguments, such as the proof of Lemma 3.25.

Note that our discussion, unlike the one in [GYZ17], omits adding another register S to model all other information that a sender and receiver could share as part of a larger protocol but which is not directly implicated in the authentication scheme. Such a register is not needed in our analysis.

Next, we give an existence lemma for total quantum authentication schemes satisfying certain parameters. This lemma not only ensures that the authentication-based construction we later discuss in Section 3.5 can indeed be instantiated, but it also provides an idea on the level of correctness and security it can acheive.

Lemma 3.4. For any strictly positive integers n and k, there exists a $\left(5 \cdot 2^{\frac{5n-k}{16}}\right)$-total quantum authentication scheme on n qubits with key set $\{0,1\}^k$.

The proof is given in the supplemental material for this chapter, specifically in Section 3.6.1, as it is somewhat orthogonal to the main ideas we pursue in this chapter. It essentially follows from a theorem describing how unitary 2-designs (as introduced in [DCEL09]) can be used to construct total quantum authentication schemes [AM17] and then choosing a suitable unitary 2-design [CLLW16]. A few additional technical arguments are needed to ensure that the key set is precisely the set of bit strings of a given length.

Finally, we give a lemma which upper bounds the probability that any fixed state is accepted by the verification procedure, when averaged over all possible keys. This allows us to make statements on what happens if an authenticated state is verified with the wrong key—a scenario which is not usually considered for authentication schemes but which is needed in this work. Intuitively, no quantum authentication scheme can admit such a fixed state ρ that is accepted with high probability over all keys, since otherwise an adversary could insert such a ρ in place of any authenticated message, and this modification would go undetected with high probability.

Lemma 3.5. Let QAS be an ϵ-total authentication scheme on the Hilbert space M of dimension greater than or equal to 2. Then, for any $\rho \in \mathcal{D}(\mathsf{Y})$, we have that

$$\mathop{\mathbb{E}}_{k\in\mathcal{K}} \mathrm{Tr}\left[\mathsf{QAS.Ver}'_k(\rho)\right] \leq 2\epsilon. \tag{3.9}$$

Before proceeding to the proof of Lemma 3.5, we will need a small lemma which essentially states that orthogonal states are mapped to orthogonal states by quantum authentication schemes.

Lemma 3.6. Let QAS be an authentication scheme. Then,

$$\Delta(\rho,\sigma)=1 \implies \Delta(\mathsf{QAS.Auth}_k(\rho),\mathsf{QAS.Auth}_k(\sigma))=1 \tag{3.10}$$

for all $k\in\mathcal{K}$ and all states $\rho,\sigma\in\mathcal{D}(\mathsf{M})$.

Proof: Since the trace distance is contractive under CPTP maps, we have that

$$\begin{aligned}
&\Delta(\mathsf{QAS.Auth}_k(\rho),\mathsf{QAS.Auth}_k(\sigma))\\
\geq&\Delta(\mathrm{Tr}_{\mathsf{F}}\circ\mathsf{QAS.Ver}_k\circ\mathsf{QAS.Auth}_k(\rho),\mathrm{Tr}_{\mathsf{F}}\circ\mathsf{QAS.Ver}_k\circ\mathsf{QAS.Auth}_k(\sigma))\\
\geq&\Delta(\rho,\sigma)\\
=&1.
\end{aligned} \tag{3.11}$$

Noting that $\Delta(\mathsf{QAS.Auth}_k(\rho),\mathsf{QAS.Auth}_k(\sigma))\leq 1$ completes the proof. ∎

We can now give the proof of Lemma 3.5.

Proof: Let $\mathsf{Z}\simeq\mathsf{Y}$, and define the attack $\Phi:\mathcal{L}(\mathsf{YZ})\to\mathcal{L}(\mathsf{YZ})$ by

$$\xi\mapsto \mathrm{Swap}_{\mathsf{YZ}}\left(\mathrm{Tr}_{\mathsf{Z}}(\xi)\otimes\rho\right). \tag{3.12}$$

In other words, an adversary implementing this attack will keep in memory the authenticated state sent by the sender and will give the receiver the state ρ.

As the authentication scheme is ϵ-total, there exists a completely positive trace non-increasing map $\Psi:\mathcal{L}(\mathsf{Z})\to\mathcal{L}(\mathsf{Z})$ such that

$$\mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\mathsf{QAS.Ver}'_k\circ\Phi\circ\mathsf{QAS.Auth}_k(\sigma)\approx_\epsilon \mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\mathsf{QAS.Ver}'_k\circ\Psi\circ\mathsf{QAS.Auth}_k(\sigma) \tag{3.13}$$

for all states $\sigma\in\mathcal{D}(\mathsf{MZ})$. In particular, consider separable states of the form $\sigma_{\mathsf{MZ}}=\tau_{\mathsf{M}}\otimes\tau'_{\mathsf{Z}}$. For such states, we have that

$$\mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\mathsf{QAS.Ver}'_k\circ\Phi\circ\mathsf{QAS.Auth}_k(\sigma)=\mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\mathsf{QAS.Ver}'_k(\rho)\otimes\mathsf{QAS.Auth}_k(\tau) \tag{3.14}$$

and

$$\mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\mathsf{QAS.Ver}'_k\circ\Psi\circ\mathsf{QAS.Auth}_k(\sigma)=\mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\tau\otimes\Psi(\tau') \tag{3.15}$$

so the security guarantee yields

$$\mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\mathsf{QAS.Ver}'_k(\rho)\otimes\mathsf{QAS.Auth}_k(\tau)\approx_\epsilon \mathop{\mathbb{E}}_{k\in K}|k\rangle\langle k|\otimes\tau\otimes\Psi(\tau'). \tag{3.16}$$

Since the trace distance is contractive under CPTP maps, we can trace out M, the message register, to obtain

$$\mathop{\mathbb{E}}_{k\in K} |k\rangle\langle k| \otimes \mathrm{Tr}\left[\mathsf{QAS.Ver}'_k(\rho)\right] \cdot \mathsf{QAS.Auth}_k(\tau) \approx_\epsilon \mathop{\mathbb{E}}_{k\in K} |k\rangle\langle k| \otimes \Psi(\tau'). \tag{3.17}$$

Using the triangle inequality and two instances of the above equation, once with $\tau = |\psi\rangle\langle\psi|$ and once with $\tau = |\phi\rangle\langle\phi|$ for orthogonal $|\psi\rangle$ and $|\phi\rangle$, we find that

$$\begin{aligned} &\mathop{\mathbb{E}}_{k\in\mathcal{K}} |k\rangle\langle k| \otimes \mathrm{Tr}\left[\mathsf{QAS.Ver}'_k(\rho)\right] \cdot \mathsf{QAS.Auth}_k(|\psi\rangle\langle\psi|) \\ \approx_{2\epsilon} &\mathop{\mathbb{E}}_{k\in K} |k\rangle\langle k| \otimes \mathrm{Tr}\left[\mathsf{QAS.Ver}'_k(\rho)\right] \cdot \mathsf{QAS.Auth}_k(|\phi\rangle\langle\phi|) \end{aligned} \tag{3.18}$$

after which we can apply Lemma 3.1 with the help of the key register, which yields

$$\mathop{\mathbb{E}}_{k\in K} \mathrm{Tr}\left[\mathsf{QAS.Ver}'_k(\rho)\right] \cdot \Delta\left(\mathsf{QAS.Auth}_k(|\psi\rangle\langle\psi|), \mathsf{QAS.Auth}_k(|\phi\rangle\langle\phi|)\right) \leq 2\epsilon. \tag{3.19}$$

It then suffices to note that $\Delta\left(\mathsf{QAS.Auth}_k(|\psi\rangle\langle\psi|), \mathsf{QAS.Auth}_k(|\phi\rangle\langle\phi|)\right) = 1$ by Lemma 3.6 since $|\psi\rangle$ is orthogonal to $|\phi\rangle$. The desired result follows. ∎

3.3 Definitions

Here, we define quantum copy-protection (Section 3.3.1) and secure software leasing (Section 3.3.2), along with their correctness and security notions. All of our definitions are for Boolean circuits only, where the input is a binary string, and the output is a single bit.

We briefly discuss the reusability of states generated by our schemes in Section 3.4.1. Finally, we define distributions on circuits and inputs which will often be used in this work in Section 3.3.3.

3.3.1 Quantum Copy-Protection

We present our definition of a copy-protection scheme, following the general lines of [CMP24]. We first define the functionality (Definition 3.7) and correctness (Definition 3.8) of a copy-protection scheme. We then define honest-malicious security in Definition 3.10, which can be contrasted with the usual definition of security (which we call malicious-malicious security) given in Definition 3.11. We note that we have rephrased the definition in [CMP24] in terms of the more standard cryptographic notion where the parameter in the definition (here, we use ϵ) characterizes the *insecurity* of a game (and hence, we strive for schemes where ϵ is small).

First, we define the functionality of *quantum copy-protection*.

Definition 3.7 (Quantum copy-protection scheme)**.** Let $\mathcal{C}$ be a set of n-bit Boolean circuits. A *quantum copy-protection* scheme for the set $\mathcal{C}$ is a pair of quantum circuits $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ such that for some space Y:

1. $\mathsf{CP.Protect}(C)$: takes as input a Boolean circuit $C \in \mathcal{C}$, and outputs a state $\rho \in \mathcal{D}(\mathsf{Y})$.
2. $\mathsf{CP.Eval}(\rho, x)$: takes a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$ and string $x \in \{0,1\}^n$ as inputs and outputs a bit b.

We will interpret the output of $\mathsf{CP.Protect}$ and $\mathsf{CP.Eval}$ as quantum states on Y and $\mathbb{C}^2$, respectively, so that, for example, for any bit b, string x and program ρ, $\mathrm{Tr}[|b\rangle\langle b| \, \mathsf{CP.Eval}(\rho, x)]$ is the probability that $\mathsf{CP.Eval}(\rho, x)$ outputs b.

Definition 3.8 (η-Correctness of copy-protection)**.** A *quantum copy-protection* scheme for a set of n-bit circuits $\mathcal{C}$, CP, is *η-correct* with respect to a family of distributions on n-bit strings $\{T_C\}_{C\in\mathcal{C}}$, if for any $C \in \mathcal{C}$ and $\rho = \mathsf{CP.Protect}(C)$, the scheme satisfies

$$\mathop{\mathbb{E}}_{x\leftarrow T_C} \mathrm{Tr}[|C(x)\rangle\langle C(x)| \, \mathsf{CP.Eval}(\rho, x)] \geq 1 - \eta. \tag{3.20}$$

Our notion of correctness differs from that of [CMP24] and other previous work by being defined with respect to a family of distributions (see Section 3.1.2). However, if the scheme is η-correct with respect to all families of distributions, then we recover the more standard definition of correctness.

We now define the notion of security for a copy-protection scheme against an adversary $\mathcal{A} = (\mathcal{P}, \mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{P}$ (Pete) is the *pirate*, and $\mathcal{A}_1$ (Bob) and $\mathcal{A}_2$ (Charlie) are *users* (see Figure 3.2). We use the $\mathsf{PiratingGame}$ from [CMP24] as the basis of our security game between a challenger and $\mathcal{A}$. The game is parametrized by: (i) a distribution D on the set of circuits $\mathcal{C}$, and (ii) a set of distributions $\{D_C\}_{C\in\mathcal{C}}$ over pairs of input strings in $\{0,1\}^n \times \{0,1\}^n$, called the *challenge distributions*.

The CP game $\mathsf{PiratingGame}_{\mathcal{A},\mathsf{CP}}$

1. The challenger samples $C \leftarrow D$ and sends $\rho = \mathsf{CP.Protect}(C)$ to $\mathcal{P}$.
2. $\mathcal{P}$ outputs a state σ on registers $\mathsf{A}_1, \mathsf{A}_2$ and sends A_1 to $\mathcal{A}_1$ and A_2 to $\mathcal{A}_2$.
3. At this point, $\mathcal{A}_1$ and $\mathcal{A}_2$ are separated and cannot communicate. The challenger samples $(x_1, x_2) \leftarrow D_C$ and sends x_1 to $\mathcal{A}_1$ and x_2 to $\mathcal{A}_2$.
4. $\mathcal{A}_1$ returns a bit b_1 to the challenger and $\mathcal{A}_2$ returns a bit b_2.
5. The challenger outputs 1 if and only if $b_1 = C(x_1)$ and $b_2 = C(x_2)$, in which case, we say that $\mathcal{A}$ wins the game.

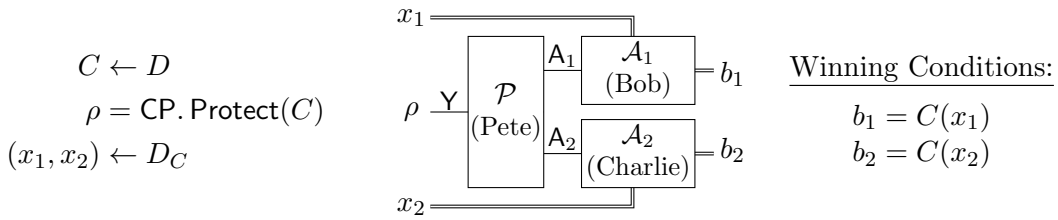


Figure 3.2: The pirating game $\mathsf{PiratingGame}_{\mathcal{A},\mathsf{CP}}$

In previous work on copy-protection, the adversary is assumed to control $\mathcal{P}$, $\mathcal{A}_1$ and $\mathcal{A}_2$, whose behaviour can be arbitrary (or, in some cases, computationally bounded). This models a setting where the potential users of pirated software are aware that the software is pirated, and willing to run their software in some non-standard way in order to make use of it. We refer to this setting as the *malicious-malicious* setting. In this setting, the action of the adversary $\mathcal{A} = (\mathcal{P}, \mathcal{A}_1, \mathcal{A}_2)$ can be specified by:

1. an arbitrary CPTP map $\Phi_{\mathcal{P}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{A}_1\mathsf{A}_2)$, representing the action of $\mathcal{P}$, where A_1 and A_2 are arbitrary spaces;
2. arbitrary two-outcome projective measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A_1, such that $\mathcal{A}_1$ (Bob) performs the measurement $\{\Pi_{x_1}, I - \Pi_{x_1}\}$ on input x_1 to obtain his output bit b_1; and
3. arbitrary two-outcome projective measurements $\{\Pi'_x\}_{x\in\{0,1\}^n}$ on A_2, such that $\mathcal{A}_2$ (Charlie) performs the measurement $\{\Pi'_{x_2}, I - \Pi'_{x_2}\}$ on input x_2 to obtain his output bit b_2.

Note that we restrict our attention to projective measurements for $\mathcal{A}_1$ and $\mathcal{A}_2$. Indeed, by a purification argument, any strategy using non-projective measurements is equivalent to a strategy with projective measurements and the extra auxiliary states needed by $\mathcal{A}_1$ and $\mathcal{A}_2$ can be provided by $\mathcal{P}$.

In contrast, one could also imagine a scenario in which users are honest, and will therefore try to evaluate the program they receive from $\mathcal{P}$ by running $\mathsf{CP.Eval}$. In that case, while $\mathcal{P}$ can still perform an arbitrary CPTP map, $\mathcal{A}_1$ and $\mathcal{A}_2$ are constrained to run $\mathsf{CP.Eval}$. It is potentially easier to design copy-protection schemes in this weaker setting, which we call the *honest-honest* setting, since the adversary is more constrained. We will consider an intermediate setting.

Diverging from previous work, we will focus on a special type of adversary, where $\mathcal{A}_1$ (Bob) performs the *honest* evaluation procedure, while $\mathcal{A}_2$ (Charlie) performs an arbitrary measurement. (See Section 3.1.1 for a discussion of this model). Specifically, we consider the following type of adversary.

Definition 3.9. An *honest-malicious adversary* for the pirating game is an adversary of the form $\hat{\mathcal{A}} = (\mathcal{P}, \mathsf{CP.Eval}, \mathcal{A}_2)$, where the party $\mathcal{P}$ implements an arbitrary CPTP map $\Phi_{\mathcal{P}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{YA}_2)$, where A_2 is any space, and $\mathcal{A}_2$ is specified by a set of arbitrary two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A_2.

For a fixed scheme $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ for a set of n-bit circuits $\mathcal{C}$, we define *honest-malicious* security with respect to distributions D and $\{D_C\}_{C\in\mathcal{C}}$ in terms of the best possible winning probability, $\Pr\left[\mathsf{PiratingGame}_{\hat{\mathcal{A}},\mathsf{CP}}\right]$, over honest-malicious adversaries $\hat{\mathcal{A}}$. Observe that there is one strategy that $\mathcal{P}$ can always facilitate, which is to pass the intact program to Bob and then let Charlie locally produce his best guess of the output, based on x_2, prior knowledge of D, and $\{D_C\}_{C\in\mathcal{C}}$[11]. This leads to a winning probability for the above game which is truly trivial to achieve, in the sense that Charlie is using a strategy that does not take any advantage of the interaction with the pirate $\mathcal{P}$ nor the program state ρ. We denote the winning probability for this strategy by $p^{\text{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}}$. In fact, assuming the scheme is η-correct with respect to the distribution family $\{T_C\}_{C\in\mathcal{C}}$ where T_C is Bob's marginal of D_C, Bob will always produce the correct answer, except with probability η. Indeed, Charlie simply considers the most likely output, given his input, thereby upper bounding the winning probability with Charlie's maximum guessing probability[12]

[11] There are other trivial strategies, *e.g.*, where Charlie gets an intact program register and Bob does not, but this is a more restricted trivial strategy, since Bob is constrained to evaluate the program honestly.

[12] The winning probability may be less than this. By the union bound, even if Bob's and Charlie's inputs are not independent, the overall success probability will be at least Charlie's maximum guessing probability, minus $p^{\text{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}} - \eta$. However, we expect η to be quite small.

Formally, we define $p^{\text{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}}$ as follows. The distributions D and $\{D_C\}_{C\in\mathcal{C}}$ yield a joint distribution $\tilde{D}$ on $\mathcal{C}\times\{0,1\}^n$ by first sampling the circuit $C\leftarrow D$ and then sampling a pair of inputs $(x_1,x_2)\leftarrow D_C$ and only taking the x_2 component. Let $\hat{D}$ be the marginal distribution of x_2 from $\tilde{D}$ and, for every x, let $\hat{D}_x$ be the marginal distribution of C from $\tilde{D}$, conditioned on $x_2=x$. Then,

$$p^{\text{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}} = \mathop{\mathbb{E}}_{x\leftarrow\hat{D}} \max_{b\in\{0,1\}} \Pr_{C\leftarrow\hat{D}_x}[C(x)=b]. \tag{3.21}$$

This is different from the security notion in [CMP24] where the trivial guessing probability is optimized over both users. For intuition, note that $p^{\text{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}}$ is always at least $1/2$, since Charlie can always output a random bit that is correct with probability $1/2$. Depending on the specific input and challenge distributions, it may be larger. We now state the main security notion for this work.

Definition 3.10 (Honest-malicious security)**.** A copy-protection scheme $\mathsf{CP}=(\mathsf{CP.Protect},\mathsf{CP.Eval})$ for a set of n-bit circuits $\mathcal{C}$ is *ϵ-honest-malicious secure with respect to the distribution D and challenge distributions $\{D_C\}_{C\in\mathcal{C}}$* if for all honest-malicious adversaries $\hat{\mathcal{A}}$,

$$\Pr\left[\mathsf{PiratingGame}_{\hat{\mathcal{A}},\mathsf{CP}}\right] \le p^{\text{marg}}+\epsilon\,, \tag{3.22}$$

where $p^{\text{marg}}=p^{\text{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}}$.

We reiterate that our definition for honest-malicious security is *statistical*: it makes no assumption on the computational power of $\hat{\mathcal{A}}$ (see Section 3.1.1).

Finally, if we modify the above definition by allowing arbitrary adversaries $\mathcal{A}=(\mathcal{P},\mathcal{A}_1,\mathcal{A}_2)$, and letting $\bar{p}^{\text{marg}}$ denote the optimal trivial guessing probability, as in Equation 3.21 but over *both* adversaries (see also [CMP24]), we recover the more standard security definition, which we call *malicious-malicious* security:

Definition 3.11 (Malicious-malicious security)**.** A copy-protection scheme CP for a set of n-bit circuits $\mathcal{C}$ is *ϵ-malicious-malicious secure with respect to the distribution D and challenge distributions $\{D_C\}_{C\in\mathcal{C}}$* if for *all* adversaries $\mathcal{A}$,

$$\Pr\left[\mathsf{PiratingGame}_{\mathcal{A},\mathsf{CP}}\right] \le \bar{p}^{\text{marg}}+\epsilon. \tag{3.23}$$

3.3.2 Secure Software Leasing

We define Secure Software Leasing (SSL) below. As with copy-protection, the basic scheme and security game mirror [CMP24] but we diverge from them in our exact notions of cor-

rectness and security. We first define the functionality (Definition 3.12) and correctness (Definition 3.13) of an SSL scheme, followed by its security (Definition 3.14).

Definition 3.12 (Secure software leasing (SSL))**.** Let $\mathcal{C}$ be a set of n-bit Boolean circuits. A *secure software leasing* scheme for $\mathcal{C}$ is a triplet of quantum circuits

$$\mathsf{SSL} = (\mathsf{SSL.Gen}, \mathsf{SSL.Lease}, \mathsf{SSL.Eval}, \mathsf{SSL.Verify}) \tag{3.24}$$

such that for some space Y:

1. $\mathsf{SSL.Gen}$: outputs a secret key sk.
2. $\mathsf{SSL.Lease}(\mathsf{sk}, C)$: takes as input a secret key sk and a circuit $C \in \mathcal{C}$, and outputs a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$.
3. $\mathsf{SSL.Eval}(\rho, x)$: takes as input a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$ and input string $x \in \{0,1\}^n$ and outputs a bit b.
4. $\mathsf{SSL.Verify}(\mathsf{sk}, \sigma, C)$: takes a secret key sk, a circuit $C \in \mathcal{C}$ and a quantum state $\sigma \in \mathcal{D}(\mathsf{Y})$, and outputs a bit v indicating acceptance or rejection.

Definition 3.13 (η-Correctness of SSL)**.** A *secure software leasing* scheme for $\mathcal{C}$, SSL, is *η-correct* with respect to a family of distributions on n-bit strings $\{T_C\}_{C\in\mathcal{C}}$, if for any $C \in \mathcal{C}$, $\mathsf{sk} \leftarrow \mathsf{SSL.Gen}$, and $\rho = \mathsf{SSL.Lease}(\mathsf{sk}, C)$, the scheme satisfies:

- Correctness of Evaluation: $\mathop{\mathbb{E}}_{x\leftarrow T_C} \mathrm{Tr}\left(|C(x)\rangle\langle C(x)|\, \mathsf{SSL.Eval}(\rho, x)\right) \geq 1 - \eta$,
- and Correctness of Verification: $\mathrm{Tr}\left(|1\rangle\langle 1|\, \mathsf{SSL.Verify}(\mathsf{sk}, \rho, C)\right) \geq 1 - \eta$.

In the above definition, recall that $\mathrm{Tr}\left(|C(x)\rangle\langle C(x)|\, \mathsf{SSL.Eval}(\rho, x)\right)$ is the probability that $\mathsf{SSL.Eval}(\rho, x)$ outputs $C(x)$, and that $\mathrm{Tr}\left(|1\rangle\langle 1|\, \mathsf{SSL.Verify}(\mathsf{sk}, \rho, C)\right)$ is the probability that $\mathsf{SSL.Verify}$ accepts.

When a scheme SSL is η-correct with respect to every distribution, we recover the more standard notion of correctness.

We base our security game, between a challenger (in this case a *Lessor*) and an adversary $\mathcal{A}$, on the $\mathsf{SSLGame}$ from [CMP24]. The game is parametrized by a distribution D over circuits in $\mathcal{C}$, and a set of challenge distributions $\{D'_C\}_{C\in\mathcal{C}}$ over inputs $\{0,1\}^n$.

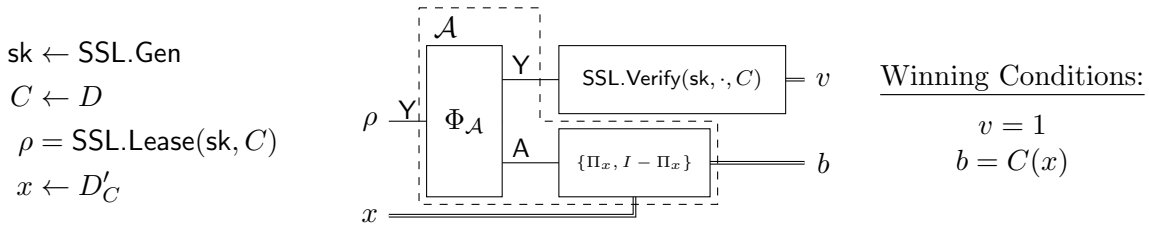


Figure 3.3: The SSL game $\mathsf{SSLGame}_{\mathcal{A},\mathsf{SSL}}$, where the behaviour of $\mathcal{A}$ is specified by a CPTP map $\Phi_{\mathcal{A}}$ and a set of two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$.

The SSL game $\mathsf{SSLGame}_{\mathcal{A},\mathsf{SSL}}$

1. The Lessor samples $C \leftarrow D$ and runs $\mathsf{SSL.Gen}$ to obtain a secret key sk. She then sends $\rho = \mathsf{SSL.Lease}(\mathsf{sk}, C)$ to $\mathcal{A}$.
2. $\mathcal{A}$ produces a state σ on registers YA and sends register Y back to the Lessor and keeps A.
3. (*Verification phase.*) The Lessor runs $\mathsf{SSL.Verify}$ on Y, the circuit C and the secret key sk and outputs the resulting bit v. If $\mathsf{SSL.Verify}$ accepts ($v = 1$), the game continues, otherwise it aborts and $\mathcal{A}$ loses.
4. The Lessor samples an input $x \leftarrow D'_C$ and sends x to $\mathcal{A}$.
5. $\mathcal{A}$ returns a bit b to the Lessor.
6. The Lessor outputs 1 if and only if $b = C(x)$ and $v = 1$, in which case, we say $\mathcal{A}$ "wins" the game.

An adversary $\mathcal{A}$ for $\mathsf{SSLGame}$ can be described by a CPTP map $\Phi_{\mathcal{A}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{YA})$ for some arbitrary space A, representing the action of $\mathcal{A}$ in Step 2, and a set of two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A such that given challenge x in Step 4, $\mathcal{A}$ obtains the bit b in Step 5 by measuring A with $\{\Pi_x, I - \Pi_x\}$ (see Figure 3.3).

As in Section 3.3.1, we define security with respect to the trivial strategy where $\mathcal{A}$ returns the program ρ to the Lessor in Step 2, and tries to guess the most likely value for b, given input x.

Formally, we define $p^{\text{triv}}_{D,\{D_C\}_{C\in\mathcal{C}}}$ as follows. The distributions D and $\{D_C\}_{C\in\mathcal{C}}$ yield a joint distribution $\tilde{D}$ on $\mathcal{C} \times \{0,1\}^n$ by first sampling $C \leftarrow D$ and then sampling $x \leftarrow D_C$. Let $\hat{D}$ be the marginal distribution of x from $\tilde{D}$ and, for every x', let $\hat{D}_{x'}$ be the marginal

distribution of C from $\tilde{D}$, conditioned on $x = x'$. Then,

$$p^{\text{triv}}_{D,\{D'_C\}_{C\in\mathcal{C}}} = \mathop{\mathbb{E}}_{x\leftarrow\hat{D}} \max_{b\in\{0,1\}} \Pr_{C\leftarrow\hat{D}_x}[C(x) = b]. \tag{3.25}$$

The above equation is very similar to p^{marg} given in Equation 3.21. However, we point out that they are defined and used in different contexts. Specifically, in PiratingGame there are two parties, Bob and Charlie, who must be challenged with inputs on which to evaluate the function. However, there is only a single party attempting to evaluate the function at the end of SSLGame. Thus, p^{marg} is defined with respect to the marginal distribution on Charlie's challenge generated by the joint challenge distribution. On the other hand, p^{triv} can be directly defined with respect to the single challenge issued in SSLGame.

We now define the security of SSL as follows.

Definition 3.14 (Security of SSL)**.** An SSL scheme SSL for a set of n-bit circuits $\mathcal{C}$ is *ϵ-secure with respect to the distribution D and challenge distributions $\{D'_C\}_{C\in\mathcal{C}}$* if for all adversaries $\mathcal{A}$,

$$\Pr[\mathsf{SSLGame}_{\mathcal{A}}] \leq p^{\text{triv}} + \epsilon\,, \tag{3.26}$$

where $p^{\text{triv}} = p^{\text{triv}}_{D,\{D'_C\}_{C\in\mathcal{C}}}$.

Observe that, as in the case with Definition 3.10, our definition provides statistical guarantees for security as we impose no conditions on the adversaries.

3.3.3 Distributions for Point Functions

The definitions of correctness and security for copy-protection and secure software leasing presented earlier in this section are parametrized by various distributions on the circuits that are encoded and the challenges that are issued.

In this section, we define notation for the distributions we will consider in the setting of point functions. First, we will consider security in the setting when the point function is chosen uniformly at random.

Definition 3.15. We denote by R be the uniform distribution on the set of point functions $\{P_p \;:\; p \in \{0,1\}^n\}$. For simplicity, we will also use R to refer to the uniform distribution on $\{0,1\}^n$, as we often conflate a point p with its corresponding point function P_p.

For a fixed point p, we will consider the distribution of inputs where p is sampled with probability $1/2$, and otherwise, a uniform $x \neq p$ is sampled.

Definition 3.16. For any bit string $p \in \{0,1\}^n$, we define $T_p^{(1/2)}$ to be the distribution on $\{0,1\}^n$ such that

- p is sampled with probability $\frac{1}{2}$ and
- any $x \neq p$ is sampled with probability $\frac{1}{2} \cdot \frac{1}{2^n-1}$.

This is a natural distribution in the setting of point functions, since it means that the function evaluates to a uniform random bit. This ensures that the output is non-trivial to guess — an adversary's advantage against challenge distributions of this form can be quantified by comparing it with their probability of correctly guessing a random bit. Furthermore, η-correctness with respect to this distribution, for some small η, ensures that evaluating the point is correct except with small probability, and that all but a small fraction of the other inputs are evaluated correctly except with small probability.

3.4 Generic Results on Definitions

Here, we give some generic results concerning definitions given in Section 3.3.

We first discuss the reusability of program states generated by copy-protection or SSL schemes in Section 3.4.1. In Section 3.4.2, we outline how a copy-protection scheme satisfying malicious-malicious security and average correctness can be used to obtain a scheme satisfying malicious-malicious security and the more standard definition of correctness. Next, in Section 3.4.3, we describe how an honest-malicious copy-protection scheme for any set of circuits $\mathcal{C}$ can be turned into an SSL scheme for $\mathcal{C}$ (Theorem 3.18). In particular, this means that the copy-protection scheme for point functions presented in Section 3.5 implies an SSL scheme for point functions. Finally, in Section 3.4.4, we refine a result from [CMP24] (tailoring it to our definitions) showing that an SSL scheme for point functions can be used to construct an SSL scheme for compute-and-compare programs (Theorem 3.21).

3.4.1 Reusability of the Program

For ease of notation, we define both $\mathsf{CP}.\mathsf{Eval}(\rho, x)$ and $\mathsf{SSL}.\mathsf{Eval}(\rho, x)$ to take a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$ and a string $x \in \{0,1\}^n$ as inputs and output a bit b. This can be extended [AL21] to a *reusable* scheme in a straightforward way, so that the evaluation procedure outputs a bit b together with a post-evaluation state $\tilde{\rho}$, which approximates ρ. In more details, we purify the evaluation procedure, and copy the output bit, before then undoing the evaluation procedure.[13]

We claim that the above procedure, used to sequentially evaluate r inputs sampled from the same distribution with respect to which the scheme is η-correct, will produce all r

[13] Alternatively, this could be seen through the lens of the coherent gentle measurement discussed in Section 2.2.3 of the previous chapter.

correct answers with a probability of at least $(1 - r\sqrt{\eta})(1 - 4r\sqrt{\eta})$. We highlight that this bound is sufficient to show that, in an asymptotic regime, a program state with negligible errors evaluated polynomially many times on randomly sampled inputs will give all the correct values with overwhelming probability.

We give the above probabilistic statement because it is impossible to give a precise figure for the number of times a program can be evaluated before it stops working altogether. Indeed, in some cases the evaluation of the program state could leave it unchanged, in which case it can be reused without limit. This will be the case for the authentication-based scheme we construct in Section 3.5, where a purified evaluation of the program for the point function for the point p on the point p will not cause any change as the evaluation produces the correct outcome with certainty. So, in this case, it is possible to correctly evaluate on p an arbitrary number of times. It follows that to give a meaningful answer on how many times a program state can be used, we must specify how the inputs to the program are selected. We believe it is reasonable to sample them according to the same distribution for which correctness is guaranteed.

We sketch a proof of our claim. It follows from a concentration inequality, the classical union bound, and Gao's quantum union bound [Gao15].[14]

First: Applying the idea of Markov's inequality to a setting where the random variable is distributed on the interval $[0, 1]$ yields a concentration inequality showing that if the expectation, over the choice of inputs, that the program produces the correct output is $1 - \eta$, then, with probability at least $1 - \sqrt{\eta}$, a randomly chosen input will be evaluated to the correct output with probability at least $1 - \sqrt{\eta}$. Second: By the classical union bound, the probability that r sampled inputs are correctly evaluated with probability at least $1 - \sqrt{\eta}$ is at least $1 - r\sqrt{\eta}$. Third: We can model the evaluation of the program state as projective measurements where the input to the program determines the measurement. Given r measurements, each producing the correct outcome with probability $1 - \sqrt{\eta}$ on the original state, Gao's quantum union bound [Gao15] yields that the sequential application of all of these measurements will all give the correct answers with probability at least $1 - 4r\sqrt{\eta}$. Multiplying this with the probability that all chosen inputs satisfy the necessary condition yields our bound of $(1 - r\sqrt{\eta})(1 - 4r\sqrt{\eta})$.

3.4.2 Malicious-Malicious Security and Correctness

We show in this work that any copy-protection scheme for point functions that is secure in the malicious-malicious setting but only satisfies correctness with respect to the distribution family $\{T_p^{(1/2)}\}_p$, in which $T_p^{(1/2)}$ samples p with probability $1/2$ and all other strings

[14] See also [OV22] for an alternative presentation of this result.

uniformly, can be combined with a pairwise independent permutation family [NR99] to obtain a scheme that is still secure in the malicious-malicious setting but is also correct with respect to any distribution. The general idea is to use a random element from a pairwise independent permutation family to remap inputs to the point function uniformly across the domain. This will ensure that, on average, no specific element will lead to a wrong evaluation with large probability. This implies correctness with respect to any distribution.

We state the implications of this construction in the next theorem, but leave the details of it and related proofs to the supplementary materials of this chapter (Section 3.6.2) as they are somewhat involved and not strictly necessary for our subsequent results.

Theorem 3.17. If there exists a copy-protection scheme for point functions which is ϵ-malicious-malicious secure with respect to the uniform distribution on points R and challenge distribution $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$ and η-correct with respect to the distribution family $\{T_p^{(1/2)}\}_p$, then there exists a copy-protection scheme for point functions which is ϵ-malicious-malicious secure with respect to the distributions R and $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$ and 2η-correct with respect to any distribution.

Recall that the malicious-malicious security setting is the standard security definition considered in previous works, and correctness with respect to any distribution is the standard notion of correctness. Thus, our construction given in Section 3.5, while it has its advantages, falls short of achieving the standard security and correctness notions by being secure only in the honest-malicious setting, and by being correct only with respect to $\{T_p^{(1/2)}\}_p$. Our result here shows that solving the former problem would also solve the latter.

3.4.3 Secure Software Leasing and Honest-Malicious Copy-Protection

Following Figure 3.1, we now outline how an honest-malicious copy-protection scheme for some set of circuits $\mathcal{C}$ can be used to create an SSL scheme for $\mathcal{C}$ (Theorem 3.18). Specifically, we use a copy-protection scheme for a set of Boolean circuits $\mathcal{C}$ on n-bits that is correct with respect to *two* families of distributions, $\{T_C\}_{C\in\mathcal{C}}$ and $\{T'_C\}_{C\in\mathcal{C}}$, and honest-malicious secure with respect to the circuit distribution D on $\mathcal{C}$, and the challenge distributions $\{T'_C \times T''_C\}_{C\in\mathcal{C}}$, to construct an SSL scheme for $\mathcal{C}$ that is correct with respect to $\{T_C\}_{C\in\mathcal{C}}$ and secure with respect to D and $\{T''_C\}_{C\in\mathcal{C}}$. Here, $T'_C \times T''_C$ denotes the product distribution of the two distributions T'_C and T''_C, which are both distributions on $\{0,1\}^n$.

Let $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ be a copy-protection scheme for a set of n-bit Boolean circuits $\mathcal{C}$. We define the secure software leasing scheme SSL for $\mathcal{C}$ as:

$\mathsf{SSL.Gen}$: Output an empty secret key $\mathsf{sk} = \emptyset$.

$\mathsf{SSL.Lease}(C)$: As the secret key is empty, the only input is the circuit C. On input C, output $\rho = \mathsf{CP.Protect}(C)$.

$\mathsf{SSL.Eval}(\rho, x)$: On input $\rho \in \mathcal{D}(\mathsf{Y})$ and $x \in \{0,1\}^n$, run $\mathsf{CP.Eval}$.

$\mathsf{SSL.Verify}(C, \sigma)$: As the secret key is empty, the only inputs are the circuit C and a state $\sigma \in \mathsf{Y}$. Sample $x \leftarrow T'_C$ and output 1 if and only if $\mathsf{CP.Eval}(\sigma, x)$ is $C(x)$.

Formally, we obtain the following theorem.

Theorem 3.18. Suppose the scheme CP is a copy-protection scheme for circuits $\mathcal{C}$, that is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$, η-correct with respect to $\{T'_C\}_{C\in\mathcal{C}}$, and ϵ-honest-malicious secure with respect to the distribution D on $\mathcal{C}$ and challenge distributions $\{T'_C \times T''_C\}_{C\in\mathcal{C}}$. Then SSL, constructed from CP as described above, is an SSL scheme for $\mathcal{C}$ that is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$ and ϵ-secure with respect to the distributions D and $\{T''_C\}_{C\in\mathcal{C}}$.

We prove this theorem via two lemmas: Lemma 3.19 for the correctness, and Lemma 3.20 for the security. We briefly sketch our arguments before giving them in full formality.

Correctness of SSL follows from the correctness of CP directly as the encoding and evaluating procedures for the programs are the same. The main intuition for the security proof is to map the honest evaluation in the scheme CP to the Lessor's verification procedure in the scheme SSL. The η-correctness of $\mathsf{CP.Eval}$ ensures that the verification is accepted with sufficiently high probability. Next, we map the malicious user Charlie's ($\mathcal{A}_2$) evaluation in $\mathsf{PiratingGame}$ to the adversary's evaluation in $\mathsf{SSLGame}$. Assuming that CP is secure, we can bound Charlie's probability of guessing the right answer, which in turn bounds the adversary's probability of guessing the right answer. Putting it together, we can conclude that the corresponding SSL scheme SSL is secure.

We remark that our proofs do not make any assumptions about the abilities of the adversaries. Hence, if the copy-protection scheme CP achieves statistical security guarantees, then so does the corresponding SSL scheme SSL.

Lemma 3.19. If the scheme CP is η-correct with respect to a family of distributions $\{T_C\}_{C\in\mathcal{C}}$ and also with respect to the family $\{T'_C\}_{C\in\mathcal{C}}$ used in the definition of SSL, then the scheme SSL described above is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$.

Proof: The proof proceeds directly by considering the definition of correctness for CP and SSL from Definitions 3.8 and 3.13. For any $C \in \mathcal{C}$, if $\rho = \mathsf{SSL.Lease}(C) = \mathsf{CP.Protect}(C)$, then

$$\mathop{\mathbb{E}}_{x\leftarrow T_C} \mathrm{Tr}\left(|C(x)\rangle\langle C(x)|\,\mathsf{SSL.Eval}(\rho, x)\right) = \mathop{\mathbb{E}}_{x\leftarrow T_C} \mathrm{Tr}\left(|C(x)\rangle\langle C(x)|\,\mathsf{CP.Eval}(\rho, x)\right) \geq 1 - \eta, \tag{3.27}$$

where the last inequality uses that CP is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$.

To satisfy the correctness requirement for SSL.Verify, note that by the construction of SSL.Verify,

$$\mathrm{Tr}\left(|1\rangle\langle 1|\,\mathsf{SSL.Verify}(C,\rho)\right) = \mathop{\mathbb{E}}_{x\leftarrow T'_C} \mathrm{Tr}\left(|C(x)\rangle\langle C(x)|\,\mathsf{CP.Eval}(\rho,x)\right) \geq 1-\eta, \tag{3.28}$$

where the last inequality uses that CP is also η-correct with respect to $\{T'_C\}_{C\in\mathcal{C}}$. Putting together Equations 3.27 and 3.28, we can conclude that SSL is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$. ∎

Lemma 3.20. If the scheme CP is ϵ-honest-malicious secure with respect to a circuit distribution D, and challenge distributions $\{T'_C \times T''_C\}_{C\in\mathcal{C}}$, then the scheme SSL described above is ϵ-secure with respect to D and $\{T''_C\}_{C\in\mathcal{C}}$.

Proof: Let $p^{\mathrm{triv}} = p^{\mathrm{triv}}_{D,\{T''_C\}_{C\in\mathcal{C}}}$ and $p^{\mathrm{marg}} = p^{\mathrm{marg}}_{D,\{T'_C\times T''_C\}_{C\in\mathcal{C}}}$ and assume for the rest of the proof that SSLGame is instantiated with circuit distribution D and challenge ensemble $\{T''_C\}_{C\in\mathcal{C}}$, and PiratingGame is instantiated with circuit distribution D and challenge ensemble $\{T'_C \times T''_C\}$.

The proof proceeds by contradiction. We use a winning adversary $\mathcal{A}_{\mathsf{SSL}}$ against the scheme SSL to construct a winning $\hat{\mathcal{A}}_{\mathsf{CP}}$ honest-malicious adversary against the scheme CP. As CP is assumed to be secure, this implies the security of SSL. More precisely, let $\mathcal{A}_{\mathsf{SSL}}$ be an adversary for SSLGame such that

$$\Pr\left[\,\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{SSL}},\mathsf{SSL}}\,\right] > p^{\mathrm{triv}} + \epsilon. \tag{3.29}$$

We construct an adversary $\hat{\mathcal{A}}_{\mathsf{CP}}$ that wins $\mathsf{PiratingGame}_{\hat{\mathcal{A}}_{\mathsf{CP}},\mathsf{CP}}$ with probability strictly more than $p^{\mathrm{marg}} + \epsilon$.

The behaviour of the adversary $\mathcal{A}_{\mathsf{SSL}}$ can be described in two parts (see also Figure 3.3). First, the adversary applies an arbitrary CPTP map $\Phi_{\mathcal{A}_{\mathsf{SSL}}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{YA})$ to his input $\rho = \mathsf{SSL.Lease}(\mathsf{sk}, C)$, sending the Y part to the Lessor (step 2 of SSLGame); and keeping the A part, for some arbitrary space A, for himself. Later, when $\mathcal{A}_{\mathsf{SSL}}$ receives the challenge x, he uses it to select a two outcome measurement $\{\Pi_x, I - \Pi_x\}$ with which to measure his register A to obtain a bit b (step 5 of SSLGame).

Consider the honest-malicious adversary $\hat{\mathcal{A}}_{\mathsf{CP}} = (\mathcal{P}, \mathsf{CP.Eval}, \mathcal{A}_2)$ such that:

- $\mathcal{P}$'s behaviour is described by the map $\Phi_{\mathcal{P}} = \Phi_{\mathcal{A}_{\mathsf{SSL}}}$, and
- $\mathcal{A}_2$'s behaviour is described by the two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$.

Then $\mathsf{PiratingGame}_{\hat{\mathcal{A}}_{\mathsf{CP}},\mathsf{CP}}$ proceeds as follows.

1. The challenger samples $C \leftarrow D$ and sends $\rho = \mathsf{CP.Protect}(C)$ to $\mathcal{P}$.
2. Upon receiving ρ, $\mathcal{P}$ computes $\sigma = \Phi_{\mathcal{A}_{\mathsf{SSL}}}(\rho) \in \mathcal{D}(\mathsf{YA})$ and sends Y to Bob, who is controlled by the challenger in the honest-malicious setting, and A to Charlie.
3. The challenger samples $x_1 \leftarrow T'_C$ and sends it to Bob, and samples $x_2 \leftarrow T''_C$ and sends it to Charlie.
4. Bob runs $\mathsf{CP.Eval}$ on Y and x_1 and outputs the resulting bit b_1.
5. Charlie measures A using the measurement $\{\Pi_{x_2}, I - \Pi_{x_2}\}$ and outputs the resulting bit b_2.
6. The challenger outputs 1 if and only if $b_1 = C(x_1)$ and $b_2 = C(x_2)$.

To see why this construction works, observe that the pirate $\mathcal{P}$ is essentially acting as $\mathcal{A}_{\mathsf{SSL}}$ in step 2 of $\mathsf{SSLGame}$, the only difference is that after applying $\Phi_{\mathcal{A}_{\mathsf{SSL}}}$, $\mathcal{A}_{\mathsf{SSL}}$ keeps the register A for himself, whereas $\mathcal{P}$ sends it to Charlie. Later Charlie behaves just as $\mathcal{A}_{\mathsf{SSL}}$ behaves in Step 5 of $\mathsf{SSLGame}$. If we define $\Pi_x^1 = \Pi_x$ and $\Pi_x^0 = I - \Pi_x$, then $\Pi_{x_2}^{C(x_2)}$ is the projector onto the part of Charlie's input state that will lead Charlie to output the correct bit, $b_2 = C(x_2)$ in $\mathsf{PiratingGame}$. It's also the projector onto the part of $\mathcal{A}_{\mathsf{SSL}}$'s memory A that leads to him outputting the correct bit $b = C(x)$ in $\mathsf{SSLGame}$. Then, letting Ψ_{Ver}^C denote the map induced by $\mathsf{SSL.Verify}(C, \cdot)$, we have:

$$\begin{aligned}
&\Pr[\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{SSL}},\mathsf{SSL}}] \\
&= \mathop{\mathbb{E}}_{C \leftarrow D} \mathop{\mathbb{E}}_{x_2 \leftarrow T''_C} \mathrm{Tr}\left((|1\rangle\langle 1| \otimes \Pi_{x_2}^{C(x_2)})(\Psi_{\mathsf{Ver}}^C \otimes \mathrm{Id}) \circ \Phi_{\mathcal{A}_{\mathsf{SSL}}}(\mathsf{SSL.Lease}(C)) \right) \\
&= \mathop{\mathbb{E}}_{C \leftarrow D} \mathop{\mathbb{E}}_{x_2 \leftarrow T''_C} \mathrm{Tr}\left((|1\rangle\langle 1| \otimes \Pi_{x_2}^{C(x_2)})(\Psi_{\mathsf{Ver}}^C \otimes \mathrm{Id}) \circ \Phi_{\mathcal{P}}(\mathsf{CP.Protect}(C)) \right).
\end{aligned} \tag{3.30}$$

Let Ψ_{Eval}^x denote the map induced by $\mathsf{CP.Eval}(\cdot, x)$. Then Ψ_{Ver}^C samples $x_1 \leftarrow T'_C$, applies $\Psi_{\mathsf{Eval}}^{x_1}$, and outputs 1 if and only if the result is $C(x)$. Thus, we can continue from above:

$$\begin{aligned}
&\Pr[\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{SSL}},\mathsf{SSL}}] \\
&= \mathop{\mathbb{E}}_{C \leftarrow D} \mathop{\mathbb{E}}_{\substack{x_1 \leftarrow T'_C \\ x_2 \leftarrow T''_C}} \mathrm{Tr}\left((|C(x_1)\rangle\langle C(x_1)| \otimes \Pi_{x_2}^{C(x_2)})(\Psi_{\mathsf{Eval}}^{x_1} \otimes \mathrm{Id}) \circ \Phi_{\mathcal{P}}(\mathsf{CP.Protect}(C)) \right) \\
&= \Pr\left[\mathsf{PiratingGame}_{\hat{\mathcal{A}}_{\mathsf{CP}},\mathsf{CP}}\right].
\end{aligned} \tag{3.31}$$

By assumption, $\Pr[\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{SSL}},\mathsf{SSL}}] > p^{\mathrm{triv}} + \epsilon$, so $\Pr\left[\mathsf{PiratingGame}_{\hat{\mathcal{A}}_{\mathsf{CP}},\mathsf{CP}}\right] > p^{\mathrm{triv}} + \epsilon$.

Additionally, for any C, x_2 is distributed according to T''_C, independent of x_1. This is also the challenge distribution in $\mathsf{SSLGame}$. Therefore, using Equations 3.21 and 3.25, $p^{\text{triv}} = p^{\text{marg}}$ which implies that,

$$\Pr\left[\mathsf{PiratingGame}_{\hat{\mathcal{A}}_{\mathsf{CP}},\mathsf{CP}}\right] > p^{\text{marg}} + \epsilon. \tag{3.32}$$

This contradicts the assumption that CP is ϵ-honest-malicious secure and completes the proof. ∎

3.4.4 Secure Software Leasing of Compute-and-Compare Circuits

In this section we present a restatement of a theorem due to [CMP24], which states that an SSL scheme for point functions that is ϵ-secure with respect to a family of distributions can be modified to get an SSL scheme for compute-and-compare programs that is also ϵ-secure with respect to a related family of distributions. We state this result with a more precise relationship between the distributions used for the point functions and the compute-and-compare programs. This yields the final implication from Figure 3.1.

Let F be a set of functions from $\{0,1\}^n$ to $\{0,1\}^m$. Let $\mathcal{F} = \{(f,y) : f \in F, y \in \{0,1\}^m\}$ be the set of compute-and-compare circuits for F, where as with point functions, we identify (f,y) with a circuit CC^f_y for the function that outputs 1 on input x if and only if $f(x) = y$.

Recall the construction of [CMP24] to transform an SSL scheme for point functions into an SSL scheme for compute-and-compare functions. The main idea here is to encode the point function for y while providing f directly and "in the clear", for example by simply describing a circuit computing f.

Let $\mathsf{PF} = (\mathsf{PF.Gen}, \mathsf{PF.Lease}, \mathsf{PF.Eval}, \mathsf{PF.Verify})$ be an SSL scheme for m-bit point functions, which is to say point functions with $\{0,1\}^m$ as domain. We define an SSL scheme $\mathsf{CC} = (\mathsf{CC.Gen}, \mathsf{CC.Lease}, \mathsf{CC.Eval}, \mathsf{CC.Verify})$ for the compute-and-compare functions $\mathcal{F}$ as follows:

$\mathsf{CC.Gen}$: Compute $\mathsf{PF.Gen}$ and output the resulting secret key sk.

$\mathsf{CC.Lease}(\mathsf{sk}, (f,y))$: On input of the secret key sk and $(f,y) \in \mathcal{F}$, output (f,ρ) where $\rho = \mathsf{PF.Lease}(\mathsf{sk}, P_y)$.

$\mathsf{CC.Eval}((f,\rho),x)$: On input $\rho \in \mathcal{D}(\mathsf{Y})$, $f \in F$, and $x \in \{0,1\}^n$ do the following:

1. Compute $y' = f(x)$.
2. Compute $\mathsf{PF.Eval}(\rho, y')$ to get a bit b and output this bit.

$\mathsf{CC.Verify}(\mathsf{sk}, (f, y), \sigma)$: On input of a secret key sk, $(f, y) \in \mathcal{F}$ and $\sigma \in \mathcal{D}(\mathsf{Y})$, compute $\mathsf{PF.Verify}(\mathsf{sk}, P_y, \sigma)$ and output the resulting bit v.

Formally, we show the following theorem.

Theorem 3.21. We fix the following distributions.

- D: A distribution over compute-and-compare functions CC^f_y, or equivalently, over $(f, y) \in \mathcal{F}$. Fixing a function $f \in F$ induces a marginal distribution D_f over $y \in \{0,1\}^m$, or equivalently, over m-bit point functions P_y.
- $\{T^{CC}_{f,y}\}_{f,y}$ and $\{D^{CC}_{f,y}\}_{f,y}$: Families of distributions over inputs $x \in \{0,1\}^n$ to compute-and-compare functions CC^f_y.
- $\{T^{PF}_{f,y}\}_{f,y}$ and $\{D^{PF}_{f,y}\}_{f,y}$: Families of distributions over inputs $z \in \{0,1\}^m$ to m-bit point functions P_y, where $T^{PF}_{f,y}$ is defined from $T^{CC}_{f,y}$ by sampling $x \leftarrow T^{CC}_{f,y}$ and outputting $f(x)$; and $D^{PF}_{f,y}$ is defined similarly from $D^{CC}_{f,y}$.

Suppose that PF is a secure software leasing scheme for point functions such that, for every $f^* \in F$, PF is η-correct with respect to the distribution family $\{T^{PF}_{f^*,y}\}_{y\in\{0,1\}^m}$ and ϵ_{f^*}-secure with respect to the circuit distribution D_{f^*} and challenge distributions $\{D^{PF}_{f^*,y}\}_{y\in\{0,1\}^m}$ where

$$\epsilon_{f^*} = \left(p^{\mathrm{triv}}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} - p^{\mathrm{triv}}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y}\right) + \epsilon. \tag{3.33}$$

Then the scheme CC, constructed from PF as described in this section, is an SSL scheme for compute-and-compare programs in $\mathcal{F}$ that is η-correct with respect to the family $\{T^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$ and ϵ-secure with respect to program distribution D and challenge distributions $\{D^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$.

The proof of correctness follows directly from definitions and the proof of security follows the same lines as the one presented in [CMP24]. For completeness, these are provided in the supplementary materials for this chapter in Section 3.6.3.

3.5 An Authentication-Based Copy-Protection Scheme

In this section, we show how to construct a copy-protection scheme for point functions, with honest-malicious security, from a total authentication scheme.

Recall that we assume that our circuits are searchable, which, for point functions, implies that there is an efficient algorithm which can produce the point p from a circuit which computes its point function. Thus, we will freely identify circuits for the point function P_p simply with p. Specifically, our copy-protection scheme will take as input a point p instead of a circuit.

3.5.1 Construction and Correctness

Let $\mathsf{QAS} = (\mathsf{QAS.Auth}, \mathsf{QAS.Ver})$ be an ϵ-total quantum authentication scheme, as in Definition 3.2, with $\epsilon \leq \frac{1}{2}$ for a message space M of dimension greater than or equal to two with key set $\mathcal{K} = \{0,1\}^n$. Fix some state $|\psi\rangle \in \mathsf{M}$.

We recall that we assume that for every key k, the action of $\mathsf{QAS.Auth}$ with this key can be modeled by an isometry $A_k : \mathsf{M} \to \mathsf{Y}$. Recall that since A_k is an isometry, $A_k A_k^\dagger$ is the projector onto $\mathrm{im}(A_k)$. Further, let $V_k : \mathsf{Y} \to \mathsf{MFX}$ be an isometry which purifies the CPTP map $\mathsf{QAS.Ver}_k$ defined in Equation 3.6, where the register X corresponds to the Hilbert space used for this purification. To simplify our notation, we will absorb X into the flag register, which we no longer assume to be two-dimensional. We can still assume that there is a unique accepting state $|\mathrm{Acc}\rangle \in \mathsf{F}$.[15] Thus, from here on, we assume that $V_k : \mathsf{Y} \to \mathsf{MF}$ is an isometry, and F has dimension at least two (but possibly larger) with $|\mathrm{Acc}\rangle$ the accepting state, and all orthogonal states rejecting.

Finally, we will write $\overline{V}_k = (\langle \mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}})V_k$ to denote the map which applies the verification, but only outputs the state corresponding to the verification procedure accepting, corresponding to the procedure $\mathsf{QAS.Ver}'_k$ described in Section 3.2.2. Then note that $\overline{V}_k = A_k^\dagger$.

From this authentication scheme and fixed state $|\psi\rangle$, which can be assumed without loss of generality to be $|0\rangle$, we construct a copy-protection scheme for point functions of length n, AuthCP, as follows:

$\mathsf{AuthCP.Protect}(p)$: On input $p \in \{0,1\}^n$, do the following:

1. Output $A_p |\psi\rangle$.

$\mathsf{AuthCP.Eval}(\sigma, x)$: On input $\sigma \in \mathcal{D}(\mathsf{Y})$ and $x \in \{0,1\}^n$, do the following:

1. Compute $\xi = V_x \sigma V_x^\dagger$. Recall that ξ is a state on registers F, the flag register, and M, the message register.
2. Measure the F register of ξ in $\{|\mathrm{Acc}\rangle\langle\mathrm{Acc}|, I - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|\}$. If the outcome obtained is "Acc", output 1. Otherwise, output 0.

We recall that correctness is parametrized by a family of input distributions to each point function, and security is parametrized by a distribution on the possible functions to be encoded and by a family of distributions on challenges to send the users Bob and Charlie. Our correctness and security are proven with respect to the following distributions:

[15] This follows from correctness, since for every state $|\psi\rangle$, we necessarily have $V_k A_k |\psi\rangle = |\psi\rangle_{\mathsf{M}} |\mathrm{Acc}\rangle_{\mathsf{F}} |X_\psi\rangle_{\mathsf{X}}$ for some state $|X_\psi\rangle$, and, by the fact that $V_k A_k$ must preserve inner products, we necessarily have $|X_\psi\rangle = |X\rangle$ independent of $|\psi\rangle$. Thus, we can let $|\mathrm{Acc}\rangle_{\mathsf{F}} |X\rangle_{\mathsf{X}}$ be the accepting state on FX.

- Our correctness will be with respect to the distribution $T_p^{(1/2)}$, as defined in Definition 3.16, which we recall is the distribution on $\{0,1\}^n$ in which p is sampled with probability $1/2$, and all other strings are sampled with probability $\frac{1}{2(2^n-1)}$.
- In our security proof, we will assume that the point p of the challenge function is chosen uniformly at random. This corresponds to the distribution R given in Definition 3.15.
- If the challenge function is specified by the point p, the challenges will be sampled independently and according to the distribution $T_p^{(1/2)}$. We will denote this distribution by $T_p^{(1/2)} \times T_p^{(1/2)}$.

We first prove the correctness of the scheme AuthCP.

Theorem 3.22. If the scheme QAS is an ϵ-total authentication scheme, then the scheme AuthCP described above is ϵ-correct with respect to the family of distributions $\{T_p^{(1/2)}\}_p$.

Proof: For all $p \in \{0,1\}^n$, it suffices to compute a lower bound on

$$\frac{1}{2}\left\|\overline{V}_p A_p \left|\psi\right\rangle\right\|^2 + \frac{1}{2}\cdot\frac{1}{2^n-1}\sum_{\substack{x\in\{0,1\}^n \\ x\neq p}}\left(1-\left\|\overline{V}_x A_p \left|\psi\right\rangle\right\|^2\right). \tag{3.34}$$

By the correctness of the authentication scheme, we have that $\left\|\overline{V}_p A_p \left|\psi\right\rangle\right\|^2 = 1$. On the other hand, by Lemma 3.5, we have that

$$\sum_{\substack{x\in\{0,1\}^n \\ x\neq p}}\left\|\overline{V}_x A_p \left|\psi\right\rangle\right\|^2 \leq 2^n\cdot 2\epsilon - 1 \tag{3.35}$$

by expanding the expectation and removing the term corresponding to $x = p$. Thus, a lower bound for Equation 3.34 is given by

$$\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{2^n-1}\left(2^n-1-\sum_{\substack{x\in\{0,1\}^n \\ x\neq p}}\left\|\overline{V}_x A_p \left|\psi\right\rangle\right\|^2\right) \geq \frac{1}{2}+\frac{1}{2}\left(1-\frac{2^n\cdot 2\epsilon-1}{2^n-1}\right) \geq 1-\epsilon, \tag{3.36}$$

as long as $\epsilon \leq 1/2$, and so the scheme is ϵ-correct with respect to the given distribution family. ∎

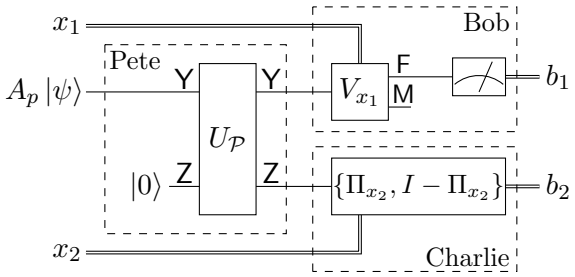


Figure 3.4: The pirating game specified to the AuthCP scheme.

3.5.2 Honest-Malicious Security

In this section, we prove the security of the scheme AuthCP in the honest-malicious setting. Formally, we prove the following theorem after establishing a few preliminary results.

Theorem 3.23. If the scheme QAS is an ϵ-total authentication scheme, then the scheme AuthCP described above is $(\frac{3}{2}\epsilon + \sqrt{2\epsilon})$-honest-malicious secure with respect to the uniform distribution R on point functions and challenge distributions $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_{p\in\{0,1\}^n}$, where R and $T_p^{(1/2)}$ are as defined in Definition 3.15 and Definition 3.16.

In fact, we can prove security with respect to a slightly more general set of challenge distributions. If we let $T_p^{(r)}$ be the distribution that samples p with probability r, and any other point uniformly, then for any $r \in [1/2, 1]$, our proof holds when Bob's input is chosen according to $T_p^{(r)}$ and Charlie's input is chosen according to $T_p^{(1/2)}$. (See Remark 3.26 following the proof of Theorem 3.23). If Bob gets the point with probability less than $1/2$, then it becomes easier for the adversary to win. Pete can simply send the program to Charlie, and give Bob a maximally mixed state. In that case, Bob will probably output 0, which is correct more than $1/2$ the time.

For the challenge distributions R and $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$, it is easy to see that Charlie's maximum guessing probability if he has no interaction with Pete, against which we measure security (see Definition 3.10), is $p^{\text{marg}} = 1/2$. We will use this fact in our security proof, which could likely be generalized to other distributions of Charlie's challenge with a different value of p^{marg}, but we do not analyze such cases.

The idea of the proof is the following. In the setting of the scheme AuthCP, the pirating game $\mathsf{PiratingGame}_{\hat{\mathcal{A}},\mathsf{AuthCP}}$ (see Figure 3.2) that an honest-malicious adversary must win is expressed in Figure 3.4. Without loss of generality, we can assume Pete's behaviour is modeled by a unitary $U_{\mathcal{P}}$ on the space YZ for an arbitrary auxiliary space Z initialized to a fixed state, which we will denote $|0\rangle$. (Note that this state can be composed of more than one qubit.)

Since the adversary is honest-malicious, we can assume that Bob is honestly evaluating the program, meaning he runs the verification procedure of the underlying authentication scheme, using the point he receives as the key, on the register Y, outputting 1 if and only if the flag register F is measured as "Acc".

Charlie's behaviour can be arbitrary, but without loss of generality, we can assume that it is specified by a family of two-outcome measurements on Z, $\{\Pi_x, I - \Pi_x\}_{x\in\{0,1\}^n}$. Charlie uses his challenge input x_2 to select a measurement to perform to obtain his output b_2.

We will break the proof into two cases. First, consider the case where $x_1 = p$. We can consider Pete's output in two orthogonal parts:

$$U_{\mathcal{P}}(A_p |\psi\rangle \otimes |0\rangle) = \left|\Gamma^p_{\mathrm{Acc}}\right\rangle + |\Gamma^p_{\mathrm{Rej}}\rangle\,, \tag{3.37}$$

where $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ is the part of the state that leads to Bob outputting 1 on input p, which is the correct bit for Bob to produce in this case. That is, $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ is the projection of Pete's output onto states where the Y register is supported on the image of A_p. When $x_1 = p$, only $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ contributes to a winning outcome. We show (Lemma 3.25) that this state is close (on average over p) to a state of the form $A_p |\psi\rangle\langle\psi| A_p^\dagger \otimes \xi_{\mathsf{Z}}$ for some subnormalized state ξ independent of p. Since Charlie's input is essentially independent of p, his winning probability is not much more than $1/2$, so the total winning probability in this case is not much more than $1/2$ (Lemma 3.24), which is scaled down by the trace of the subnormalized state ξ, representing the fact that the probability that Bob outputs the correct bit is $\left\|\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\|^2$.

The other case is when $x_1 \neq p$. In that case, we need to consider the contribution of both terms $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ and $\left|\Gamma^p_{\mathrm{Rej}}\right\rangle$, as well as their cross term. We can bound the contribution of the first term to just over $\frac{1}{2}\operatorname{Tr}(\xi)$ because Charlie's input is close to p-independent. As for the contribution of the second term, in the worst case, the second term is of the form $\alpha |0\rangle_{\mathsf{Y}} \otimes A_p |\psi\rangle$ for some scaling factor α. This corresponds to the strategy that Pete just sends Charlie the program. Charlie can evaluate the program and be correct with probability close to 1, and Bob will output 0 with probability close to 1, which is the correct bit in this case, since $x_1 \neq p$. So we trivially upper bound the contribution of this term by $\left\|\left|\Gamma^p_{\mathrm{Rej}}\right\rangle\right\|^2$. However, as this increases, the size of $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ and thus $\operatorname{Tr}(\xi)$ decreases, so the probability of being correct in the $x_1 = p$ case goes down. We find that the total contribution, ignoring the cross term, is at most negligibly more than $1/2$. Finally, we show that the cross-term is negligible by the correctness of the scheme AuthCP.

We first state and prove the necessary lemmas, before formalizing the above argument. The following lemma is simply stating that if Charlie gets an input state that is independent of the point p, then his guess as to whether $x_2 = p$ will be independent of p, and so will be correct with probability $1/2$.

Lemma 3.24. Suppose p is chosen uniformly at random, and $x_2 \leftarrow T_p^{(1/2)}$, so that with probability $1/2$, $x_2 = p$, and otherwise x_2 is uniform on $\{0,1\}^n \setminus \{p\}$. Let $\Pi_{x_2}^1 = \Pi_{x_2}$ and $\Pi_{x_2}^0 = I - \Pi_{x_2}$, so $\Pi_{x_2}^{P_p(x_2)} = \Pi_{x_2}$ when $x_2 = p$, and otherwise $\Pi_{x_2}^{P_p(x_2)} = I - \Pi_{x_2}$. Then, for any density matrix σ, $\mathbb{E}_{p,x_2} \operatorname{Tr}\left(\Pi_{x_2}^{P_p(x_2)}\sigma\right) = \frac{1}{2}$.

Proof: It suffices to compute

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p,x_2} \operatorname{Tr}\left(\Pi_{x_2}^{P_p(x_2)}\sigma\right) \\
&= \frac{1}{2^n}\sum_{p\in\{0,1\}^n}\left(\frac{1}{2}\operatorname{Tr}(\Pi_p\sigma) + \frac{1}{2}\frac{1}{2^n-1}\sum_{x_2\neq p}\operatorname{Tr}((I-\Pi_{x_2})\sigma)\right) \\
&= \frac{1}{2}\frac{1}{2^n}\sum_{p\in\{0,1\}^n}\operatorname{Tr}(\Pi_p\sigma) + \frac{1}{2}\left(1 - \frac{1}{2^n}\sum_{x_2\in\{0,1\}^n}\operatorname{Tr}(\Pi_{x_2}\sigma)\right) = \frac{1}{2}
\end{aligned} \tag{3.38}$$

which is the desired result. ∎

The following lemma tells us that in the part of Pete's output that will be accepted by Bob in the $x_1 = p$ case, Bob's input from Pete is essentially $A_p|\psi\rangle$, and Charlie's input from Pete is almost independent of p. Recall that A_p is an isometry, so $A_pA_p^\dagger$ is the projector onto $\operatorname{im}(A_p)$.

Lemma 3.25. Let $\left|\Gamma_{\mathrm{Acc}}^p\right\rangle_{\mathsf{YZ}} = (A_pA_p^\dagger \otimes I_{\mathsf{Z}})U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle)$ be the projection of Pete's output onto states supported on $\operatorname{im}(A_p)$ in the Y register. Then, there exists a subnormalized state $\xi \in \mathcal{D}(\mathsf{Z})$ such that

$$\mathop{\mathbb{E}}_p \Delta\left(\left|\Gamma_{\mathrm{Acc}}^p\right\rangle\!\left\langle\Gamma_{\mathrm{Acc}}^p\right|, A_p|\psi\rangle\langle\psi|A_p^\dagger \otimes \xi\right) \leq \epsilon. \tag{3.39}$$

Proof: By the security of the total authentication scheme, there exists a completely positive trace non-increasing map $\Psi : \mathcal{L}(\mathsf{Z}) \to \mathcal{L}(\mathsf{Z})$ such that

$$\begin{aligned}
&\mathop{\mathbb{E}}_p |p\rangle\langle p| \otimes (\overline{V}_p \otimes I_{\mathsf{Z}})(U_{\mathcal{P}})_{\mathsf{YZ}}(A_p|\psi\rangle\langle\psi|A_p^\dagger \otimes |0\rangle\langle 0|_{\mathsf{Z}})(U_{\mathcal{P}})_{\mathsf{YZ}}^\dagger(\overline{V}_p^\dagger \otimes I_{\mathsf{Z}}) \\
\approx_\epsilon &\mathop{\mathbb{E}}_p |p\rangle\langle p| \otimes (\overline{V}_pA_p)|\psi\rangle\langle\psi|_{\mathsf{M}}(A_p^\dagger\overline{V}_p^\dagger) \otimes \Psi(|0\rangle\langle 0|).
\end{aligned} \tag{3.40}$$

Using the fact that $\overline{V}_p = A_p^\dagger = A_p^\dagger(A_pA_p^\dagger)$ (that is, project onto states in the image of A_p, and then invert A_p), we have

$$\begin{aligned}
(\overline{V}_p \otimes I_{\mathsf{Z}})U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle_{\mathsf{Z}}) &= (\overline{V}_p \otimes I_{\mathsf{Z}})(A_pA_p^\dagger \otimes I_{\mathsf{Z}})U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle_{\mathsf{Z}}) \\
&= (\overline{V}_p \otimes I_{\mathsf{Z}})\left|\Gamma_{\mathrm{Acc}}^p\right\rangle.
\end{aligned} \tag{3.41}$$

Then by Lemma 3.1, and letting $\xi = \Psi(|0\rangle\langle 0|)$, we can continue from Equation 3.40 to get:

$$\begin{aligned} \mathop{\mathbb{E}}_{p} \Delta\left(\overline{V}_p \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right| \overline{V}_p^\dagger, \overline{V}_p A_p |\psi\rangle\langle\psi| A_p^\dagger \overline{V}_p^\dagger \otimes \xi\right) \leq \epsilon \\ \mathop{\mathbb{E}}_{p} \Delta\left(\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|, A_p |\psi\rangle\langle\psi| A_p^\dagger \otimes \xi\right) \leq \epsilon, \end{aligned} \tag{3.42}$$

where we used the fact that $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ and $A_p|\psi\rangle$ are both orthogonal to the kernel of $\overline{V}_p$, so the isometry $\overline{V}_p$ preserves the distance between them. ∎

We now proceed to prove our main theorem of this section, Theorem 3.23.

Proof: For a fixed p, x_1 and x_2, let q_1^{p,x_2} be the adversary's winning probability when $x_1 = p$, and let q_0^{p,x_1,x_2} be the winning probability when $x_1 \neq p$. Then the total winning probability is given by

$$\frac{1}{2} \mathop{\mathbb{E}}_{\substack{p\leftarrow R,\\ x_2\leftarrow T_p^{(1/2)}}} q_1^{p,x_2} + \frac{1}{2} \mathop{\mathbb{E}}_{\substack{p\leftarrow R,\\ x_1\leftarrow\{0,1\}^n\setminus\{p\},\\ x_2\leftarrow T_p^{(1/2)}}} q_0^{p,x_1,x_2}. \tag{3.43}$$

If $|\Gamma^p\rangle := U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle)$ is Pete's output for a fixed p, and $\Pi_{x_2}^{P_p(x_2)}$ is defined to be Π_p when $x_2 = p$ and $I - \Pi_{x_2}$ otherwise, we have that

$$\begin{aligned} q_1^{p,x_2} &= \left\| (|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}} \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})(V_p \otimes \mathrm{Id}_{\mathsf{Z}})|\Gamma^p\rangle \right\|^2 \\ \text{and} \quad q_0^{p,x_1,x_2} &= \left\| ((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes I_{\mathsf{M}} \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})(V_{x_1} \otimes \mathrm{Id}_{\mathsf{Z}})|\Gamma^p\rangle \right\|^2. \end{aligned} \tag{3.44}$$

We will upper bound q_1^{p,x_2} and q_0^{p,x_1,x_2} separately.

Recall that we can write Pete's output as

$$|\Gamma^p\rangle = \left|\Gamma^p_{\mathrm{Acc}}\right\rangle + \left|\Gamma^p_{\mathrm{Rej}}\right\rangle, \tag{3.45}$$

where

$$\begin{aligned} \left|\Gamma^p_{\mathrm{Acc}}\right\rangle &= (A_p A_p^\dagger \otimes I_{\mathsf{Z}})|\Gamma^p\rangle \\ \text{and} \quad \left|\Gamma^p_{\mathrm{Rej}}\right\rangle &= ((I_{\mathsf{Y}} - A_p A_p^\dagger) \otimes I_{\mathsf{Z}})|\Gamma^p\rangle. \end{aligned} \tag{3.46}$$

The $x_1 = p$ case. We begin by upper bounding q_1^{p,x_2}. We first show there is no contribution from the second term:

$$\begin{aligned}
&(|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}} \otimes \Pi_{x_2}^{P_p(x_2)})(V_p \otimes I_{\mathsf{Z}}) \left|\Gamma^p_{\mathrm{Rej}}\right\rangle \\
&= (|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}} \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})(V_p(I_{\mathsf{Y}} - A_pA_p^\dagger) \otimes I_{\mathsf{Z}}) \left|\Gamma^p\right\rangle \\
&= 0
\end{aligned} \tag{3.47}$$

because

$$\begin{aligned}
(\langle\mathrm{Acc}| \otimes I_{\mathsf{M}})V_p(I_{\mathsf{Y}} - A_pA_p^\dagger) &= \overline{V}_p(I_{\mathsf{Y}} - A_pA_p^\dagger) \\
&= A_p^\dagger A_p A_p^\dagger(I_{\mathsf{Y}} - A_pA_p^\dagger) \\
&= 0.
\end{aligned} \tag{3.48}$$

Above we used the fact that $\overline{V}_p = A_p^\dagger = A_p^\dagger(A_pA_p^\dagger)$ which is to say that $\overline{V}_p$ simply projects onto states in the image of A_p, and then inverts A_p. Recall by the prior lemma that we can approximate $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|$ by $A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi$ for a subnormalized density operator ξ which is independent of p. Let $\delta_p = \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right| - A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi$ be the error of this approximation for a specific p. Thus (omitting implicit tensored identities):

$$\begin{aligned}
q_1^{p,x_2} &= \mathrm{Tr}\left((|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_F \otimes \Pi_{x_2}^{P_p(x_2)})V_p \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right| V_p^\dagger\right) \\
&= \mathrm{Tr}\left(V_p^\dagger(|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_F \otimes \Pi_{x_2}^{P_p(x_2)})V_p(A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi + \delta_p)\right) \\
&\leq \mathrm{Tr}\left(V_p^\dagger \left|\mathrm{Acc}\right\rangle\!\left\langle\mathrm{Acc}\right| V_pA_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \Pi_{x_2}^{P_p(x_2)}\xi\right) \\
&\quad + \Delta\left(\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|, A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi\right) \\
&= \mathrm{Tr}\left(\Pi_{x_2}^{P_p(x_2)}\xi\right) + \Delta\left(\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|, A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi\right).
\end{aligned} \tag{3.49}$$

By Lemma 3.24, we have $\mathrm{Tr}(\xi)\,\mathbb{E}_{p,x_2} \mathrm{Tr}\left(\Pi_{x_2}^{P_p(x_2)} \frac{\xi}{\mathrm{Tr}(\xi)}\right) = \mathrm{Tr}(\xi)/2$. Combining this with Lemma 3.25, we conclude with:

$$\mathop{\mathbb{E}}_{p,x_2} q_1^{p,x_2} \leq \frac{\mathrm{Tr}(\xi)}{2} + \epsilon. \tag{3.50}$$

The $x_1 \neq p$ case: We will analyze the probability in three parts, as follows:

$$\begin{aligned} q_0^{p,x_1,x_2} &= \left\| ((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}}) V_{x_1} |\Gamma^p\rangle \right\|^2 \\ &\leq \underbrace{\left\| ((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes \Pi_{x_2}^{P_p(x_2)}) V_{x_1} \left|\Gamma^p_{\mathrm{Acc}}\right\rangle \right\|^2}_{=T_1^{p,x_1,x_2}} \\ &+ \underbrace{\left\| ((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes \Pi_{x_2}^{P_p(x_2)}) V_{x_1} \left|\Gamma^p_{\mathrm{Rej}}\right\rangle \right\|^2}_{=T_2^{p,x_1,x_2}} \\ &+ \underbrace{2\left| \left\langle \Gamma^p_{\mathrm{Rej}} \right| (V_{x_1}^\dagger (I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle \right|}_{=T_{\mathrm{cross}}^{p,x_1,x_2}}. \end{aligned} \tag{3.51}$$

We begin with the first term, whose analysis is similar to the $x_1 = p$ case. We have:

$$\begin{aligned} T_1^{p,x_1,x_2} &= \mathrm{Tr}\left((V_{x_1}^\dagger (I - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|) V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)}) (A_p |0\rangle\langle 0| A_p^\dagger \otimes \xi + \delta_p) \right) \\ &\leq \mathrm{Tr}\left(\Pi_{x_2}^{P_p(x_2)} \xi \right) + \Delta(\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|, A_p |0\rangle\langle 0| A_p^\dagger \otimes \xi). \end{aligned} \tag{3.52}$$

Thus, just as we concluded with Equation 3.50, we can conclude

$$\mathop{\mathbb{E}}_{p,x_1,x_2} T_1^{p,x_1,x_2} \leq \frac{\mathrm{Tr}(\xi)}{2} + \epsilon, \tag{3.53}$$

again, by Lemma 3.24 and Lemma 3.25.

For the second term, we will use the naive bound:

$$\begin{aligned} T_2^{p,x_1,x_2} &\leq \left\| \left|\Gamma^p_{\mathrm{Rej}}\right\rangle \right\|^2 \\ &= 1 - \left\| \left|\Gamma^p_{\mathrm{Acc}}\right\rangle \right\|^2 \\ &\leq 1 - \mathrm{Tr}\left(A_p |0\rangle\langle 0| A_p^\dagger \otimes \xi \right) + \Delta\left(\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|, A_p |0\rangle\langle 0| A_p^\dagger \otimes \xi \right) \\ &= 1 - \mathrm{Tr}(\xi) + \Delta\left(\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\!\left\langle\Gamma^p_{\mathrm{Acc}}\right|, A_p |0\rangle\langle 0| A_p^\dagger \otimes \xi \right). \end{aligned} \tag{3.54}$$

Then by Lemma 3.25, we have

$$\mathop{\mathbb{E}}_{p,x_1,x_2} T_2^{p,x_1,x_2} \leq 1 - \mathrm{Tr}(\xi) + \epsilon. \tag{3.55}$$

Finally, we upper bound the cross-term. The idea is that $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ and $\left|\Gamma^p_{\mathrm{Rej}}\right\rangle$ are orthogonal in the Y register. This is, of course, also true once we apply $\Pi_{x_2}^{P_p(x_2)}$ to the Z register. Applying the projector $V_{x_1}^\dagger (I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) V_{x_1}$ to the Y register could change

this, however, we will argue that, by correctness of the scheme, this projector cannot change the state $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ very much, because its first register is in $\mathrm{im}(A_p)$, and trying to decode with a different key, $x_1 \neq p$, should result in rejection with high probability. We have:

$$\begin{aligned}
&T^{p,x_1,x_2}_{\mathrm{cross}} \\
&= 2\left|\left\langle \Gamma^p_{\mathrm{Rej}}\right| (I_{\mathsf{Y}} \otimes \Pi^{P_p(x_2)}_{x_2}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle - \left\langle \Gamma^p_{\mathrm{Rej}}\right| (V^\dagger_{x_1} |\mathrm{Acc}\rangle\langle \mathrm{Acc}| V_{x_1} \otimes \Pi^{P_p(x_2)}_{x_2}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right| \\
&= 2\left|\left\langle \Gamma^p_{\mathrm{Rej}}\right| (V^\dagger_{x_1} |\mathrm{Acc}\rangle\langle \mathrm{Acc}| V_{x_1} \otimes \Pi^{P_p(x_2)}_{x_2}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right| \\
&\leq 2\left\|(\langle \mathrm{Acc}| V_{x_1} \otimes \Pi^{P_p(x_2)}_{x_2}) \left|\Gamma^p_{\mathrm{Rej}}\right\rangle\right\| \cdot \left\|(\langle \mathrm{Acc}| V_{x_1} \otimes \Pi^{P_p(x_2)}_{x_2}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\| \\
&\leq 2\left\|(\langle \mathrm{Acc}| V_{x_1} \otimes I_{\mathsf{Z}}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\| = 2\left\|(\overline{V}_{x_1} \otimes I_{\mathsf{Z}}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\|,
\end{aligned} \tag{3.56}$$

where we use the Cauchy-Schwarz inequality to obtain the third line. Since $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ is supported on $\mathrm{im}(A_p)$ in the first register, it has a Schmidt decomposition of the form:

$$\left|\Gamma^p_{\mathrm{Acc}}\right\rangle = \sum_\ell \beta_\ell (A_p |u_\ell\rangle)_{\mathsf{Y}} \otimes |v_\ell\rangle_{\mathsf{Z}}\,. \tag{3.57}$$

Taking the expectation and applying Jensen's inequality, we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{p,x_1,x_2} T^{p,x_1,x_2}_{\mathrm{cross}} &\leq 2 \mathop{\mathbb{E}}_{p,x_1} \sqrt{\sum_\ell |\beta_\ell|^2 \left\|\overline{V}_{x_1} A_p |u_\ell\rangle\right\|^2} \\
&\leq 2\sqrt{\sum_\ell |\beta_\ell|^2 \mathop{\mathbb{E}}_{p,x_1} \left\|\overline{V}_{x_1} A_p |u_\ell\rangle\right\|^2}.
\end{aligned} \tag{3.58}$$

We next want to appeal to Lemma 3.5, which implies that $\mathbb{E}_{p,x_1\leftarrow\{0,1\}^n} \left\|\overline{V}_{x_1} A_p |u\rangle\right\|^2 \leq 2\epsilon$ for any pure state $|u\rangle$. However, notice that p and x_1 are not uniformly distributed, because while p is uniform, x_1 is uniform over $\{0,1\}^n \setminus \{p\}$. However, since for any p we have $\left\|\overline{V}_p A_p |u\rangle\right\|^2 = 1$, we have:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{p\leftarrow\{0,1\}^n,\\ x_1\leftarrow\{0,1\}^n\setminus\{p\}}} \left\|\overline{V}_{x_1} A_p |u_\ell\rangle\right\|^2 \\
&= \frac{2^{2n}}{2^n(2^n-1)} \left(\mathop{\mathbb{E}}_{\substack{p\leftarrow\{0,1\}^n,\\ x_1\leftarrow\{0,1\}^n}} \left\|\overline{V}_{x_1} A_p |u_\ell\rangle\right\|^2 - \frac{1}{2^{2n}} \sum_{p\in\{0,1\}^n} \left\|\overline{V}_p A_p |u_\ell\rangle\right\|^2 \right) \\
&\leq 2\epsilon + \frac{1}{2^n-1} 2\epsilon - \frac{1}{2^n-1}
\end{aligned} \tag{3.59}$$

which is at most 2ϵ as long as $\epsilon \leq 1/2$. Thus we can continue:

$$\begin{aligned} \mathop{\mathbb{E}}_{p,x_1,x_2} T_{\text{cross}}^{p,x_1,x_2} &\leq 2\sqrt{\sum_{\ell} |\beta_\ell|^2}\sqrt{2\epsilon} \\ &= 2\sqrt{2\epsilon}. \end{aligned} \tag{3.60}$$

Combining Equation 3.53, Equation 3.55, and Equation 3.58 into Equation 3.51, we conclude the $x_1 \neq p$ case with:

$$\begin{aligned} \mathop{\mathbb{E}}_{p,x_1,x_2} q_0^{p,x_1,x_2} &\leq \mathop{\mathbb{E}}_{p,x_1,x_2} T_1^{p,x_1,x_2} + \mathop{\mathbb{E}}_{p,x_1,x_2} T_2^{p,x_1,x_2} + \mathop{\mathbb{E}}_{p,x_1,x_2} T_{\text{cross}}^{p,x_1,x_2} \\ &\leq \frac{1}{2}\operatorname{Tr}(\xi) + \epsilon + 1 - \operatorname{Tr}(\xi) + \epsilon + 2\sqrt{2\epsilon}. \end{aligned} \tag{3.61}$$

Conclusion. We can now combine Equation 3.50 and Equation 3.61 to get an upper bound on the total winning probability of

$$\begin{aligned} &\frac{1}{2}\mathop{\mathbb{E}}_{p,x_2} q_1^{p,x_2} + \frac{1}{2}\mathop{\mathbb{E}}_{p,x_1,x_2} q_0^{p,x_1,x_2} \\ &\leq \frac{1}{2}\left(\frac{1}{2}\operatorname{Tr}(\xi) + \epsilon\right) + \frac{1}{2}\left(1 - \frac{1}{2}\operatorname{Tr}(\xi) + 2\epsilon + 2\sqrt{2\epsilon}\right) \\ &= \frac{1}{2} + \frac{3}{2}\epsilon + \sqrt{2\epsilon}. \end{aligned} \tag{3.62}$$

Noting that $p^{\text{marg}}_{R,\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p} = \frac{1}{2}$ completes the proof. ∎

Remark 3.26. We note that if our challenge distribution instead chooses Bob's input so that $x_1 = p$ with probability r, for $r \geq 1/2$, and all other points uniformly, then Equation 3.62 would instead give us:

$$\begin{aligned} &r\mathop{\mathbb{E}}_{p,x_2} q_1^{p,x_2} + (1-r)\mathop{\mathbb{E}}_{p,x_1,x_2} q_0^{p,x_1,x_2} \\ &\leq r\left(\frac{1}{2}\operatorname{Tr}(\xi) + \epsilon\right) + (1-r)\left(1 - \frac{1}{2}\operatorname{Tr}(\xi) + 2\epsilon + 2\sqrt{2\epsilon}\right) \\ &= \frac{1}{2}(2r-1)\operatorname{Tr}(\xi) + 1 - r + (2-r)\epsilon + 2(1-r)\sqrt{2\epsilon} \\ &\leq \frac{1}{2}(2r-1) + 1 - r + (2-r)\epsilon + 2(1-r)\sqrt{2\epsilon} \\ &= \frac{1}{2} + (2-r)\epsilon + 2(1-r)\sqrt{2\epsilon}. \end{aligned} \tag{3.63}$$

We therefore have $((2-r)\epsilon + 2(1-r)\sqrt{2\epsilon})$-honest-malicious security under this more general challenge distribution, where Bob's input is distributed as $T_p^{(r)}$ and Charlie's input is

distributed as $T_p^{(1/2)}$.

3.6 Supplementary Materials for Chapter 3

We collect here supplementary material for Chapter 3. These are mainly proofs which were omitted from the main text for ease of exposition.

3.6.1 Proof of Lemma 3.4 Concerning the Existence of Total Quantum Authentication Schemes with Certain Parameters

We prove here Lemma 3.4. However, there are a few technical points which must be covered before. In short, these points tell us that we can substitute the key set $\mathcal{K}$ of a QAS with a set $\mathcal{K}'$ with little loss of security (Lemma 3.30), provided that there is map $f : \mathcal{K}' \to \mathcal{K}$ which maps a uniformly random variable on $\mathcal{K}'$ to an almost uniformly random variable on $\mathcal{K}$ (Definition 3.27). Our proof is then the application of these technical arguments to existing theorems concerning unitary 2-designs and how they can be used to construct a QAS (Theorems 3.31 and 3.32).

We begin by defining the notion of an ϵ-uniform map $f : \mathcal{A} \to \mathcal{B}$. This means that all pre-images of elements $b \in \mathcal{A}$ are, in some sense, ϵ-close to the same size.

Definition 3.27. A map $f : \mathcal{A} \to \mathcal{B}$ between finite sets is ϵ-uniform if

$$\frac{1}{2}\sum_{b\in\mathcal{B}}\left|\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|}-\frac{1}{|\mathcal{B}|}\right|\leq\epsilon. \tag{3.64}$$

We next show how ϵ-uniform maps can change an expectation.

Lemma 3.28. Let $f : \mathcal{A} \to \mathcal{B}$ be an ϵ-uniform map between finite sets and $g : \mathcal{B} \to [0, 1]$ be a map. Then, $|\mathbb{E}_a\, g(f(a)) - \mathbb{E}_b\, g(b)| \leq \epsilon$.

Proof: We first note that

$$\begin{aligned}&\left|\mathop{\mathbb{E}}_{a} g(f(a))-\mathop{\mathbb{E}}_{b} g(b)\right|\\&=\left|\sum_{b\in\mathcal{B}}\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|}\cdot g(b)-\sum_{b\in\mathcal{B}}\frac{1}{|\mathcal{B}|}\cdot g(b)\right|\\&\leq\left|\sum_{b\in\mathcal{B}}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|}-\frac{1}{|\mathcal{B}|}\right)g(b)\right|.\end{aligned} \tag{3.65}$$

Let $\mathcal{S} = \{b \in \mathcal{B} \mid \left|f^{-1}(b)\right||\mathcal{B}| - |\mathcal{A}| \geq 0\}$ and $\mathcal{S}' = \mathcal{B} \setminus \mathcal{A}$. We then have that

$$\begin{aligned}&\left|\sum_{b\in\mathcal{B}}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right)g(b)\right| \\ &= \left|\sum_{b\in\mathcal{S}}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right)g(b) + \sum_{b\in\mathcal{S}'}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right)g(b)\right|.\end{aligned} \tag{3.66}$$

Note that the $\mathcal{S}$ term in the right-hand side of the above equation is positive and the $\mathcal{S}'$ term is negative. Recalling that $g(b) \leq 1$, we then have

$$\begin{aligned}&\left|\sum_{b\in\mathcal{B}}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right)g(b)\right| \\ &\leq \max\left\{\sum_{b\in\mathcal{S}}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right), \sum_{b\in\mathcal{S}'}\left(\frac{1}{|\mathcal{B}|} - \frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|}\right)\right\}.\end{aligned} \tag{3.67}$$

Lastly, we note that

$$\sum_{b\in\mathcal{S}}\left(\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right) = \underbrace{\frac{1}{2}\sum_{b\in\mathcal{B}}\left|\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right|}_{=\epsilon} = \sum_{b\in\mathcal{S}'}\left(\frac{1}{|\mathcal{B}|} - \frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|}\right) \tag{3.68}$$

which follows from the fact that $\sum_{b\in\mathcal{S}}\left|f^{-1}(b)\right| = |\mathcal{A}| - \sum_{b\in\mathcal{S}'}\left|f^{-1}(b)\right|$ and direct calculations. ∎

To complete our exposition on ϵ-uniform maps, we give a lemma guaranteeing their existence for appropriate values of ϵ.

Lemma 3.29. For any finite sets $\mathcal{A}$ and $\mathcal{B}$, there is an $|\mathcal{B}|/(4|\mathcal{A}|)$-uniform map $f : \mathcal{A} \to \mathcal{B}$.

Proof: Without loss of generality, we can assume that $\mathcal{A} = \{0, \ldots, |\mathcal{A}| - 1\}$ and that $\mathcal{B} = \{0, \ldots, |\mathcal{B}| - 1\}$. It then suffices to take f to be $x \mapsto x \pmod{|\mathcal{B}|}$. Indeed, let $r = |\mathcal{A}|/|\mathcal{B}|$ and $\ell = |\mathcal{A}| - \lfloor r \rfloor |\mathcal{B}|$. We then have that

$$\begin{aligned}\frac{1}{2}\sum_{b\in\mathcal{B}}\left|\frac{\left|f^{-1}(b)\right|}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right| &= \frac{1}{2}\left(\ell\cdot\left|\frac{\lfloor r\rfloor+1}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right| + (|\mathcal{B}| - \ell)\cdot\left|\frac{\lfloor r\rfloor}{|\mathcal{A}|} - \frac{1}{|\mathcal{B}|}\right|\right) \\ &= 1 - r + 2\lfloor r\rfloor - \frac{\lfloor r\rfloor}{r} - \frac{\lfloor r\rfloor^2}{r}\end{aligned} \tag{3.69}$$

where we drop the absolute values by noting that $|\mathcal{B}|(\lfloor r\rfloor+1) - |\mathcal{A}| \geq 0$ and $|\mathcal{B}|\lfloor r\rfloor - |\mathcal{A}| \leq 0$.

Letting $r = w + p$ for $w \in \mathbb{N}$ and $0 \leq p < 1$ and noting that $\lfloor r \rfloor = w$, we have that

$$1 - r + 2\lfloor r \rfloor - \frac{\lfloor r \rfloor}{r} - \frac{\lfloor r \rfloor^2}{r} = \frac{p - p^2}{r} \leq \frac{1}{4r} \tag{3.70}$$

where the equality is obtained by direct calculation and the inequality by noting that $p - p^2 \leq \frac{1}{4}$. ∎

Now, we can apply ϵ-uniform maps to change the keys of an QAS scheme in a manner which will allows us to bound any change in the security of the scheme.

Lemma 3.30. Let $\mathsf{S} = \{(\mathsf{Auth}_k, \mathsf{Ver}_k)\}_{k \in \mathcal{K}}$ be an ϵ-total QAS. Let $\mathcal{K}'$ be a finite set and $f : \mathcal{K}' \to \mathcal{K}$ be an ϵ'-uniform map. Finally, for every $k' \in \mathcal{K}'$, we define

$$\mathsf{fAuth}_{k'} = \mathsf{Auth}_{f(k')} \quad \text{and} \quad \mathsf{fVer}_{k'} = \mathsf{Ver}_{f(k')}. \tag{3.71}$$

Then, $\mathsf{fS} = \{(\mathsf{fAuth}_k, \mathsf{fVer}_k)\}_{k \in \mathcal{K}'}$ is an $(\epsilon + \epsilon')$-total QAS.

Proof: Let Φ be an attack against the fS scheme. Then, it is also a valid attack against the S scheme. As S is an ϵ-total QAS, there exists a completely positive trace non-increasing map Ψ such that

$$\mathop{\mathbb{E}}_{k \in \mathcal{K}} |k\rangle\langle k| \otimes \mathsf{Ver}'_k \circ \Phi \circ \mathsf{Auth}_k(\rho) \approx_\epsilon \mathop{\mathbb{E}}_{k \in \mathcal{K}} |k\rangle\langle k| \otimes \mathsf{Ver}'_k \circ \Psi \circ \mathsf{Auth}_k(\rho) \tag{3.72}$$

for all $\rho \in \mathcal{D}(\mathsf{MZS})$. It then suffices to show that

$$\mathop{\mathbb{E}}_{k' \in \mathcal{K}'} \left|k'\right\rangle\!\left\langle k'\right| \otimes \mathsf{fVer}'_{k'} \circ \Phi \circ \mathsf{fAuth}_{k'}(\rho) \approx_{\epsilon+\epsilon'} \mathop{\mathbb{E}}_{k' \in \mathcal{K}'} \left|k'\right\rangle\!\left\langle k'\right| \otimes \mathsf{fVer}'_{k'} \circ \Psi \circ \mathsf{fAuth}_{k'}(\rho) \tag{3.73}$$

for all $\rho \in \mathcal{D}(\mathsf{MZS})$ to prove the claim.

Fix a state ρ and, to lighten notation, define the operators

$$\begin{aligned} \alpha_k &= \mathsf{Ver}'_k \circ \Phi \circ \mathsf{Auth}_k(\rho), & \beta_k &= \mathsf{Ver}'_k \circ \Psi \circ \mathsf{Auth}_k(\rho), \\ \phi_{k'} &= \mathsf{fVer}'_{k'} \circ \Phi \circ \mathsf{fAuth}_{k'}(\rho), \quad \text{and} & \varphi_{k'} &= \mathsf{fVer}'_{k'} \circ \Psi \circ \mathsf{fAuth}_{k'}(\rho). \end{aligned} \tag{3.74}$$

Next, note that $\phi_{k'} = \alpha_{f(k')}$, $\varphi_{k'} = \beta_{f(k')}$, and that by Lemma 3.1, the inequalities in Equations 3.72 and 3.73 are equivalent to

$$\mathop{\mathbb{E}}_{k} \Delta(\alpha_k, \beta_k) \leq \epsilon \quad \text{and} \quad \mathop{\mathbb{E}}_{k'} \Delta(\phi_{k'}, \varphi_{k'}) \leq \epsilon + \epsilon' \tag{3.75}$$

respectively. Finally, by Lemma 3.28, we have that

$$\left|\mathop{\mathbb{E}}_{k}\Delta(\alpha_k,\beta_k)-\mathop{\mathbb{E}}_{k'}\Delta(\phi_{k'},\varphi_{k'})\right| = \left|\mathop{\mathbb{E}}_{k}\Delta(\alpha_k,\beta_k)-\mathop{\mathbb{E}}_{k'}\Delta(\alpha_{f(k')},\beta_{f(k')})\right| \leq \epsilon' \tag{3.76}$$

which implies that $\mathbb{E}_{k'}\Delta(\phi_{k'},\varphi_{k'}) \leq \mathbb{E}_k\Delta(\alpha_k,\beta_k)+\epsilon' \leq \epsilon+\epsilon'$ which completes the proof. ∎

Finally, we recall a theorem which states that any unitary 2-design can be used to construct a total QAS. For our needs, it suffices to know that a unitary 2-design on n qubits is a set of unitary operators on n-qubits satisfying certain conditions (see [DCEL09]).

Theorem 3.31 ([AM17])**.** A unitary 2-design on $n+t$ qubits can be used to construct a $\left(2^{\frac{6-t}{3}}\right)$-total QAS for n qubits where the key set is the unitary 2-design.

Theorem 3.32 ([CLLW16])**.** There exists a unitary 2-design on q qubits with $2^{5q}-2^{3q}$ elements.

We can now give the proof of Lemma 3.4. Recall that Lemma 3.4 states that for any strictly positive n and k, there exists a $\left(5\cdot 2^{\frac{5n-k}{16}}\right)$-total QAS on n qubits with key set is precisely $\{0,1\}^k$.

Proof: We consider two cases: when $k \geq 5n+38$ and when $k < 5n+38$.

If $k < 5n+38$, then $5\cdot 2^{\frac{5n-k}{16}} \geq 1$ and so it suffices to find a 1-total QAS on n qubits. Taking the authentication and verification maps to be the identity (with an extra output of an accept flag in the verification map) for every key is sufficient.

We now consider the non-trivial case of $k \geq 5n+38$. From Theorems 3.31 and 3.32, we have the existence of a $\left(2^{\frac{6-t}{3}}\right)$-QAS on n qubits with a key set of size $2^{5(n+t)}-2^{3(n+t)}$. From Lemma 3.29, there exists an ϵ'-uniform map from $\{0,1\}^k$ to this key set for

$$\epsilon' = \frac{1}{4}\cdot\frac{2^{5(n+t)}-2^{3(n+t)}}{2^k} \tag{3.77}$$

Thus, by Lemma 3.30, there exist an ϵ-QAS on a qubits with key set $\{0,1\}^k$ for

$$\begin{aligned}\epsilon &\leq 2^{\frac{6-t}{3}} + \frac{1}{4}\cdot\frac{2^{5(n+t)}-2^{3(n+t)}}{2^k}\\ &= 2^{2-\frac{t}{3}} + 2^{5n+5t-k-2} - 2^{3n+3t-k-2}\\ &< 2^{2-\frac{t}{3}} + 2^{5n+5t-k-2}\end{aligned} \tag{3.78}$$

where we obtain the last inequality by simply dropping the third term which will allow us to simplify our upcoming analysis. It then suffices to find the optimal value of t for given and fixed n and k.

Direct methods (i.e. first and second derivatives) yield that the optimal value of t to minimize the above upper bound is

$$t_{\text{opt}} = \frac{12 - 3\log_2(15) + 3k - 15n}{16}. \tag{3.79}$$

However, from Theorem 3.31, t must be a non-negative integer. Thus, we take $t = \lfloor t_{\text{opt}} \rfloor$. The condition that $k \geq 5n + 38$ also yields that t is strictly positive.

Finally, substituting this choice of t in our upper bound of ϵ, we have that

$$\begin{aligned}
\epsilon &< 2^{2-\frac{\lfloor t_{\text{opt}} \rfloor}{3}} + 2^{5n+5\lfloor t_{\text{opt}} \rfloor - k - 2} \\
&\leq 2^{2-\frac{t_{\text{opt}}-1}{3}} + 2^{5n+5t_{\text{opt}}-k-2} \\
&= \frac{2^{\frac{23}{4}}}{15^{\frac{15}{16}}} \cdot 2^{\frac{5n-k}{16}} \\
&< 5 \cdot 2^{\frac{5n-k}{16}}
\end{aligned} \tag{3.80}$$

which is the desired result. ∎

3.6.2 Proof of Theorem 3.17 Concerning Malicious-Malicious Security and Correctness

Here, we give the proof of Theorem 3.17. We recall that our goal is to construct a malicious-malicious secure copy-protection scheme for point functions which is correct with respect to all distributions when given a malicious-malicious secure copy-protection scheme for point functions which is correct with respect to $\{T_p^{(1/2)}\}_p$. Our construction relies on pairwise independent permutations [NR99], which we define below.

Definition 3.33 (Pairwise Independent Permutation)**.** A pairwise independent permutation on $\{0,1\}^n$ is a family of functions $\{h_r : \{0,1\}^n \to \{0,1\}^n\}_{r \in \mathcal{R}}$ for some finite set $\mathcal{R}$ such that

1. every h_r is a permutation and
2. for all distinct $x_0, x_1 \in \{0,1\}^n$ and distinct $y_0, y_1 \in \{0,1\}^n$,

$$\Pr_r\left[(h_r(x_0), h_r(x_1)) = (y_0, y_1)\right] = \frac{1}{2^n}\frac{1}{2^n - 1} \tag{3.81}$$

where r is sampled uniformly at random from $\mathcal{R}$.

A straightforward construction of a pairwise independent permutation on bit strings $\{0,1\}^n$, mentioned in [NR99], is to consider all functions in the finite field of 2^n elements of the form $x \mapsto m \cdot x + b$ where $r = (m, b)$ and m is not the zero element.

Now, let $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ be a copy-protection scheme for point functions of length n and fix a pairwise independent family of permutations $\{h_r\}_{r\in\mathcal{R}}$ on the set $\{0,1\}^n$. We define another copy-protection scheme for point functions of length n, denoted $\mathsf{MIX}^{\mathsf{CP}} = (\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Protect}, \mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval})$, as follows:

$\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Protect}(p)$: On input of p (representing the point function P_p), output

$$\sum_{r\in\mathcal{R}} \frac{1}{|\mathcal{R}|} |r\rangle\langle r| \otimes \mathsf{CP.Protect}(h_r(p)). \tag{3.82}$$

$\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval}((r,\sigma), x)$: On input x and program (r,σ), output $\mathsf{CP.Eval}(\sigma, h_r(x))$.

We call a set of challenge distributions $\{D_p\}_{p\in\{0,1\}^n}$ *symmetric* if for $p \leftarrow R$, where we recall that R is the uniform distribution on points, and $(x_1, x_2) \leftarrow D_p$, the probability of any triple (p, x_1, x_2) is the same as $(\pi(p), \pi(x_1), \pi(x_2))$ for any permutation π on $\{0,1\}^n$. Equivalently, $D_p(x_1, x_2)$ can only depend on whether $x_1 = x_2$, whether $x_1 = p$ and whether $x_2 = p$. In particular, the set of product distributions $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$ is symmetric.

In the remainder of this section, we show the following:

Theorem 3.34. If the scheme CP is ϵ-malicious-malicious secure with respect to the uniform distribution on points R and any symmetric set of challenge distributions $\{D_p\}_{p\in\{0,1\}^n}$, and η-correct with respect to the distribution family $\{T_p^{(1/2)}\}_p$, then $\mathsf{MIX}^{\mathsf{CP}}$ is ϵ-malicious-malicious secure with respect to R and $\{D_p\}_{p\in\{0,1\}^n}$ and 2η-correct with respect to any distribution.

Note that this theorem clearly implies Theorem 3.17. We begin by showing that $\mathsf{MIX}^{\mathsf{CP}}$ is 2η-correct:

Lemma 3.35. If the scheme CP is η-correct with respect to the distribution family $\{T_p^{(1/2)}\}_p$, then $\mathsf{MIX}^{\mathsf{CP}}$ is 2η-correct with respect to any distribution family.

Proof: For any $p, x \in \{0,1\}^n$, and bit b, the probability that $\mathsf{CP.Eval}$ outputs b on input x, when evaluating the program $\mathsf{CP.Protect}(p)$ is $\mathrm{Tr}(|b\rangle\langle b| \mathsf{CP.Eval}(\mathsf{CP.Protect}(p), x))$. By the η-correctness of CP under the distribution $\{T_p^{(1/2)}\}_p$, we have, for any point $y \in \{0,1\}^n$ we have that

$$\frac{1}{2}\mathrm{Tr}\left(|1\rangle\langle 1| \mathsf{CP.Eval}(\mathsf{CP.Protect}(y), y)\right) + \frac{1}{2}\sum_{x\neq y}\frac{1}{2^n-1}\mathrm{Tr}\left(|0\rangle\langle 0| \mathsf{CP.Eval}(\mathsf{CP.Protect}(y), x)\right) \tag{3.83}$$

is at least $1-\eta$. This implies that

$$\frac{1}{2}\operatorname{Tr}\left(|1\rangle\langle 1|\,\mathsf{CP}.\,\mathsf{Eval}(\mathsf{CP}.\,\mathsf{Protect}(y),y)\right)\geq\frac{1}{2}-\eta, \tag{3.84}$$

and

$$\frac{1}{2}\sum_{x\neq y}\frac{1}{2^n-1}\operatorname{Tr}\left(|0\rangle\langle 0|\,\mathsf{CP}.\,\mathsf{Eval}(\mathsf{CP}.\,\mathsf{Protect}(y),x)\right)\geq\frac{1}{2}-\eta. \tag{3.85}$$

Fix p. The probability that $\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval}(\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Protect}(p),p)$ outputs the correct value of 1 is then

$$\begin{aligned}
&\operatorname{Tr}\left(|1\rangle\langle 1|\,\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval}(\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Protect}(p),p)\right)\\
=&\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\operatorname{Tr}\left(|1\rangle\langle 1|\,\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval}((r,\mathsf{CP}.\,\mathsf{Protect}(h_r(p))),p)\right)\\
=&\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\operatorname{Tr}\left(|1\rangle\langle 1|\,\mathsf{CP}.\,\mathsf{Eval}(\mathsf{CP}.\,\mathsf{Protect}(h_r(p))),h_r(p))\right)\\
\geq&1-2\eta
\end{aligned} \tag{3.86}$$

where the last inequality follows from Equation 3.84.

For any $p'\neq p$, the probability that $\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval}(\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Protect}(p),p')$ outputs the correct value of 0 is:

$$\begin{aligned}
&\operatorname{Tr}\left(|0\rangle\langle 0|\,\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Eval}(\mathsf{MIX}^{\mathsf{CP}}.\mathsf{Protect}(p),p')\right)\\
=&\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\operatorname{Tr}\left(|0\rangle\langle 0|\,\mathsf{CP}.\,\mathsf{Eval}(\mathsf{CP}.\,\mathsf{Protect}(h_r(p)),h_r(p'))\right)\\
=&\frac{1}{|\mathcal{R}|}\sum_{y\in\{0,1\}^n}\sum_{x\neq y}\sum_{\substack{h_r(p)=y,\\ h_r(p')=x}}\operatorname{Tr}\left(|0\rangle\langle 0|\,\mathsf{CP}.\,\mathsf{Eval}(\mathsf{CP}.\,\mathsf{Protect}(y),x)\right)\\
=&\frac{1}{2^n}\sum_{y\in\{0,1\}^n}\frac{1}{2^n-1}\sum_{x\neq y}\operatorname{Tr}\left(|0\rangle\langle 0|\,\mathsf{CP}.\,\mathsf{Eval}(\mathsf{CP}.\,\mathsf{Protect}(y),x)\right)\\
\geq&\,1-2\eta,
\end{aligned} \tag{3.87}$$

where the last equality follows by the fact that $\{h_r\}_{r\in\mathcal{R}}$ forms a pairwise independent family of permutations and the inequality follows from Equation 3.85. Thus, as this scheme is $1-2\eta$ correct on all points, it is $1-2\eta$ correct for any distribution. ∎

For the proof of security, we will actually show that any mixture of malicious-malicious secure schemes is malicious-malicious secure (Theorem 3.37). We also show that if $\mathsf{CP}.\,\mathsf{Protect}(p)$ is an ϵ-malicious-malicious secure encoding, then for each r the scheme that encodes p as $\mathsf{CP}.\,\mathsf{Protect}(h_r(p))$ is also ϵ-malicious-malicious secure (Lemma 3.36). The combination of

these two facts completes the proof of Theorem 3.34, since $\mathsf{MIX}^{\mathsf{CP}}$ is a mixture, in the sense of Theorem 3.37, of schemes of the form described in Lemma 3.36.

Lemma 3.36. Suppose the scheme CP is ϵ-malicious-malicious secure with respect to the uniform distribution on points R and any symmetric set of challenge distributions $\{D_p\}_{p\in\{0,1\}^n}$, and let π be any permutation on $\{0,1\}^n$. Then if $\mathsf{CP}'.\mathsf{Protect}(p) = \mathsf{CP}.\mathsf{Protect}(\pi(p))$, CP' is ϵ-malicious-malicious secure with respect to R and $\{D_p\}_p$.

Proof: Let $(\mathcal{P}, \mathcal{A}_1, \mathcal{A}_2)$ be an adversary for CP' in $\mathsf{PiratingGame}$ (see Figure 3.2) with success probability q, where $\mathcal{P}$'s action is given by the CPTP map $\Phi_{\mathcal{P}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{A}_1\mathsf{A}_2)$; the action of $\mathcal{A}_1$ (Bob) is described by a set of two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A_1, such that on challenge input x_1, Bob does the measurement $\{\Pi_{x_1}, I - \Pi_{x_1}\}$ on A_1 to obtain the bit b_1; and similarly, the action of $\mathcal{A}_2$ (Charlie) is described by a set of two-outcome measurements $\{\Lambda_x\}_{x\in\{0,1\}^n}$ on A_2. Let $\Pi^1_{x_1} = \Pi_{x_1}$ and $\Pi^0_{x_1} = I - \Pi_{x_1}$, so that $\Pi^{P_p(x_1)}_{x_1}$ projects onto the part of Bob's input that leads Bob to output the correct answer; and similarly define $\Lambda^b_{x_2}$. Then, since p is sampled with probability $R(p) = 1/2^n$, and given that p is sampled, a challenge (x_1, x_2) is sampled with probability $D_p(x_1, x_2)$, we have that

$$\begin{aligned}
q &= \sum_{p,x_1,x_2} \frac{1}{2^n} D_p(x_1,x_2) \operatorname{Tr}\left(((\Pi^{P_p(x_1)}_{x_1})^{\mathsf{A}_1} \otimes (\Lambda^{P_p(x_2)}_{x_2})^{\mathsf{A}_2}) \Phi_{\mathcal{P}}(\mathsf{CP}'.\mathsf{Protect}(p)) \right) \\
&= \sum_{p,x_1,x_2} \frac{1}{2^n} D_p(x_1,x_2) \operatorname{Tr}\left((\Pi^{P_p(x_1)}_{x_1} \otimes \Lambda^{P_p(x_2)}_{x_2}) \Phi_{\mathcal{P}}(\mathsf{CP}.\mathsf{Protect}(\pi(p))) \right) \\
&= \sum_{p,x_1,x_2} \frac{1}{2^n} D_p(\pi^{-1}(x_1),\pi^{-1}(x_2)) \operatorname{Tr}\left((\Pi^{P_{\pi^{-1}(p)}(\pi^{-1}(x_1))}_{\pi^{-1}(x_1)} \otimes \Lambda^{P_{\pi^{-1}(p)}(\pi^{-1}(x_2))}_{\pi^{-1}(x_2)}) \Phi_{\mathcal{P}}(\mathsf{CP}.\mathsf{Protect}(p)) \right) \\
&= \sum_{p,x_1,x_2} \frac{1}{2^n} D_p(x_1,x_2) \operatorname{Tr}\left((\Pi^{P_p(x_1)}_{\pi^{-1}(x_1)} \otimes \Lambda^{P_p(x_2)}_{\pi^{-1}(x_2)}) \Phi_{\mathcal{P}}(\mathsf{CP}.\mathsf{Protect}(p)) \right),
\end{aligned} \tag{3.88}$$

where in the last step, we used the symmetry of $\{D_p\}_p$. Then letting $\Pi'_x = \Pi_{\pi^{-1}(x)}$ and $\Lambda'_x = \Lambda_{\pi^{-1}(x)}$, and defining $\mathcal{A}'_1$ and $\mathcal{A}'_2$ so that their respective strategies are to perform the measurements $\{\Pi'_{x_1}, I - \Pi'_{x_1}\}$ and $\{\Lambda'_{x_2}, I - \Lambda'_{x_2}\}$ upon receiving their respective challenges x_1 and x_2, we have that $(\mathcal{P}, \mathcal{A}'_1, \mathcal{A}'_2)$ is an adversary for CP with success probability q. Thus, we must have $q \leq \frac{1}{2} + \epsilon$. ∎

Theorem 3.37. Let D be any distribution on points, and $\{D_p\}_p$ any set of challenge distributions. Suppose $\{\mathsf{CP}_r = (\mathsf{CP}_r.\mathsf{Protect}, \mathsf{CP}_r.\mathsf{Eval})\}_{r\in\mathcal{R}}$ is a family of point function encoding schemes, each with ϵ-malicious-malicious security with respect to D and $\{D_p\}_p$. Define a scheme MX by

$\mathsf{MX}.\mathsf{Protect}(p)$: On input p, output $\sum_{r\in\mathcal{R}} \frac{1}{|\mathcal{R}|} |r\rangle\langle r| \otimes \mathsf{CP}_r.\mathsf{Protect}(p)$.

$\mathsf{MX.Eval}((r,\sigma),x)$: On input x and program (r,σ), output $\mathsf{CP}_r.\mathsf{Eval}(\sigma,x)$.

Then MX is ϵ-malicious-malicious secure with respect to D and $\{D_p\}_p$.

Proof: Let $(\mathcal{P},\mathcal{A}_1,\mathcal{A}_2)$ be an adversary for MX in $\mathsf{PiratingGame}$ that succeeds with probability q, where $\mathcal{P}$'s action is given by the CPTP map $\Phi_{\mathcal{P}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{A}_1\mathsf{A}_2)$; the action of $\mathcal{A}_1$ (Bob) is described by a set of two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A_1, such that on challenge input x_1, Bob does the measurement $\{\Pi_{x_1}, I-\Pi_{x_1}\}$ on A_1 to obtain the bit b_1; and similarly, the action of $\mathcal{A}_2$ (Charlie) is described by a set of two-outcome measurements $\{\Lambda_x\}_{x\in\{0,1\}^n}$ on A_2. Let $\Pi^1_{x_1} = \Pi_{x_1}$ and $\Pi^0_{x_1} = I - \Pi_{x_1}$, so that $\Pi^{P_p(x_1)}_{x_1}$ projects onto the part of Bob's input that leads Bob to output the correct answer; and similarly define $\Lambda^b_{x_2}$. Then, since p is sampled with probabilty $D(p)$, and given that p is sampled, a challenge (x_1,x_2) is sampled with probability $D_p(x_1,x_2)$, we have:

$$\begin{aligned} q &= \sum_{p,x_1,x_2} D(p)D_p(x_1,x_2)\,\mathrm{Tr}\left(((\Pi^{P_p(x_1)}_{x_1})^{\mathsf{A}_1}\otimes(\Lambda^{P_p(x_2)}_{x_2})^{\mathsf{A}_2})\Phi_{\mathcal{P}}(\mathsf{MX.Protect}(p))\right) \\ &= \sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\sum_{p,x_1,x_2} D(p)D_p(x_1,x_2)\,\mathrm{Tr}\left((\Pi^{P_p(x_1)}_{x_1}\otimes\Lambda^{P_p(x_2)}_{x_2})\Phi_{\mathcal{P}}(|r\rangle\langle r|\otimes\mathsf{CP}_r.\mathsf{Protect}(p))\right), \end{aligned} \tag{3.89}$$

so there exists $r\in\mathcal{R}$ such that

$$\begin{aligned} q &\le \sum_{p,x_1,x_2} D(p)D_p(x_1,x_2)\,\mathrm{Tr}\left((\Pi^{P_p(x_1)}_{x_1}\otimes\Lambda^{P_p(x_2)}_{x_2})\Phi_{\mathcal{P}}(|r\rangle\langle r|\otimes\mathsf{CP}_r.\mathsf{Protect}(p))\right) \\ &= \sum_{p,x_1,x_2} D(p)D_p(x_1,x_2)\,\mathrm{Tr}\left((\Pi^{P_p(x_1)}_{x_1}\otimes\Lambda^{P_p(x_2)}_{x_2})\Phi_r(\mathsf{CP}_r.\mathsf{Protect}(p))\right), \end{aligned} \tag{3.90}$$

where $\Phi_r(\cdot) := \Phi_{\mathcal{P}}(|r\rangle\langle r|\otimes\cdot)$. Thus, if $\mathcal{P}'$ is an adversary who performs the map Φ_r, $(\mathcal{P}',\mathcal{A}_1,\mathcal{A}_2)$ is an adversary for CP_r that succeeds with probability at least q, and so we must have $q\le\frac{1}{2}+\epsilon$. ∎

3.6.3 Proof of Theorem 3.21 Concerning Secure Software Leasing for Compute-and-Compare Circuits

Here, we give the proof of Theorem 3.21 which pertains to the correctness and security of the construction of a secure software leasing scheme CC for compute-and-compare circuits from a secure software software leasing scheme for point functions PF. This construction is described in Section 3.4.4 and was originally given in in [CMP24].

We do this in two parts: we prove correctness in Lemma 3.38 and prove security in Lemma 3.39. We emphasize that these proofs follow the same arguments as in [CMP24].

Lemma 3.38. If for all $f \in F$, the scheme PF is η-correct with respect to the family of distributions $\{T^{PF}_{f,y}\}_{y\in\{0,1\}^m}$, then the scheme CC described in Section 3.4.4 is η-correct with respect to the family of distributions $\{T^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$.

Proof: The proof proceeds by considering the definition of correctness for CC and PF from Definition 3.13. Let sk be the secret key generated by $\mathsf{CC.Gen} = \mathsf{PF.Gen}$. For any pair $(f, y) \in \mathcal{F}$, observe that $\mathsf{CC.Lease}(\mathsf{sk}, (f, y))$ outputs the program given by (f, ρ) where $\rho = \mathsf{PF.Lease}(\mathsf{sk}, P_y)$. Thus the probability that $\mathsf{CC.Eval}$, given the program (f, ρ) and input $x \leftarrow T^{CC}_{f,y}$, outputs the correct value $\mathsf{CC}^f_y(x) = P_y(f(x))$ is given by

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x\leftarrow T^{CC}_{f,y}} \mathrm{Tr}\left(\left|\mathsf{CC}^f_y(x)\right\rangle\!\left\langle\mathsf{CC}^f_y(x)\right| \mathsf{CC.Eval}\left((f,\rho),x\right)\right)\\
=&\mathop{\mathbb{E}}_{x\leftarrow T^{CC}_{f,y}} \mathrm{Tr}\left(|P_y(f(x))\rangle\!\langle P_y(f(x))| \,\mathsf{PF.Eval}\left(\rho, f(x)\right)\right)\\
=&\mathop{\mathbb{E}}_{x\leftarrow T^{CC}_{f,y}} \mathrm{Tr}\left(|P_y(f(x))\rangle\!\langle P_y(f(x))| \,\mathsf{PF.Eval}\left(\mathsf{PF.Lease}(\mathsf{sk}, P_y), f(x)\right)\right)\\
=&\mathop{\mathbb{E}}_{z\leftarrow T^{PF}_{f,y}} \mathrm{Tr}\left(|P_y(z)\rangle\!\langle P_y(z)| \,\mathsf{PF.Eval}\left(\mathsf{PF.Lease}(\mathsf{sk}, P_y), z\right)\right) \geq 1-\eta,
\end{aligned} \tag{3.91}$$

where the third equality used the definition of $T^{PF}_{f,y}$ with $z = f(x)$ and the last inequality uses that PF is η-correct with respect to $\{T^{PF}_{f,y}\}_{y\in\{0,1\}^m}$.

To satisfy the correctness requirement for $\mathsf{CC.Verify}$, note that the probability that $\mathsf{CC.Verify}$ accepts $\rho = \mathsf{PF.Lease}(\mathsf{sk}, P_y)$ is:

$$\mathrm{Tr}(|1\rangle\!\langle 1| \,\mathsf{CC.Verify}(\mathsf{sk}, (f, y), \rho)) = \mathrm{Tr}(|1\rangle\!\langle 1| \,\mathsf{PF.Verify}(\mathsf{sk}, P_y, \rho)) \geq 1 - \eta, \tag{3.92}$$

where the last inequality uses the η-correctness of PF. With Equations 3.91 and 3.92, we can conclude that CC is η-correct with respect to the ensemble $\{T^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$. ∎

We now move to the security guarantee for the scheme CC. Observe that, the program distribution D for compute-and-compare programs is a distribution on $(f, y) \in \mathcal{F}$. For every f, we obtain a marginal distribution on y, which we denote by D_f which could be different in each f. Similarly, the set of challenge distributions for compute-and-compare functions, $\{D^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$, yields a distribution over $\{0, 1\}^n$ for each compute-and-compare function $(f, y) \in \mathcal{F} = F \times \{0, 1\}^m$. A set of challenge distributions for point functions would have a distribution over $\{0, 1\}^m$ for each point function P_y such that $y \in \{0, 1\}^m$. For every fixed f, we can obtain such a distribution $D^{PF}_{f,y}$ by sampling $x \leftarrow D^{CC}_{f,y}$ and outputting $f(x)$ (as defined in Theorem 3.21). We require the scheme PF to be secure against program distribution D_f and challenge distributions $\{D^{PF}_{f,y}\}_{y\in\{0,1\}^m}$ for *every* $f \in F$. We then get

the following theorem showing the security of CC, the scheme constructed above.

Lemma 3.39 (Restatement of Theorem 7 of [CMP24])**.** If for all $f^* \in F$ the scheme PF is ϵ_{f^*}-secure for

$$\epsilon_{f^*} = \left(p^{\text{triv}}_{D,\{D^{\text{CC}}_{f_y}\}} - p^{\text{triv}}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y}\right) + \epsilon \tag{3.93}$$

with respect to circuit distribution D_f and challenge distributions $\{D^{PF}_{f,y}\}_{y\in\{0,1\}^m}$, then the scheme CC described above is ϵ-secure with respect to D and $\{D^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$.

Note that in [CMP24] the above theorem was presented for a specific family of distributions. Our restatement is applicable to all distributions. We provide the proof below for completeness.

Proof: The proof proceeds via contradiction by showing that an adversary $\mathcal{A}_{\mathsf{CC}}$ that can win the game $\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{CC}},\mathsf{CC}}$ for compute-and-compare functions can be used to construct an adversary $\mathcal{A}_{\mathsf{PF}}$ that can win the game $\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{PF}},\mathsf{PF}}$ for point functions.

Assume that an adversary $\mathcal{A}_{\mathsf{CC}}$ can win the game $\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{CC}},\mathsf{CC}}$ instantiated with the program distribution D and challenge distributions $\{D^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$ for the scheme CC with probability strictly greater than $p^{\text{triv}}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} + \epsilon$. Then, there exists some $f^* \in F$ such that the probability that (f^*, y) is sampled from D for at least one y is non-zero, and $\mathcal{A}_{\mathsf{CC}}$ wins the game SSLGame with probability strictly greater than $p^{\text{triv}}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} + \epsilon$ conditioned on a pair of the form (f^*, y) being sampled.

We now construct an adversary $\mathcal{A}_{\mathsf{PF}}$ for PF that wins $\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{PF}},\mathsf{PF}}$ instantiated with the program distribution D_{f^*} and challenge distributions $\{D^{PF}_{f^*,y}\}_{y\in\{0,1\}^m}$ with probability strictly greater than $p^{\text{triv}}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y} + \epsilon_{f^*}$. This yields a contradiction.

Recall that the behaviour of $\mathcal{A}_{\mathsf{CC}}$ can be described in two parts (see also Figure 3.3). First, the adversary applies a CPTP map $\Phi_{\mathcal{A}_{\mathsf{CC}}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{YA})$ to his input ρ, sending the Y part to the Lessor (step 2 of SSLGame), and keeping the A part for himself. Later, when $\mathcal{A}_{\mathsf{CC}}$ receives the challenge x, he uses it to select a two outcome measurement $\{\Pi_x, I - \Pi_x\}$ with which to measure his register A to obtain a bit b (step 5 of SSLGame).

Consider an adversary $\mathcal{A}_{\mathsf{PF}}$ so that the game $\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{PF}},\mathsf{PF}}$ proceeds as follows:

1. The Lessor samples $y \leftarrow D_{f^*}$ and runs PF.Gen to obtain a secret key sk. She then sends $\rho = \mathsf{PF.Lease}(\mathsf{sk}, P_y)$ to $\mathcal{A}_{\mathsf{PF}}$.

2. $\mathcal{A}_{\mathsf{PF}}$ makes use of $\Phi_{\mathcal{A}_{\mathsf{CC}}}$, which expects as input an encoded compute-and-compare program produced by CC.Lease. Such a program has the form (f, ξ), for $f \in F$ and $\xi \in \mathcal{D}(\mathsf{Y})$ where ξ is an encoded point function produced by PF.Lease. In other words, the output space of CC.Lease is $\mathsf{Y}' = \mathsf{FY}$ where $\mathsf{F} = \text{span}\{|f\rangle : f \in F\}$. The pair

(f^*, ρ) fits this description. $\mathcal{A}_{\mathsf{PF}}$ computes $\sigma = \Phi_{\mathcal{A}_{\mathsf{CC}}}(f^*, \rho) \in \mathcal{D}(\mathsf{Y'A}) = \mathcal{D}(\mathsf{FYA})$ and sends the Y part of σ back to the Lessor and keeps the FA part.

3. The Lessor runs $\mathsf{PF.Verify}(\mathsf{sk}, P_y, \cdot) = \mathsf{CC.Verify}(\mathsf{sk}, (f^*, y), \cdot)$ on the register Y and aborts if the resulting bit v is not 1.

4. The Lessor samples $z \leftarrow D^{PF}_{f^*,y}$, which is an m-bit string, and sends z to $\mathcal{A}_{\mathsf{PF}}$.

5. $\mathcal{A}_{\mathsf{PF}}$ samples x according to the restriction of $D^{CC}_{f^*,y}$ to the set of pre-images $f^{*-1}(z)$. Denote this distribution by $D^{CC}_{f^*,y,z}$.[16] He then measures his register A using the two-outcome measurement $\{\Pi_x, I - \Pi_x\}$ and returns the resulting bit b to the Lessor.

6. The Lessor outputs 1 if and only if $b = P_y(x)$ and $v = 1$.

Let $\Pi^1_x = \Pi_x$ and $\Pi^0_x = I - \Pi_x$. Let $\Psi^{P_y}_{\mathsf{Ver}}$ be the map induced by $\mathsf{PF.Verify}(\mathsf{sk}, P_y, \cdot)$, and $\Psi^{\mathsf{CC}^{f^*}_y}_{\mathsf{Ver}}$ the map induced by $\mathsf{CC.Verify}(\mathsf{sk}, (f^*, y), \cdot)$, and note that these are the same map by construction. The winning probability of the adversary is given by

$$\begin{aligned}&\Pr[\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{PF}},\mathsf{PF}}]\\ &= \sum_{y,z\in\{0,1\}^m} \sum_{x\in\{0,1\}^n} D_{f^*}(y) D^{PF}_{f^*,y}(z) D^{CC}_{f^*,y,z}(x) \operatorname{Tr}\left((|1\rangle\langle 1| \otimes \Pi^{P_y(z)}_x)(\Psi^{P_y}_{\mathsf{Ver}} \otimes \mathrm{Id}_{\mathsf{FA}}) \circ \Phi_{\mathcal{A}_{\mathsf{CC}}}(f^*, \rho)\right).\end{aligned} \tag{3.94}$$

By definition of the various distributions, we can instead sample x and z by sampling x from $D^{CC}_{f^*,y}$ and setting $z = f(x)$. Hence, we can rewrite this probability as

$$\begin{aligned}&\sum_{y\in\{0,1\}^m, x\in\{0,1\}^n} D_{f^*}(y) D^{CC}_{f^*,y}(x) \operatorname{Tr}\left((|1\rangle\langle 1| \otimes \Pi^{P_y(f(x))}_x)(\Psi^{\mathsf{CC}^{f^*}_y}_{\mathsf{Ver}} \otimes \mathrm{Id}_{\mathsf{FA}}) \circ \Phi_{\mathcal{A}_{\mathsf{CC}}}(f^*, \rho)\right)\\ &= \sum_{y\in\{0,1\}^m, x\in\{0,1\}^n} D_{f^*}(y) D^{CC}_{f^*,y}(x) \operatorname{Tr}\left((|1\rangle\langle 1| \otimes \Pi^{\mathsf{CC}^f_y(x)}_x)(\Psi^{\mathsf{CC}^{f^*}_y}_{\mathsf{Ver}} \otimes \mathrm{Id}_{\mathsf{FA}}) \circ \Phi_{\mathcal{A}_{\mathsf{CC}}}(f^*, \rho)\right).\end{aligned} \tag{3.95}$$

Finally, note that $(f^*, \rho) = \mathsf{CC.Lease}(\mathsf{sk}, (f^*, y))$, so this is exactly the probability that the adversary $\mathcal{A}_{\mathsf{CC}}$ wins the game $\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{CC}},\mathsf{CC}}$ conditioned on f^* being the sampled function.

[16] As the adversary is computationally unbounded, they have sufficient computational resources to construct f^{*-1} given f^* and do this sampling. Note that by definition of $D^{PF}_{f^*,y}$ this set is not empty.

By assumption, this is strictly more than $p^{\text{triv}}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} + \epsilon$. Thus,

$$\begin{aligned}\Pr[\mathsf{SSLGame}_{\mathcal{A}_{\mathsf{PF}},\mathsf{PF}}] &> p^{\text{triv}}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} + \epsilon \\ &= p^{\text{triv}}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y} + \left(p^{\text{triv}}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} - p^{\text{triv}}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y}\right) + \epsilon \qquad (3.96) \\ &= p^{\text{triv}}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y} + \epsilon_{f^*}\end{aligned}$$

which is a contradiction. ∎

Chapter 4

Uncloneable Quantum Advice*

Chapter Abstract

The famous no-cloning principle has been shown recently to enable a number of uncloneable cryptographic primitives, including the copy-protection of certain functionalities. Here we address for the first time *unkeyed* quantum uncloneablity, via the study of a complexity-theoretic tool that enables a computation, but that is natively unkeyed: *quantum advice*. Remarkably, this is an application of the no-cloning principle in a context where the quantum states of interest are *not* chosen by a random process. We establish unconditional constructions for promise problems admitting *uncloneable quantum advice* and, assuming the feasibility of quantum copy-protecting certain functions, for languages with uncloneable advice. Along the way, we note that state complexity classes, introduced by Rosenthal and Yuen (ITCS 2022) — which concern the computational difficulty of synthesizing sequences of quantum states — can be naturally generalized to obtain state *cloning* complexity classes. We make initial observations on these classes, notably obtaining a result analogous to the existence of undecideable problems.

Our proof technique defines and constructs *ingenerable sequences of finite bit strings*, essentially meaning that they cannot be generated by any uniform circuit family with non-negligible probability. We then prove a generic result showing that the difficulty of accomplishing a computational task on uniformly random inputs implies its difficulty on any fixed, ingenerable sequence. We use this result to derandomize quantum cryptographic games that relate to cloning, and then incorporate a result of Kundu and Tan (arXiv 2022) to obtain uncloneable advice. Applying this two-step process to a monogamy-of-entanglement game yields a promise problem with uncloneable advice, and applying it to the quantum copy-protection of pseudorandom functions with super-logarithmic output lengths yields a language with uncloneable advice.

*This chapter reflects work which was completed in collaboration with Anne Broadbent and Martti Karvonen and which has been published in *IACR Communications in Cryptology* [BKL24]. It has been lightly edited from its published version for inclusion in this thesis, including by the incorporation of technical details found in the full version [BKL23].

Chapter Contents

4.1 Introduction

The no-cloning principle has been a key tenet of quantum information theory from its very early days [Die82, WZ82].[2] This principle tells us that it is, in general, impossible to create a perfect copy of an *unknown* quantum state. In these works, the state is unknown in the sense that it is selected uniformly at random from two possibilities. Follow up results, *e.g.*: [BH96, BDE+98, Wer98], also give bounds on the ability to create close-but-imperfect copies of states chosen from some set.

Quantum cryptography, which seeks to leverage quantum mechanical phenomena to achieve cryptographic goals, has greatly benefited from the no-cloning principle. Indeed, this principle establishes a stark qualitative difference between classical information, which can be perfectly copied, and quantum information, which cannot. In its most generic cryptographic application, it implies that a malicious eavesdropper cannot, in general, keep a perfect transcript of the quantum communication between two honest parties. This idea can be understood as being at the core of the security of many quantum cryptographic schemes, such as prepare-and-measure quantum key distribution protocols, *e.g.*: [BB84], and quantum money schemes, *e.g.*: [Wie83].

This leads to a fruitful avenue of research in quantum cryptography, often referred to as *uncloneable cryptography*, which can be summarized by the following question: *Which notions from classical information processing can be made "uncloneable" by the addition of quantum mechanics and its no-cloning principle?* One such example, as initially set-out by Aaronson [Aar09], is the task of copy-protecting a function or a program. Quantum copy-protection, for a family of functions $\mathcal{F}$, can be understood as the process of encoding a member f of $\mathcal{F}$ as a quantum state ρ_f such that the following are satisfied:

- Correctness: There is an honest procedure which outputs $f(x)$ on input of ρ_f and x.
- Security: It is infeasible to "split" ρ_f into two quantum systems, both allowing the evaluation of f using any, possibly malicious, procedures.

Clearly, this is impossible to achieve in a purely classical setting but the no-cloning principle opens the door to achieving this in the quantum setting. Since Aaronson's introduction of this idea, there has been a flurry of works considering the question of quantum copy-protection, *e.g.*: [ALL+21, CMP24], and the closely related notion of secure software leasing, *e.g.*: [AL21, BJL+21a].

[2] Diek, Wooters, and Zurek proved the no-cloning principle to argue that a proposed protocol for faster-than-light communication [Her82] was unphysical. However, a proof by Parks [Par70] of this principle already existed in the literature, albeit with no explicit connections to quantum information theory. See the work of Ortigoso [Ort18] for more on the history of the no-cloning principle.

A key aspect of existing constructions is that their security guarantees only hold if the function which is encoded is chosen at random from some set and not disclosed to the end-user. More formally, the family $\mathcal{F}$ is often understood as being the set of all maps $f(k,\cdot)$ generated from a single keyed function $f : \{0,1\}^\kappa \times \{0,1\}^d \to \{0,1\}^c$. The recipient of the copy-protected program then receives the state $\rho_{f(k,\cdot)}$ for a random key k which is unknown to them. More to the point, current approaches to quantum copy-protection cannot be applied to specific, unkeyed, functionalities and it is unclear if we can adapt them towards the copy-protection of the ability to solve a fixed and known decision problem.

In this work, we initiate the study of copy-protecting unkeyed functionalities, *i.e.*: copy-protecting fixed functions which are not chosen at random from a larger set. We frame our main results as the construction of *uncloneable quantum advice* for certain promise problems and languages. This framing occurs naturally since advice can already be understood as a program helping a user to solve a decision problem [Wat09]. One drawback of our work is that the resulting advice is either uncomputable, or *extremely* computationally expensive to generate. We leave addressing this issue to future work.

The fact that uncloneable advice can be achieved — as we show in this work — implies that, in some cases, an interesting trade-off can be made in quantum copy-protection. We do not need to restrict ourselves the copy-protecting *families* of functions: We can copy-protect *specific* functions, provided that we allow the party generating the program state or quantum advice to be computationally more powerful than the adversaries. The feasibility of such a trade-off had, to our knowledge, not yet been explored in the literature.

Organization. We complete our introduction by giving a high-level overview of our contributions in Section 4.1.1 and highlighting some open questions in Section 4.1.2. We then proceed to review some more technical preliminaries in Section 4.2 and formalize our novel technical tool, ingenerable sequences, in Section 4.3. We then examine cloning complexity classes in Section 4.4 before continuing with the main goal of this work in Section 4.5. In this last section, we briefly review quantum copy-protection, formally define uncloneable advice, and show how to instantiate this notion for one class of promise problems and one class of languages.

Acknowledgements. This work was supported by the Air Force Office of Scientific Research under award FA9550-20-1-0375, Canada's NSERC, and the University of Ottawa's Research Chairs program.

4.1.1 Contributions

We review here the main contributions in this work. We note that, for pedagogical reasons, this high-level review does not follow the same order as our detailed presentation.

4.1.1.1 Uncloneable Quantum Advice

Our first conceptual contribution is the formal definition of *uncloneable quantum advice*.

The prototypical complexity class with *quantum* advice is denoted **BQP**/qpoly. It contains precisely the decision problems $P = (P_{\text{yes}}, P_{\text{no}})$ for which there exists a fixed sequence of quantum states $(\rho_n)_{n\in\mathbb{N}}$ and an efficient (*i.e.* polynomial-time) family of quantum circuits $(C_n)_{n\in\mathbb{N}}$ such that on input of a problem instance $p \in P$ and the corresponding advice state $\rho_{|p|}$, the circuit $C_{|p|}$ accepts (respectively, rejects) with probability at least $2/3$ if $p \in P_{\text{yes}}$ (respectively, $p \in P_{\text{no}}$). The "poly" in "qpoly"signifies that the advice states can contain at most a polynomial number of qubits. The "q", of course, is for "quantum". Quantum advice was first introduced in [NY04] and further developed in [Aar05].

An important aspect of advice is that it is a fundamentally *non-uniform* resource and that there is no restriction on the computational complexity of generating it. In fact, the advice states may even encode the solutions to uncomputable problems.

In a survey of quantum complexity theory [Wat09], Watrous states that "[q]uantum advice is a formal abstraction that addresses this question: How powerful is quantum software?" Indeed, the circuit family $(C_n)_{n\in\mathbb{N}}$ from the previous paragraph can be understood as simply outputting the single bit which results from "running the software" encoded in the advice state $\rho_{|p|}$ with the problem instance p as the input. More formally, the decision problem P naturally defines a map $P_{\text{yes}} \cup P_{\text{no}} \to \{0,1\}$ where a problem instance p is mapped to 1 if it is an element of P_{yes} and mapped to 0 if it is an element of P_{no}. Thus, ρ_n can be seen as a quantum program which allows a holder to correctly evaluate this map on all inputs of length n.

Given the close parallel between quantum advice and quantum programs, it is natural that our definition of *uncloneable* quantum advice should follow closely the definition of *uncloneable* quantum programs, as formalized by quantum copy-protection.

More precisely, we will say that the quantum advice states $(\rho_n)_{n\in\mathbb{N}}$ for some decision problem P are *uncloneable* if no efficient adversary A can split the advice state ρ_n between two other adversaries B and C such that both of them can simultaneously solve independently sampled problem instances p_B and p_C of length n with more than a negligible advantage over $\frac{1}{2}$. This is precisely the security requirement that a copy-protection scheme would aim to provide for the map $P_{\text{yes}} \cup P_{\text{no}} \to \{0,1\}$ described above. We specify here that the problem instances p_B and p_C are sampled uniformly at random among all *yes* instances

with probability $\frac{1}{2}$ and uniformly at random among all *no* instances with probability $\frac{1}{2}$. Formalizing this, as we do in Definition 4.30 of Section 4.5.2, yields a complexity class which we denote **neglQP**/upoly, where the "u" denotes the uncloneability of the advice. The class **neglQP**, for its part, is precisely the class of problems which can be solved with negligible error in quantum polynomial time. As we now sketch, we define **neglQP**/upoly and not simply **BQP**/upoly because it is possible that these two classes are distinct.

It is straightforward to see that **neglQP** = **BQP** via the standard amplification technique of repeating a computation a polynomial number of times and taking a majority vote on the result. The same argument also yields **neglQP**/qpoly = **BQP**/qpoly, provided that polynomially many copies of the basic advice state are provided. However, it is unclear if **neglQP**/upoly = **BQP**/upoly, as this basic amplification technique fails to yield this equality. Indeed, if polynomially many copies of an advice state are provided, half can be given to one party and half to another. This breaks the uncloneability guarantee. It is also unclear if the other common amplification technique of Marriot and Watrous [MW05] yields this equality between complexity classes. Indeed, their technique does not require sending many copies of the advice state, but it does require changing the advice and there is no *a priori* guarantee that this preserves uncloneability.

4.1.1.2 Cloning Complexity Classes

As a second conceptual contribution, we define *cloning complexity classes.* The uncloneability guarantee for advice states discussed in the previous section is *not* simply that it is impossible to efficiently implement the transformation $\rho_n \mapsto \rho_n \otimes \rho_n$. Indeed, if it was possible to efficiently transform the advice state ρ_n into $\sigma_n \otimes \sigma_n$, where $\sigma_n \neq \rho_n$ but where each σ_n would still allow malicious users to correctly solve the decision problem, then ρ_n fails to be uncloneable advice in the sense described in Section 4.1.1.1. In fact, this distinction between the (in)ability to copy a quantum state and the (in)ability of copying its underlying operational capabilities is at the source of many interesting aspects and challenges of uncloneable cryptography.

Nonetheless, a clearly necessary condition for a sequence of states $(\rho_n)_{n\in\mathbb{N}}$ to be uncloneable advice for some decision problem is for the transformation $\rho_n \mapsto \rho_n \otimes \rho_n$ to be infeasible to implement. This naturally leads us to consider questions of the following form, with a particular interest in cases where the answer is negative:

> *Given a sequence $(\rho_n)_{n\in\mathbb{N}}$ of states, are there circuits $(C_n)_{n\in\mathbb{N}}$ satisfying some given computational constraints such that $C_n(\rho_n) \approx \rho_n \otimes \rho_n$ for all n?*

Our insight here, covered in Section 4.4, is that this type of question generalizes those captured by the *state complexity classes* recently studied by Metger, Rosenthal, and Yuen

[RY22, MY23]. As an explicit example of such a class, a sequence of states $(\rho_n)_{n\in\mathbb{N}}$ is in the state complexity class **statePSPACE** if and only if there exists a uniform polynomial-space circuit family $(C_n)_{n\in\mathbb{N}}$ such that C_n outputs a state overwhelmingly close to ρ_n on the empty input. The questions of the form above can be recast to form *cloning complexity classes*. These classes can then be understood as the "$1 \to 2$" generalization of the "$0 \to 1$" state complexity classes. We formalize this in Definition 4.15.

We note in passing that existing works on the no-cloning principle do not answer cloning complexity questions of the form presented above as they all consider the difficulty of cloning a state sampled at random from some larger set. Our cloning complexity classes explicitly remove this randomness from the cloning task.

The crux of this work is not focused on cloning complexity classes, but the tools and techniques we developed to instantiate uncloneable quantum advice do yield two interesting results on these classes and how they relate to state complexity classes:

- There is a sequence of fixed quantum states which cannot be *generated* by any uniform circuit family, but which can be cloned by an efficient circuit family (Lemma 4.17).
- There is a sequence of fixed quantum states which cannot be *cloned* by any uniform circuit family, even those implementing arbitrarily large computations (Lemma 4.18).

These two results, with an additional generic result showing that cloning cannot be more difficult than generating (Lemma 4.16) are visualized in Figure 4.1 which locates sequences of states with respect to two axes: one denoting the difficulty to generate them and one denoting the difficulty to clone them. Moreover, Lemma 4.18 can be interpreted as establishing a distinction between cloneable and uncloneable sequences of states, similar to how maps can be partitioned between those which are computable and those which are not. We leave the study of complexity-theoretic analogues (*e.g.*: establishing distinctions between efficiently cloneable and uncloneable sequences) for future work.

We emphasize again that cloning complexity classes are not the main topic of this work; however, they do offer an interesting way to connect some aspects of our work with existing results. We also believe that the study of cloning complexity classes could be of central interest and importance for further developments in uncloneable cryptography.

4.1.1.3 Ingenerable Sequences of Bit Strings

The main technical tool which we develop in this work are *ingenerable sequences* of bit strings. This is the content of Section 4.3.

In a nutshell, a sequence $(s_n)_{n\in\mathbb{N}}$ of bit strings $s_n \in \{0,1\}^*$ is computably (respectively, exponential-time) ingenerable if every uniform (respectively, exponential-time) cir-

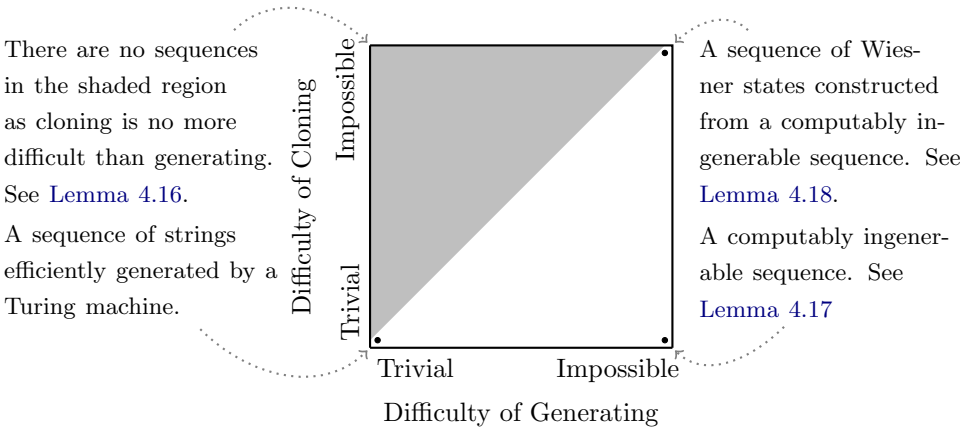


Figure 4.1: An annotated and informal representation of the space of sequences of quantum states contrasting the difficulty to generate and to clone a given sequence. Ingenerable sequences, which are novel to this work, are discussed in more details in Section 4.1.1.3.

cuit family $(C_n)_{n\in\mathbb{N}}$ eventually always fails at generating the elements of the sequence with sufficiently large probability. More precisely, there exists a polynomial p such that for every uniform (respectively, exponential-time) family of quantum circuits $(C_n)_{n\in\mathbb{N}}$ the quantity $\langle s_n| C_n(\varepsilon) |s_n\rangle$, where ε denotes the empty input, is eventually strictly smaller than $p(n)\cdot 2^{-|s_n|}$. We emphasize that the polynomial p is universal, in the sense that it does not depend on the circuit family $(C_n)_{n\in\mathbb{N}}$. However, the value of n at which point the inequality begins to be satisfied can change from one circuit family to the next.

A counting argument demonstrates that such sequences exist unconditionally. While computably ingenerable sequence are by their nature uncomputable, we also show that certain exponential-time ingenerable sequence can be computed by classical deterministic Turing machines in triple exponential time. Our argument is constructive, in the sense that it uniquely describes an ingenerable sequence, even in the case of computably ingenerable sequences where we cannot actually compute their elements. Further, we give an explicit classical deterministic algorithm running in triple exponential time which computes an exponential-time ingenerable sequence. These results are stated explicitly in Theorem 4.9.

Note that ingenerable sequences are purely classical information and so they can trivially be copied. This immediately yields Lemma 4.17 previously discussed in Section 4.1.1.2.

Ingenerable sequences are useful in this work due to a generic result, Theorem 4.10, relating the difficulty of a computational task on uniformly random inputs to its difficulty on inputs picked from an ingenerable sequence. Specifically, we show that if a uniform (respectively, exponential-time) circuit family $(C_n)_{n\in\mathbb{N}}$, which takes as input a string of at least super-logarithmic and at most polynomial length, outputs the bit 1 with negligible

probability on uniformly random inputs, then it must output 1 with negligible probability on inputs fixed from a computably (respectively, exponential-time) ingenerable sequence $(s_n)_{n\in\mathbb{N}}$. In other words, for any suitable ingenerable sequence $(s_n)_{n\in\mathbb{N}}$ it holds that

$$n \mapsto \mathop{\mathbb{E}}_{x} \langle 1| C_n(x) |1\rangle \text{ is negligible } \implies n \mapsto \langle 1| C_n(s_n) |1\rangle \text{ is negligible.} \tag{4.1}$$

We emphasize here that this holds for *all* uniform (respectively, exponential-time) circuit families $(C_n)_{n\in\mathbb{N}}$ while the ingenerable sequence $(s_n)_{n\in\mathbb{N}}$ remains fixed and independent of the circuit family.

This result allows us to derandomize certain cryptographic games related to cloning. For example, in Theorem 4.14, we construct a fixed sequence of Wiesner states[3] which cannot be cloned by any uniform circuit family. Specifically, we show that any uniform circuit family attempting to copy this fixed sequence will generate states having negligible fidelity with perfect copies of the original ones. This follows from applying our above result to the theorem of [MVW13] showing that the ability to clone a uniformly random Wiesner state is negligible in the length of this state. It then suffices to replace the uniformly random Wiesner state with one which is determined by a computably ingenerable sequence.

4.1.1.4 A Promise Problem with Uncloneable Advice from MoE games

We now present our first main result: a promise problem $P = (P_{\text{yes}}, P_{\text{no}})$ which admits uncloneable advice. This is the content of Section 4.5.3. Recall that for a promise problem we do not require P_{yes} and P_{no} to partition $\{0,1\}^*$: some strings may not be an element of either set, but we are promised to only get problem instances from $P_{\text{yes}} \cup P_{\text{no}}$.

First, we consider an exponential-time ingenerable sequence $(x_n)_{n\in\mathbb{N}}$ and we define the set P_{yes} to be all strings y where the inner product $x_n \cdot y$ is 1. Similarly, the set P_{no} will be all strings which have an inner product of 0 with the elements of the ingenerable sequence. Evidently, this problem cannot be efficiently solved without advice on the sequence $(x_n)_{n\in\mathbb{N}}$. Indeed, if we were able to solve this problem without any advice, we would be able to generate the ingenerable sequence. Now, the obvious advice to directly give is x_n as this would certainly allow an honest user to solve all problem instances. However, this advice is trivially cloneable as it is a classical piece of information. So, instead, we will give as advice the Wiesner state $|x_n^{\theta_n}\rangle$ for a string θ_n which will be determined shortly. We know from prior works on monogamy-of-entanglement (MoE) games [TFKW13] that it is infeasible for an adversary to split a random Wiesner state $\left|x^{\theta}\right\rangle$ in such a way that two parties can

[3] A Wiesner state [Wie83] is a tensor product of single-qubit computational basis states, $|0\rangle$ and $|1\rangle$, and their Hadamard conjugates, $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. For two bit strings x and θ, we let $\left|x^{\theta}\right\rangle = H^{\theta_1}|x_1\rangle \otimes \cdots \otimes H^{\theta_n}|x_n\rangle$, where H is the Hadamard operator, denote a Wiesner state.

recover the string x if they know θ, provided that the initial splitting adversary does not know θ. We recall and formalize this result as Lemma 4.37 in our detailed exposition. In some sense, the state $\left|x^{\theta}\right\rangle$ encodes information on the string x in an uncloneable way, which is what we need. However, three interrelated issues now arise.

The first problem is that if θ_n is fixed — and also x_n for that matter — how do we make sure that an adversary is not able to create copies of the fixed state $|x_n^{\theta_n}\rangle$? Lemma 4.37 considers the case of uniformly random Wiesner states, not a fixed sequence of such states. The solution is to also take θ_n from an exponential-time ingenerable sequence. For technical reasons, which can be roughly understood as making sure that x_n and θ_n are not too correlated, we will consider a single exponential-time ingenerable sequence $(s_n)_{n\in\mathbb{N}}$ and parse each string s_n as the concatenation (x_n, θ_n) of two strings of length n. We can then apply our generic derandomization theorem for ingenerable sequences, Theorem 4.10 discussed previously in Section 4.1.1.3, to the MoE game described above to show that no triplet of polynomial-time (or, even, exponential-time) adversaries can have a non-negligible advantage at winning this game when challenged with the fixed state $|x_n^{\theta_n}\rangle$.

Second, we have argued above that polynomial-time adversaries cannot win the MoE game of [TFKW13] with a non-negligible probability when challenged with the fixed sequence states $|x_n^{\theta_n}\rangle$. However, this game requires both guessing adversaries to correctly determine the complete string x_n to win. Solving the problem P only requires them to determine the inner product $x_n \cdot y$ for some string y, a task which may be easier. To bridge this gap, it suffices to use the recent result of Kundu and Tan [KT22], formalized in Lemma 4.1, which states that a negligible probability of both guessing adversaries determining x_n implies that the probability of both guessing adversaries determining $x_n \cdot y$, for independent and uniformly random values of y, is at most negligibly greater than $\frac{1}{2}$.

Finally, how does the honest user of the advice know the string θ_n? We have sketched above our argument as to why the advice state $|x_n^{\theta_n}\rangle$ is uncloneable, but not yet as to how it is useful. The naive way to use this state as advice for this problem is to measure it in the θ_n-Wiesner basis to obtain the string x_n. This requires the knowledge of θ_n, something which is ingenerable, and not known to the honest user so far. To overcome this issue, we will modify our problem to be a *promise* problem. The promise is that every problem instance y is prefixed with the fixed string $\theta_{|y|}$. In other words, we are giving θ_n as part of the problem instance. Because of this, θ_n is not accessible to an adversary attempting to split the advice state before knowing a problem instance.

In summary, we start with the monogamy-of-entanglement game of [TFKW13], derandomize it via exponential-time ingenerable sequences, and then apply Kundu and Tan's lemma. We then extract a promise problem with uncloneable advice from this construction.

We further note that this construction can be generalized by replacing the monogamy-

of-entanglement game with any sufficiently secure uncloneable encryption scheme [BL20] where the key k_n used and the message m_n to be be encoded play the role of θ_n and x_n, respectively, and would also be determined by an ingenerable sequence. The resulting construction, up to the use of ingenerable sequences to derandomize the scheme, is conceptually similar to Kundu and Tan's construction of uncloneable encryption for a single-bit message with *variable keys* [KT22].

As a last remark, we highlight that we use *exponential-time* ingenerable sequences in this construction. As previously mentioned, we show that there exist such sequences which can be computed in triple exponential time. Thus, if P is constructed from an exponential-time ingenerable sequence which can be computed in triple exponential time, then P can be enumerated by a classical deterministic Turing machine. Hence, our result is more than merely existential: we show how to construct P in such a way that the *yes* and *no* instances can be enumerated and we give an explicit algorithm for this enumeration.[4]

4.1.1.5 A Language with Uncloneable Advice from Copy-Protected PRFs

Our second main result is the construction of a *language* with uncloneable advice. Here, we *do* require P_{yes} and P_{no} to partition $\{0,1\}^*$. This is the content of Sections 4.5.4 and 4.5.5.

To construct this language, we lean even more on the connection between advice and programs: Our starting building block is the assumption that it is possible to copy-protect pseudorandom functions with super-logarithmic outputs. This can be achieved under certain assumptions [CLLZ21]. At a high level, we can understand the problem instances of the promise problem described in the previous section as pairs (θ, y) where the string θ was used as an "input" for the "program" encoded by the Wiesner state $\left|x^\theta\right\rangle$. Evaluating a Wiesner state in this paradigm then consists of measuring it in the θ-Wiesner basis and outputting the resulting x. The reason we obtained a promise problem, and not a language, is that this "program" has a deterministic answer on only one input: θ.

Copy-protected programs are not limited to only one "good input". If ρ_f is the copy-protected program of the function f, then evaluating ρ_f on any input x will yield $f(x)$ with high probability. Thus, we can reuse the same template as in the previous promise problem: problem instances are pairs (x, y) where x is treated as an input to a program state ρ_f to obtain a string $f(x)$ with near certainty. We then compute the inner product $f(x) \cdot y$ to determine if (x, y) is an element of the language. However, since ρ_f can be properly evaluated on *any* input, unlike the Wiesner state "program", we need not promise that the

[4] Strictly speaking, we give a deterministic classical algorithm in the proof of Theorem 4.9 computing in triple exponential time the elements of an exponential-time ingenerable sequence. It is then easy to see how to construct an algorithm enumerating P_{yes} and P_{no} when given an algorithm which computes θ_n and x_n for any $n \in \mathbb{N}$.

input x is some particular well-behaved string: it can be arbitrary. Thus, we can obtain a language, and not merely a promise problem from this type of construction.

Here, as for our promise problem, the result of Kundu and Tan (Lemma 4.1) is used to relate the difficulty of two guessers independently and simultaneously determining $f(x) \cdot y$ to their difficulty of determining $f(x)$. We also make use of exponential-time ingenerable sequences to derandomize the function which is copy-protected all the while maintaining the uncloneability of the resulting program.

4.1.2 Open Questions

This work is the first to study uncloneable advice and shows that this concept is, in principle, achievable. Beyond identifying other languages and promise problems which admit uncloneable advice, this work leaves many other open questions. We highlight two previously mentioned questions which we believe to be of particular interest.

Can we say more on the complexity of cloning quantum states? As illustrated in Figure 4.1 and discussed in Section 4.1.1.2, our work establishes the existence of sequences of states at the simultaneous extremes of cloning and generating complexity.

Are there scenarios between the extremes sketched above? In other words, how densely populated is the lower-right triangle of Figure 4.1? Our results imply the existence of a sequence of states that are impossible to clone in polynomial time — or indeed in exponential time — but which can be perfectly generated in triple exponential time. Can we close this gap? For example, can we find a fixed sequence which cannot be cloned in polynomial time, but can be generated in exponential time?

In a sense, our work initiates a theory of "computability of cloning" by showing a separation between sequences of states which can and cannot be cloned. We believe that further and more fine-grained work in this direction would help establish a theory on the complexity of cloning quantum states. We hope that such a theory would then provide useful tools for uncloneable cryptography and other aspects of quantum information theory.

What is the minimal computational power needed to produce uncloneable advice states? The previous question highlighted the "computational gap" we have found for uncloneable sequences of quantum states. We can also ask this question specifically about advice states: what is the minimal computational power needed to generate advice states for languages or promise problems which are uncloneable with respect to polynomial-time adversaries?

Can we "boost" the success probability of an honest party using uncloneable advice while maintaining uncloneability? As discussed in Section 4.1.1.1, the standard error reduction technique of giving multiple copies of the advice state fails to maintain the required uncloneability of the initial state. Furthermore, the well-known amplification technique of Marriott and Watrous [MW05], developed in the context of quantum *proofs* and the **QMA** complexity class, does not appear to be directly applicable to the context of uncloneable advice. We highlight two obstacles in applying the result of Marriott and Watrous to uncloneable advice. First, the Marriott-Watrous technique requires changing the proof state and it is unclear if this modification to the state would preserve the required uncloneability property. Second, and perhaps more fatal, is that directly applying the Marriott-Watrous technique to quantum *advice*, instead of quantum *proofs*, does not in general yield quantum *advice* as the required change to the quantum state depends on the problem instance for which we wish to increase the probability of success. This dependency is acceptable in the context of *proofs* as these may depend on the problem instance, but is unacceptable in the context of *advice* which must be independent of the problem instance.

Addressing this question would elucidate the relation between the two complexity classes **neglQP**/upoly and **BQP**/upoly. Moreover, finding a generic boosting technique for uncloneable advice could have wider repercussions in uncloneable cryptography by effectively lowering the necessary threshold for correctness.

4.2 Preliminaries

4.2.1 Notation and Terminology

Sets, bit strings, miscellaneous. We let $\mathbb{N}$ denote the non-negative integers, $\mathbb{R}$ denote the real numbers, $\mathbb{R}^+$ denote the strictly positive real numbers, and $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$, denote the non-negative real numbers. For any $n \in \mathbb{N}$, we let $[n]$ denote the set $\{i \in \mathbb{N} \,:\, 1 \leq i \leq n\}$. For any set $\mathcal{X}$, we let $\mathcal{P}(\mathcal{X})$ denote the power set of $\mathcal{X}$.

For all $n \in \mathbb{N}$, let $\{0,1\}^n$ be the set of bit strings of length n and $\{0,1\}^* = \cup_{n\in\mathbb{N}}\{0,1\}^n$ be the set of all bit strings of finite length. We denote the length of the bit string x by $|x|$ and denote the empty bit string by ε, which is to say that $\{0,1\}^0 = \{\varepsilon\}$. If x and y are two strings of the same length, we let $x \cdot y$ denote their inner product modulo 2. The result of concatenating two bit strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ is denoted by $(x,y) \in \{0,1\}^{n+m}$, using implicitly the isomorphism $\{0,1\}^n \times \{0,1\}^m \cong \{0,1\}^{n+m}$. For all $n \in \mathbb{N}$ we let 0^n and 1^n denote the all-zero and all-one bit strings in $\{0,1\}^n$, respectively.

Logarithms are always taken in base 2.

Probability theory. If $\mathcal{X}$ is a set, we use $\mathbb{E}_{x \leftarrow \mathcal{X}} f(x)$ to denote the expectation of f when x is sampled uniformly at random from $\mathcal{X}$. If α is a random variable, we use $x \leftarrow \alpha$ to denote that x is sampled according to α.

Asymptotic analysis. We will use the ω, Ω, and $\mathcal{O}$ notation to characterize the asymptotic behaviour of functions from $\mathbb{N}$ to $\mathbb{R}_0^+$ [Knu76].[5] To ease notation, we will often omit the argument variable in this notation, writing $\mathcal{O}(f)$ instead of $\mathcal{O}(f(n))$, or $\Omega(\log)$ instead of $\Omega(\log(n))$.

Given two functions $f, g : \mathbb{N} \to \mathbb{R}$, we say that f is *eventually smaller* than g if there exists a $n_0 \in \mathbb{N}$ such that $n \geq n_0 \implies f(n) \leq g(n)$. A function $f : \mathbb{N} \to \mathbb{R}$ is said to be *negligible* if it is eventually smaller than $n \mapsto n^{-c}$ for all $c \in \mathbb{N}$ (*e.g.* [Bel02]). Recall that a map 2^{-f} is negligible if and only if $f \in \omega(\log)$ and that if η is negligible, f is non-decreasing, and $f \in \Omega(n^r)$ for some $r \in \mathbb{R}^+$, then $\eta \circ f$ is also negligible.

Turing machines and their run times. We only consider deterministic Turing machines in this work and we refer to standard references, such as [AB09] or [LV19], for more details. We do, however, establish some notation and terminology.

A Turing machine $\mathtt{T}$ takes as input a string $x \in \{0,1\}^*$ and either (i) halts and produces as output a string $y \in \{0,1\}^*$, or (ii) never halts, in which case it does not produce an output. We write $\mathtt{T}(x) = y$ to denote the fact that the Turing machine $\mathtt{T}$ halts and produces the output y on input x. We say that a Turing machine $\mathtt{T}$ runs in polynomial-time if and only if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that on input of any string x, the machine $\mathtt{T}$ halts after at most $p(|x|)$ steps. For this paper, we identify efficient (quantum) computations with polynomial-time (quantum) computations. We also say that a Turing machine runs in exponential, double exponential, or triple exponential-time if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that on input of any string x, the machine halts after at most $2^{p(|x|)}$, $2^{2^{p(|x|)}}$, or $2^{2^{2^{p(|x|)}}}$ steps, respectively.

We let $\mathcal{T}$ denote the set of all Turing machines. Note that $\mathcal{T}$ is countable and so there exists a bijective map $\tau : \mathbb{N} \to \mathcal{T}$.

Finally, for maps $f : \mathbb{N} \to \mathbb{N}$, we say that a Turing machine $\mathtt{T}$ computes f, perhaps within some time bound, if on input of 1^n, i.e. the bit string composed of n instances of 1, it outputs $1^{f(n)}$. In particular, if $\mathtt{T}$ can compute $f : \mathbb{N} \to \mathbb{N}$ and $\mathtt{T}(1^n)$ halts within $T(n)$ steps, then $f(n) \leq T(n)$ as $\mathtt{T}$ can write at most one symbol per step.

[5] Note that we follow Brassard [Bra85] by always treating $\omega(f)$, $\Omega(f)$, and $\mathcal{O}(g)$ as sets and avoiding "one-way equations".

Quantum mechanics, Hilbert spaces, operators, and channels. Quantum mechanics, to the extent needed for this work, is a theory concerning linear operators on finite-dimensional complex Hilbert spaces. We refer the reader to the standard introductory textbooks of Watrous [Wat18] and Nielsen and Chuang [NC10] for more details. The sets of linear and unitary operators on a given Hilbert space $\mathcal{H}$ are denoted, respectively, by $\mathcal{L}(\mathcal{H})$ and $\mathcal{U}(\mathcal{H})$. We recall that a *channel* is a completely-positive trace preserving map $\Phi : \mathcal{L}(\mathcal{H}) \to \mathcal{L}(\mathcal{H}')$ and that channels precisely coincide with the set of all possible transformations that a quantum system may undergo over a finite time period. We also recall that by a *state*, we mean a density operator.

In this work, we assume that all Hilbert spaces are finite tensor products of $\mathbb{C}^2$, which is to say that we only consider spaces representing finite numbers of qubits. We suppose that $\left(\mathbb{C}^2\right)^{\otimes n} \simeq \mathbb{C}^{2^n}$ admits $\{|s\rangle\}_{s\in\{0,1\}^n}$ as an orthonormal basis and, when appropriate, we identify a string s with its corresponding density operator $|s\rangle\langle s|$.

The trace distance between two states is given by $\frac{1}{2}\|\rho - \sigma\|_1$ where $\|\cdot\|_1$ denotes the trace norm, which is also the $p = 1$ case of the more general Schatten p-norms. We denote the completely bounded trace norm, or "diamond norm" on channels by $\|\cdot\|_\diamond$ and recall that it is a good operational measure of distance between channels as $\|\Phi - \Psi\|$ essentially corresponds to the ability of a user to *distinguish* between Φ and Ψ [Wat18, Sec. 3.3].

Gate sets and quantum circuits. A *gate set* $\mathcal{G}$ is a finite set of channels. We assume that all elements of a gate set are channels of the form $L \mapsto ULU^\dagger$ for some unitary operator $U \in \mathcal{U}(\mathbb{C}^{2^{\otimes n}})$ acting on n qubits whose whose entries in the computational basis are algebraic. Note that n need not be identical across unitaries. There are two exceptions: A gate set $\mathcal{G}$ may also include the state preparation map $P : \mathbb{C} \to \mathcal{L}(\mathbb{C}^2)$, given by $c \mapsto c\,|0\rangle\langle 0|$, and the single-qubit trace $\mathrm{Tr} : \mathcal{L}(\mathbb{C}^2) \to \mathbb{C}$.

For any gate set $\mathcal{G}$, we let $\langle\mathcal{G}\rangle$ denote the set of channels generated by $\mathcal{G}$. More formally, $\langle\mathcal{G}\rangle$ is precisely the set of all channels Ψ which can be expressed as

$$\Psi = \left(\bigotimes_{j=1}^{w_1} \Psi_{1,j}\right) \circ \left(\bigotimes_{j=1}^{w_2} \Psi_{2,j}\right) \circ \cdots \circ \left(\bigotimes_{j=1}^{w_d} \Psi_{d,j}\right) = \bigcirc_{i=1}^{d} \bigotimes_{j=1}^{w_i} \Psi_{i,j} \tag{4.2}$$

for some non-negative integers $d, w_1, \ldots, w_d \in \mathbb{N}$ and channels $\Psi_{i,j} \in \mathcal{G}$. We call an expression of the form presented on the right-hand side of Equation 4.2 a *circuit* built from $\mathcal{G}$ implementing the map Ψ and we denote the set of all circuits built from $\mathcal{G}$ by $\langle\mathcal{G}\rangle_{\mathrm{c}}$. Formally, we distinguish between different circuits even if they implement the same channel but, when convenient, we identify a circuit with the channel it implements. A *circuit family* is a sequence of circuits $C = (C_n)_{n\in\mathbb{N}}$ built from the same gate set. We occasionally write

"a circuit family C" without the explicit sequence.

An *encoding* for a gate set $\mathcal{G}$ is a surjective map $e : \{0,1\}^* \to \langle \mathcal{G} \rangle_{\mathrm{c}}$. If $e(s) = C$, we call the string s a description of C. Encodings must satisfy additional conditions [Wat09] which we do not exhaustively list. We require an encoding e to be such that every $\Psi_{i,j}$ term (or lack thereof) in the circuit $e(s)$ must be efficiently computable from i, j, and s. Every gate set admits an encoding.

We say that C is an (a,b)-circuit if it implements a channel $C : \mathcal{L}(\mathbb{C}^{2^{\otimes a}}) \to \mathcal{L}(\mathbb{C}^{2^{\otimes b}})$ for $a, b \in \mathbb{N}$, which is to say that it takes a qubits as inputs and outputs b qubits. We say that C is a (a,b)-circuit family if each C_n is an $(a(n), b(n))$-circuit for $a, b : \mathbb{N} \to \mathbb{N}$. We say that a circuit family C is *uniform* if there exists a Turing machine $\mathtt{T}$ which, on input of 1^n, outputs a description of C_n in some prescribed encoding. We say that C is *polynomial-time*, *exponential-time*, or *triple exponential-time* if such a $\mathtt{T}$ exists and runs in polynomial-time, exponential-time, or triple exponential-time, respectively.

A gate set $\mathcal{G}$ is said to be *universal* if for any $\epsilon > 0$ and any channel Φ there exists a channel $\Psi \in \langle \mathcal{G} \rangle$ such that $\|\Phi - \Psi\|_\diamond \leq \epsilon$. As a consequence of the Solovay-Kitaev theorem [Kit97] and its algorithmic implementations [DN06, BGT21], for any computable map $t : \mathbb{N} \to \mathbb{N}$, any universal gate set $\mathcal{G}$, and any uniform circuit family C' built from any (possibly non-universal) gate set $\mathcal{G}'$, there exists a uniform circuit family C built from $\mathcal{G}$ such that $\|C_n - C'_n\|_\diamond \leq 2^{-t(n)}$. Moreover, if t is efficiently computable, we can replace "uniform" with "polynomial-time" or "exponential-time". Unless otherwise specified, we work with the universal gate set composed of the state preparation map, the single-qubit trace, and conjugations by the Toffoli, Hadamard, or phase-shift unitaries [Wat09].

4.2.2 A Simultaneous Quantum Goldreich-Levin Theorem

We recall a recent result of Kundu and Tan [KT22, Lemma 23] and recast it slightly to consider uniform and polynomial-time circuit families. In some sense, this result extends the quantum Goldreich-Levin theorem [AC02] to a setting with two adversaries. We note that an alternative proof of this lemma can also be found in [AKL23].

This lemma concerns a scenario where two non-communicating parties, Bob and Charlie, share a bipartite quantum state ρ_{x_B,x_C}. This state represents the information each party has on the value of two bit strings $(x_B, x_C) \in \{0,1\}^n \times \{0,1\}^n$ sampled according to a random variable ξ. The proof given in [KT22] for this lemma shows that if Bob and Charlie can simultaneously, independently, and respectively determine the values of the inner products $x_B \cdot y_B$ and $x_C \cdot y_C$ with probability at least $\frac{1}{2} + \delta$, where the probability is taken over the uniform random sampling of y_B and y_C (which are then given to Bob and Charlie, respectively) and the sampling of (x_B, x_C) from ξ, then they could *instead* simultaneously,

independently, and respectively determine x_B and x_C with probability at least $\delta^3/2$.

In more technically, the proof shows how to transform a circuit guessing the inner product $x_L \cdot y_L$ for either $L \in \{B, C\}$ into a circuit guessing x_L. The idea is to adapt the Bernstein-Vazirani algorithm [BV97] and replace the oracle for the inner product $y \mapsto x \cdot y$ with a purified version of the circuit guessing the inner product. A key property of this transformation, not explicitly mentioned in [KT22], is that this transformation preserves relevant complexity bounds in the sense that applying it circuit-by-circuit to a uniform (respectively, polynomial-time or exponential-time) circuit family yields another uniform (respectively, polynomial-time or exponential-time) circuit family.

In practice, as in [KT22], we take the contrapositive of the above. We state that if Bob and Charlie have a negligible probability of computing x_B and x_C, then they have at most a negligible advantage in computing $x_B \cdot y_B$ and $x_C \cdot y_C$.

Lemma 4.1. Let $\ell, b, c : \mathbb{N} \to \mathbb{N}$ be maps and, for all $n \in \mathbb{N}$, let ξ_n be a random variable on the set $\{0,1\}^{\ell(n)} \times \{0,1\}^{\ell(n)}$. For all $n \in \mathbb{N}$ and pairs (x_B, x_C) in the support of ξ_n, let ρ_{n,x_B,x_C} be a density operator on $b(n) + c(n)$ qubits.

If for every pair (B', C') of uniform (b, ℓ)- and (c, ℓ)-circuit families the map

$$n \mapsto \mathop{\mathbb{E}}_{(x_B,x_C)\leftarrow\xi_n} \langle x_B, x_C| \left(B'_n \otimes C'_n\right) (\rho_{n,x_B,x_C}) |x_B, x_C\rangle \tag{4.3}$$

is negligible, then for every pair (B, C) of uniform $(\ell + b, 1)$- and $(c + \ell, 1)$-circuit families, respectively, the map

$$n \mapsto \mathop{\mathbb{E}}_{\substack{y_B,y_C\leftarrow\{0,1\}^{\ell(n)} \\ (x_B,x_C)\leftarrow\xi_n}} \langle x_B \cdot y_B, x_C \cdot y_C| (B_n \otimes C_n) (y_B \otimes \rho_{n,x_B,x_C} \otimes y_C) |x_B \cdot y_B, x_C \cdot y_C\rangle - \frac{1}{2} \tag{4.4}$$

is negligible.

Moreover, the above holds if we replace all instances of "uniform" with "polynomial-time" or replace all instances of "uniform" with "exponential-time".

4.3 Ingenerable Sequences of Bit Strings

In this section, we define the notion of ingenerable sequences, demonstrate that such sequences exist, and illustrate a few of their simple properties.

Before proceeding, we emphasize that while our definitions and results are formulated in the language of quantum circuits, *there is nothing inherently quantum at play in this section*, with the exception of Section 4.3.3. In particular, the definitions can easily be

rephrased to fit a wide variety of computational models, including classical circuits or classical Turing machines. However, we believe the resulting notions are only interesting when the computational model is *probabilistic* and *uniform*. Moreover, the proofs of our main existence theorems essentially only rely on the assumption that there are countably many instances in the computational model.

Thus, essentially all definitions and results of this section, excluding Section 4.3.3, still hold if we were to replace the word "circuit", where the "quantum" is implicit, with the words "probabilistic classical Turing machine" or "uniform classical circuits supplied with additional random bits". Our definitions would also hold for "non-uniform circuits", classical or quantum, but our existence theorems would not. Of course, sequences which are ingenerable with respect to one model of computation need not be ingenerable with respect to another.

4.3.1 Definitions and Existence

We first introduce terminology concerning sequences of bit strings.

Definition 4.2. Let $\ell : \mathbb{N} \to \mathbb{N}$ be a map. An ℓ-sequence is a sequence of bit strings $(s_n)_{n\in\mathbb{N}}$ such that $s_n \in \{0,1\}^{\ell(n)}$ for all $n \in \mathbb{N}$.

In practice, we will concern ourselves with sequences of linear length, such as $2n$-sequences, or of polynomial length. However, the theory developed in this section is applicable to arbitrary ℓ-sequences.

Now, we formally define the notion of q-ingenerability for some map $q : \mathbb{N} \to \mathbb{R}$. In short, a sequence is q-ingenerable if every circuit family under consideration *eventually always fails* to produce the elements of the sequence with probability at least q. We also define a weaker notion of this idea, which we call *weak* ingenerability. A sequence is weakly ingenerable if every circuit family *fails infinitely often* to produce the elements of the sequence with probability at least q.

We immediately see that every q-ingenerable sequence is weakly q-ingenerable. Essentially all of our constructions will use ingenerable sequences, but weak ingenerability will be easier to compare and contrast with other existing definitions and notions.

Finally, we consider ingenerability with respect to two classes of circuit families. A computably ingenerable sequence is one which cannot be generated by any uniform circuit family. An exponential-time ingenerable sequence is one which cannot be generated by any exponential-time circuit family. This can be adapted straightforwardly to other classes of circuit families but these two are sufficient for us. The main advantage of exponential-time ingenerable sequences over computably ingenerable sequences is that certain such sequences are computable unlike computably ingenerable sequences, see Theorem 4.9.

Recall from the end of Section 4.2.1 that, for two maps $a, b : \mathbb{N} \to \mathbb{N}$, an (a, b)-circuit family C is a sequence of circuits $(C_n)_{n\in\mathbb{N}}$ where C_n takes $a(n)$ qubits as input and produces $b(n)$ qubits as output.[6]

Definition 4.3. Let $\mathcal{G}$ be a gate set and $q : \mathbb{N} \to \mathbb{R}$ be a map. An ℓ-sequence $(s_n)_{n\in\mathbb{N}}$ is computably (respectively, exponential-time) q-ingenerable with respect to $\mathcal{G}$ if for every uniform (respectively, exponential-time) $(0, \ell)$-circuit family C built from $\mathcal{G}$ the inequality

$$\langle s_n | C_n(\varepsilon) | s_n \rangle < q(n) \tag{4.5}$$

holds for all values, except a finite number of them, of $n \in \mathbb{N}$.

We say that $(s_n)_{n\in\mathbb{N}}$ is *weakly* computably (respectively, exponential-time) q-ingenerable with respect to $\mathcal{G}$ if we only require the above inequality to hold for infinitely many $n \in \mathbb{N}$.

It is trivial to show that the concept of ingenerability depends on the underlying gate set $\mathcal{G}$, which is why we make this set explicit in the definition. For example, consider the gate set composed of only the $|0\rangle$ state preparation map, $\mathcal{G} = \{c \mapsto c\,|0\rangle\langle 0|\}$, and the gate set $\mathcal{G}' = \mathcal{G} \cup \{L \mapsto XLX\}$ where we add conjugation by the $X = |0\rangle\langle 1| + |1\rangle\langle 0|$ unitary to $\mathcal{G}$. Clearly, the n-sequence $(1^n)_{n\in\mathbb{N}}$ is computably c-ingenerable with respect to $\mathcal{G}$ for any constant $c > 0$, but not with respect to $\mathcal{G}'$.

The above example is a bit contrived since $\mathcal{G}$ and $\mathcal{G}'$ are not very interesting gate sets. In particular, neither are universal. However, it appears difficult to show that the notion of q-ingenerability coincides even for two distinct but universal gate sets $\mathcal{G}$ and $\mathcal{G}'$. This is due to the fact that universality only guarantees that a circuit built from $\mathcal{G}'$ can *approximate* a circuit built from $\mathcal{G}$ to arbitrarily small *but non-zero* error and that q-ingenerability is defined via a simple inequality. For example, it is *a priori* possible for there to exist an ℓ-sequence $(s_n)_{n\in\mathbb{N}}$ and a uniform $(0, \ell)$-circuit family $(C_n)_{n\in\mathbb{N}}$ built from $\mathcal{G}$ such that $\langle s_n | C_n(\varepsilon) | s_n \rangle = 1$ for all $n \in \mathbb{N}$, but that for every uniform $(0, \ell)$-circuit family $(C'_n)_{n\in\mathbb{N}}$ built from $\mathcal{G}'$ we would have that $\langle s_n | C'_n(\varepsilon) | s_n \rangle < 1$ for all $n \in \mathbb{N}$. Such a sequence would be 1-ingenerable with respect to $\mathcal{G}'$, but not with respect to $\mathcal{G}$.

We will later give a slightly modified definition of ingenerability, Definition 4.6, which will be sufficient for our needs and which will avoid the issue of any possible dependence on the choice of the gate set. This definition will also have the added benefit of not requiring an explicit choice of a q map. First, however, we demonstrate the existence of q-ingenerable ℓ-sequences with respect to any given gate set $\mathcal{G}$ and for any suitable choices of q and ℓ.

[6] In this context, we interpret any $k \in \mathbb{N}$ as the constant map $n \mapsto k$.

Theorem 4.4. Let $\ell : \mathbb{N} \to \mathbb{N}$ and $q : \mathbb{N} \to \mathbb{R}$ be maps such that $q(n) \cdot 2^{\ell(n)} > n + 1$ for all $n \in \mathbb{N}$. Then, for any gate set $\mathcal{G}$ there exists an ℓ-sequence which is exponential-time q-ingenerable with respect to $\mathcal{G}$.

Proof: We first construct an ℓ-sequence $(s_n)_{n\in\mathbb{N}}$ of bit strings and then we show that it is exponential-time q-ingenerable with respect to $\mathcal{G}$. We interpret the output of all Turing machines in this proof as circuits constructed from $\mathcal{G}$.

Let $\mathcal{R} = \{r_k(n) = 2^{(n+2)^k}\}_{k\in\mathbb{N}}$ be a set of maps. In the context of this proof, these maps will be taken as bounds on the run times of various Turing machines. We note two useful properties of the set $\mathcal{R}$ before proceeding. First, $r_i \leq r_{i+1}$ for all $i \in \mathbb{N}$. Second, if $g \in \mathcal{O}(2^{n^{k_0}})$ for some $k_0 \in \mathbb{N}$, then there exists an $r_{k'} \in \mathcal{R}$ such that $g \leq r_{k'}$.

Now, let $\mathcal{T}$ be the set of all Turing machines and let $\Phi : \mathbb{N} \to \mathcal{T} \times \mathcal{R}$ be a map such that for all $\mathtt{T} \in \mathcal{T}$ there are infinitely many $r \in \mathcal{R}$ such that the pair $(\mathtt{T}, r)$ is in the image of Φ. By the previously discussed properties of $\mathcal{R}$, this implies that if $\mathtt{T}$ runs in exponential-time, then there exists an $n_{\mathtt{T}}$ such that $\Phi(n_{\mathtt{T}}) = (\mathtt{T}, r)$ and $\mathtt{T}(1^n)$ halts in at most $r(n)$ steps for all $n \in \mathbb{N}$. Such a map Φ exists as $\mathcal{T} \times \mathcal{R}$ is countable and hence there is a surjection from $\mathbb{N}$ to $\mathcal{T} \times \mathcal{R}$. Note that Φ being a surjection is a sufficient, *but not necessary*, condition. To ease later notation, let $\tau : \mathbb{N} \to \mathcal{T}$ and $\rho : \mathbb{N} \to \mathcal{R}$ be the unique maps satisfying $\Phi(n) = (\tau(n), \rho(n))$ for all $n \in \mathbb{N}$.

Now, for every $t, n \in \mathbb{N}$, define the set[7]

$$\mathcal{S}_{t,n} = \begin{cases} \left\{x \in \{0,1\}^{\ell(n)} : \langle x| \tau(t)(1^n)(\varepsilon) |x\rangle \geq q(n)\right\} & \text{if, on input of } 1^n, \ \tau(t) \text{ halts} \\ & \text{in at most } \rho(t)(n) \text{ steps and} \\ & \text{yields a } (0, \ell(n))\text{-circuit.} \\ \varnothing & \text{else.} \end{cases} \tag{4.6}$$

In other words, $\mathcal{S}_{t,n}$ is the set of strings generated with probability at least $q(n)$ by the circuit described by the t-th Turing machine on input 1^n, provided that the machine indeed halts in at most $\rho(t)(n)$ steps and outputs the description of a $(0, \ell(n))$-circuit. Under these assumptions, we have that $\sum_{x\in\{0,1\}^{\ell(n)}} \langle x| \tau(t)(1^n)(\varepsilon) |x\rangle = 1$ which implies that

$$|\mathcal{S}_{t,n}| \leq \frac{1}{q(n)}. \tag{4.7}$$

Note that this inequality also holds if $\tau(t)(1^n)$ does not halt within the specified number

[7] Recall that $\tau(t)(1^n)(\varepsilon)$ is a quantum state, when this expression is defined. Indeed, we consider the t-th Turing machine, $\tau(t)$, and run it on input 1^n. This yields the bit string $\tau(t)(1^n)$, assuming the machine halts. Implicitly, we interpret this bit string as a quantum circuit, which itself implicitly defines a quantum channel. Assuming this quantum channel takes as input the empty system, we apply it to the empty quantum state ε to obtain as output the state $\tau(t)(1^n)(\varepsilon)$.

of steps or does not yield a $(0, \ell(n))$-circuit as $|\mathcal{S}_{t,n}| = 0$ in this case and our assumptions imply that q must be strictly positive. Next, define

$$\mathcal{S}_n = \bigcup_{t \in \{0,\ldots,n\}} \mathcal{S}_{t,n}. \tag{4.8}$$

By our assumption on the maps ℓ and q, we have that

$$|\mathcal{S}_n| \leq (n+1) \cdot \frac{1}{q(n)} < 2^{\ell(n)}, \tag{4.9}$$

which implies that $\{0,1\}^{\ell(n)} \setminus \mathcal{S}_n$ is non-empty. For all $n \in \mathbb{N}$, we take s_n to be the first element, in lexicographic order, of $\{0,1\}^{\ell(n)} \setminus \mathcal{S}_n$. This yields an ℓ-sequence $(s_n)_{n\in\mathbb{N}}$.

Now, we show that $(s_n)_{n\in\mathbb{N}}$ is exponential-time q-ingenerable with respect to $\mathcal{G}$. Consider an exponential-time $(0,\ell)$-circuit family $(C_n)_{n\in\mathbb{N}}$. If no such circuit family exists, then $(s_n)_{n\in\mathbb{N}}$ is vacuously exponential-time q-ingenerable and we are done.

Let $\mathtt{T}$ be an exponential-time Turing machine which generates this circuit family. By our previous discussion on $\mathcal{R}$ and Φ, there exists a $t_{\mathtt{T}} \in \mathbb{N}$ such that $\Phi(t_{\mathtt{T}}) = (\mathtt{T}, r)$ and where $\mathtt{T}(1^n)$ halts in at most $r(n)$ steps. It is now sufficient to show that

$$n \geq t_{\mathtt{T}} \implies \langle s_n | \, C_n(\varepsilon) \, | s_n \rangle < q(n). \tag{4.10}$$

Assume this is not the case. Then, there is an $n' \geq t_{\mathtt{T}}$ such that $\langle s_{n'} | \, C_{n'}(\varepsilon) \, | s_{n'} \rangle \geq q(n')$. Thus, by definition, $s_{n'} \in \mathcal{S}_{t_{\mathtt{T}},n'} \subseteq \mathcal{S}_{n'}$. Contradiction, since $s_{n'} \in \{0,1\}^{\ell(n')} \setminus \mathcal{S}_{n'}$. ∎

Note that the above proof is constructive in the sense that once q, ℓ, $\mathcal{G}$ and Φ are fixed, then the sequence $(s_n)_{n\in\mathbb{N}}$ is uniquely determined.

We also note that by modifying the set $\mathcal{R}$ used in the proof, we can obtain sequences which are q-ingenerable with respect to circuit families generated by Turing machines operating under various time constraints. An important example for this work is the case where $\mathcal{R} = \{r_k(n) = \infty\}_{k\in\mathbb{N}}$ and where, in a slightly informal use of notation, we say that a Turing machine halts in at most infinitely many steps, denoted as ∞ steps, if and only if it does eventually halt. We state this formally below. The proof is simply to follow the proof of Theorem 4.4, but with this particular $\mathcal{R}$.

Theorem 4.5. Let $\ell : \mathbb{N} \to \mathbb{N}$ and $q : \mathbb{N} \to \mathbb{R}$ be maps such that $q(n) \cdot 2^{\ell(n)} > n + 1$ for all $n \in \mathbb{N}$. Then, for any gate set $\mathcal{G}$ there exists an ℓ-sequence which is computably q-ingenerable with respect to $\mathcal{G}$.

We now address the question of the possible dependence of our definition of ingenerable

sequences on the choice of the underlying gate set. Looking ahead, this work will be mainly concerned with ℓ-sequences which are computably or exponential-time $(p \cdot 2^{-\ell})$-ingenerable for a polynomial p, but that the particular polynomial is not important. With this in mind, we formulate the following definition of ingenerability which will be sufficient for our needs as well as independent of the choice of the gate set. We also note that this definition naturally extends to the notion of weak ingenerability.

Definition 4.6. Let $\ell : \mathbb{N} \to \mathbb{N}$ be a map. An ℓ-sequence is said to be computably (respectively, exponential-time) ingenerable if for every gate set $\mathcal{G}$ there exists a polynomial $p_{\mathcal{G}} : \mathbb{N} \to \mathbb{R}$ such that the sequence is computably (respectively, exponential-time) $(p_{\mathcal{G}} \cdot 2^{-\ell})$-ingenerable with respect to $\mathcal{G}$.

By the Solovay-Kitaev theorem, we will see that $(p \cdot 2^{-\ell})$-ingenerability with respect to a universal gate set for any polynomial $p : \mathbb{N} \to \mathbb{R}$ is a sufficient condition to be ingenerable. By definition, it is also a necessary condition. However, a technical condition on the computability of ℓ is needed to formally prove this. The following lemma, which characterizes when ingenerability is a trivial property, will help us account for this technicality.

Lemma 4.7. Let $\ell : \mathbb{N} \to \mathbb{N}$ be a map. The following two statements are equivalent:

1. Every ℓ-sequence is computably (respectively, exponential-time) ingenerable.
2. Either ℓ is uncomputable (respectively, can't be computed in exponential-time), or $\ell \in \mathcal{O}(\log)$.

Proof: We first show that the second statement implies the first.

If ℓ is uncomputable (respectively, cannot be computed in exponential-time), then there are no uniform (respectively, exponential-time) $(0, \ell)$-circuit families, as a Turing machine producing circuits with the appropriate number of output qubits could be used to compute ℓ. As there are no suitable $(0, \ell)$-circuit family, it is vacuously true that every ℓ-sequence is computably (respectively, exponential-time) ingenerable.

If $\ell \in \mathcal{O}(\log)$, then there exists $n', c \in \mathbb{N}$ such that $n \geq n' \implies \ell(n) \leq c \log(n)$, which implies that $2^{-\ell(n)} \geq n^{-c}$ for all sufficiently large values of n. Take $p(n) = n^{c+1}$ and set $\tilde{n} = \max\{n', 2\}$ so that $n \geq \tilde{n} \implies p(n) \cdot 2^{-\ell(n)} \geq n \geq 2$. Now, consider any ℓ-sequence $(s_n)_{n \in \mathbb{N}}$ and any uniform $(0, \ell)$-circuit family $(C_n)_{n \in \mathbb{N}}$ built from any gate set $\mathcal{G}$. It follows that for all $n \geq \tilde{n}$, we have that

$$\langle s_n | \, C_n(\varepsilon) \, | s_n \rangle \leq 1 < 2 \leq p(n) \cdot 2^{-\ell(n)}. \tag{4.11}$$

Thus, $(s_n)_{n \in \mathbb{N}}$ is computably ingenerable and hence also exponential-time ingenerable.

We now show that the first statement implies the second. Assume that every ℓ-sequence is computably (respectively, exponential-time) ingenerable and assume that ℓ is computable (respectively, computable in exponential-time), as otherwise we are done. Consider the gate set $\mathcal{G}$ composed of precisely the $|0\rangle$ state preparation map, the ℓ-sequence $(0^{\ell(n)})_{n\in\mathbb{N}}$, and the uniform (respectively, exponential-time) $(0,\ell)$-circuit family $(C_n)_{n\in\mathbb{N}}$ constructed from $\mathcal{G}$ where C_n is the parallel composition of $\ell(n)$ instances of the $|0\rangle$ state preparation map. Clearly, $\left\langle 0^{\ell(n)}\right| C_n(\varepsilon) \left|0^{\ell(n)}\right\rangle = 1$. But, $(0^{\ell(n)})_{n\in\mathbb{N}}$ is assumed to be computably (respectively, exponential-time) ingenerable. Thus, there exists a polynomial $p_{\mathcal{G}} : \mathbb{N} \to \mathbb{R}$ such that $1 < p_{\mathcal{G}}(n) \cdot 2^{-\ell(n)}$ for all sufficiently large values of n. Hence $\ell(n) \leq \log(p_{\mathcal{G}}(n))$ for all sufficiently large values of n, which implies that $\ell \in \mathcal{O}(\log)$. ∎

Lemma 4.8. Let $\mathcal{G}$ be a universal gate set and $\ell : \mathbb{N} \to \mathbb{N}$ be any map (respectively, any efficiently computable map). If an ℓ-sequence is computably (respectively, exponential-time) $(p \cdot 2^{-\ell})$-ingenerable with respect to $\mathcal{G}$ for a polynomial $p : \mathbb{N} \to \mathbb{R}$, then it is computably (respectively, exponential-time) ingenerable.

Proof: Let $(s_n)_{n\in\mathbb{N}}$ be an ℓ-sequence which is $(p \cdot 2^{-\ell})$-ingenerable with respect to $\mathcal{G}$. Further, assume that ℓ is computable, as otherwise $(s_n)_{n\in\mathbb{N}}$ is already known to be ingenerable by Lemma 4.7.

Let $\mathcal{G}'$ be any gate set and let $(C'_n)_{n\in\mathbb{N}}$ be a uniform (respectively, exponential-time) family of $(0,\ell)$-circuits built from this set. Now, let $(\tilde{C}_n)_{n\in\mathbb{N}}$ be a uniform family of $(0,\ell)$-circuits with gates from the universal gate set $\mathcal{G}$ such that $\|C_n - \tilde{C}_n\|_\diamond \leq 2^{-\ell(n)}$ for all $n \in \mathbb{N}$. Such a uniform (respectively, exponential-time) family exists by the Solovay-Kitaev theorem. Since $(s_n)_{n\in\mathbb{N}}$ is computably (respectively, exponential-time) $(p \cdot 2^{-\ell})$-ingenerable with respect to $\mathcal{G}$, we have that

$$\langle s_n| C_n(\varepsilon) |s_n\rangle \leq \langle s_n| \tilde{C}_n(\varepsilon) |s_n\rangle + 2^{-\ell(n)} < (p(n)+1) \cdot 2^{-\ell(n)} \tag{4.12}$$

for all sufficiently large values of n. Hence, the ℓ-sequence $(s_n)_{n\in\mathbb{N}}$ is computably (respectively, exponential-time) $(p+1)\cdot 2^{-\ell}$-ingenerable with respect to $\mathcal{G}'$. This yields the desired result as $p+1$ is a polynomial. ∎

The main result of this section is the following theorem which asserts the existence of computably ingenerable and exponential-time ℓ-sequences for any suitable choice of ℓ.

Theorem 4.9. There exists a computably ingenerable ℓ-sequence for any map $\ell : \mathbb{N} \to \mathbb{N}$.

Moreover, if ℓ is efficiently computable, then there exists an exponential-time ingenerable ℓ-sequence which can be computed in triple exponential-time by a classical deterministic

Turing machine.

Proof: Let $\mathcal{G}$ be a universal gate set and let $p(n) = n + 2$. Then, by Theorem 4.5, there exists a computably $(p \cdot 2^{-\ell})$-ingenerable ℓ-sequence with respect to $\mathcal{G}$, which by Lemma 4.8 is a computably ingenerable ℓ-sequence.

Keeping the same $\mathcal{G}$ and p and assuming that ℓ can be efficiently computed, we now show how to fix the map Φ in the proof of Theorem 4.4 to obtain an ℓ-sequence which is exponential-time ingenerable with respect to $\mathcal{G}$ but which can be computed in triple exponential-time by a classical deterministic Turing machine. By Lemma 4.8, this is an exponential-time ingenerable sequence.

Recall that $\mathcal{R} = \left\{r_k(n) = 2^{(n+2)^k}\right\}_{k\in\mathbb{N}}$. Let $\langle\cdot\rangle : \{0,1\}^* \to \mathcal{T}$ be a mapping from bit strings to Turing machines and let $\mathtt{U}$ be a Turing machine satisfying

$$\mathtt{U}(s, x, 1^t) = \begin{cases} (1, \langle s\rangle(x)) & \text{if } \langle s\rangle \text{ halts within t steps on input } x \\ 0 & \text{else} \end{cases} \tag{4.13}$$

and which runs in polynomial time. In other words, $\mathtt{U}$ is an efficient universal Turing machine with a time bound.[8] Further, let $b : \mathbb{N} \to \{0,1\}^*$ be the natural bijection between the non-negative integers and the list of all finite bit strings taken in lexicographic order, starting with the empty string, i.e. $b(0) = \varepsilon$, $b(1) = 0$, $b(2) = 1$, and $b(3) = 00$.

Let $(\alpha_n)_{n\in\mathbb{N}}$ and $(\beta_n)_{n\in\mathbb{N}}$ be sequences of non-negative integers such that the following three conditions are satisfied. First, $\alpha_n, \beta_n \leq n$ for all $n \in \mathbb{N}$. Second, $n \mapsto (\alpha_n, \beta_n)$ can be computed efficiently in n. Third, for every $a \in \mathbb{N}$, there are infinitely many distinct $b \in \mathbb{N}$ such that (a, b) is in the image of $n \mapsto (\alpha_n, \beta_n)$.[9] We define Φ by $n \mapsto (\langle b(\alpha_n)\rangle, r_{\beta_n})$.

By our assumptions on $(\alpha_n)_{n\in\mathbb{N}}$ and $(\beta_n)_{n\in\mathbb{N}}$, the map Φ we have constructed here satisfies the assumptions made of it in the proof of Theorem 4.4, namely that for every $\mathtt{T} \in \mathcal{T}$ there are infinitely many $r \in \mathcal{R}$ such that $(\mathtt{T}, r)$ is in the image of Φ.

Now that we have fixed Φ, the ℓ-sequence $(s_n)_{n\in\mathbb{N}}$ constructed in the proof of Theorem 4.4 is fixed. It now suffices to show how it can be computed in triple exponential-time.

Recall from the proof of Theorem 4.4 that s_n is the first element, in lexicographic order,

[8] Such a machine $\mathtt{U}$ and encoding $\langle\cdot\rangle$ can be found implicitly in [NW06].

[9] As a concrete example, we can take $(\alpha_n)_{n\in\mathbb{N}}$ to be the sequence obtained by concatenating each finite sequence of the form $0, 1, 2, \ldots, n$ for all $n \in \mathbb{N}$ and take $(\beta_n)_{n\in\mathbb{N}}$ to be the sequence obtained by repeating, in increasing order, every element $n \in \mathbb{N}$ exactly $n + 1$ times.

of the non-empty set $\{0,1\}^{\ell(n)} \setminus \mathcal{S}_n$ where $\mathcal{S}_n = \cup_{t=0}^{n} \mathcal{S}_{t,n}$ and

$$\mathcal{S}_{t,n} = \begin{cases} \left\{x \in \{0,1\}^{\ell(n)} : \langle x| \, \tau(t)(1^n)(\varepsilon) \, |x\rangle \geq q(n)\right\} & \text{if, on input of } 1^n, \ \tau(t) \text{ halts} \\ & \text{in at most } \rho(t)(n) \text{ steps and} \\ & \text{yields a } (0,\ell(n))\text{-circuit.} \\ \varnothing & \text{else.} \end{cases} \tag{4.14}$$

where τ and ρ are the projections on the first and second component of Φ, respectively. Thus, to compute s_n, it suffices to be able to check the membership of a string in the sets $\mathcal{S}_{t,n}$ for all $t \leq n$. We describe a Turing machine $\mathtt{T}'$ running in triple exponential-time which finds the first string, in lexicographic order, which is not in any of these sets.

At a high-level, the triple exponential run time of $\mathtt{T}'$ can be explained as follows. The first exponential is due to the need to simulate another Turing machine running in exponential-time. The second exponential is essentially due to the fact that $\mathtt{T}'$ will be simulating more and more complex Turing machines as n grows (specifically due to the fact that $n \mapsto r_n(n)$ is not exponential in n, but is double exponential). The third exponential is due to the "brute force" numerical computation to analyse the quantum circuits produced by the simulated Turing machines.

Formally, let $\mathtt{T}'$ be a Turing machine which, on input of 1^n, does the following:

1. Compute $\ell(n)$.
   (By assumption, this can be done efficiently.)

2. Initialize a string $s = 0^{\ell(n)}$ and a counter $t = 0$, both of which will be updated as the algorithm proceeds.
   (This can be done efficiently in $\ell(n)$.)

3. If $t > n$, go to step 4. Else, do the following:

   (a) Compute α_t and β_t.
   (This can be done efficiently in t, and so efficiently in n as $t \leq n$.)

   (b) Run $\mathtt{U}(b(\alpha_t), 1^n, 1^{r_{\beta_t}(n)})$ and let z denote the output.
   (By assumption, there exists a polynomial p, which we can assume to be non-decreasing, such that this takes at most $p(|b(\alpha_t)| + n + r_{\beta_t}(n))$ steps. As $\alpha_t, \beta_t \leq t \leq n$, $|b(n)| \leq n$, and $r_i \leq r_{i+1}$, this then takes at most $p(2n + r_n(n))$ steps. Note that $r_n(n) = 2^{(n+2)^n}$. Since $(n+2)^n \in \mathcal{O}(2^{n^2})$, we conclude that $r_n(n)$ and $p(2n + r_n(n))$ are at most double exponential in n.)

   (c) If the first bit of z is 0, which is to say that the Turing machine being simulated did not terminate within the desired number of steps, increment t by 1 and go

back to step 3. Else, let z' be all but the first bit of z and check that z' encodes a $(0, \ell(n))$ circuit. If it does not, increment t by 1 and go back to step 3. If it does, then do the following:
(This can be done efficiently in $\ell(n)$ and the length of z'. Since $\ell(n)$ is at at most double exponential in n and the length of z' is at most double exponential in n, as it is the result of a simulation which ran for at most $r_n(n)$ simulated steps, then this check can be executed in time double exponential in n.)

i. Let C be the $(0, \ell(n))$ circuit described by z'. Check if the inequality $\langle s| C(\varepsilon) |s\rangle < p(n) \cdot 2^{-\ell(n)}$ holds. If so, increment t by 1 and go back to step 3. If not, set $t = 0$, update s to be the next string of length $\ell(n)$ in lexicographic order, and go back to step 3.
(The time complexity of this check is dominated by the complexity of computing the value $\langle s| C(\varepsilon) |s\rangle$. Recall that C is described by z' which was produced after at most $r_n(n)$ simulated steps. Hence, $|z'| \leq r_n(n)$ and so the circuit C can have at most $r_n(n)$ gates. Thus, we can compute the value $\langle s| C(\varepsilon) |s\rangle$ in time exponential in $r_n(n)$, which is triple exponential in n as $r_n(n)$ is double exponential in n.)

4. Output s.

The fact that $\mathtt{T}'(1^n)$ indeed halts and outputs s_n follows from the proof of Theorem 4.4. The time complexity of this algorithm is dominated by step 3.c.i, with each execution of that step taking time which is at most triple exponential in n. Furthermore, this step is executed at most $(n+1)2^{\ell(n)}$ times (at most $2^{\ell(n)}$ strings verified each at most $(n+1)$ times), which is itself at most triple exponential in n. Thus, $\mathtt{T}'$ runs in at most triple exponential time on input of 1^n, which is the desired result. ∎

4.3.2 Replacing Uniformly Random Strings with Ingenerable Strings

The goal of this section is to prove Theorem 4.10 which is stated below. This theorem can be understood as stating that if a given computation succeeds with sufficiently small probability over uniformly random inputs, then applying that same computation on fixed inputs determined by an ingenerable ℓ-sequence also yields a negligible success probability, provided that ℓ is large enough.

Theorem 4.10. Let $(s_n)_{n\in\mathbb{N}}$ be a computably (respectively, exponential-time) ingenerable ℓ-sequence for any $\ell : \mathbb{N} \to \mathbb{N}$ (respectively, $\ell \in \mathcal{O}(n^k)$ for some $k \in \mathbb{N}$). Then for any uniform (respectively, exponential-time) family $(C_n)_{n\in\mathbb{N}}$ of $(\ell, 1)$-circuits, there exists a polynomial

p such that for all sufficiently large values of n (where "sufficiently large" might depend on $(C_n)_{n\in\mathbb{N}}$) we have that

$$\langle 1|\, C_n(s_n)\, |1\rangle \leq p(n) \cdot \left(2^{-\ell(n)} + \mathop{\mathbb{E}}_{x\leftarrow\{0,1\}^{\ell(n)}} \langle 1|\, C_n(x)\, |1\rangle\right). \tag{4.15}$$

In particular, if $\ell \in \omega(\log)$ and $n \mapsto \mathbb{E}_{x\leftarrow\{0,1\}^{\ell(n)}} \langle 1|\, C_n(x)\, |1\rangle$ is negligible, then so is the map $n \mapsto \langle 1|\, C_n(s_n)\, |1\rangle$.

The proof of Theorem 4.10 is simply an application of the following lemma which considers a naive and non-efficient way to transform a circuit C for a decision problem into a circuit $\tilde{C}$ for a search problem. It then gives a lower bound on the probability that $\tilde{C}$ outputs any given string.

Lemma 4.11. Let $n \in \mathbb{N}$ and let C be an $(n,1)$-circuit. Now, let $\tilde{C}$ be a $(0,n)$-circuit which implements the following algorithm:

1. Initialize an empty set $\mathcal{S}$.
2. Iterating over all strings $x \in \{0,1\}^n$, do the following: Run $C(x)$ and measure the output in the computational basis. If the result is 1, add x to $\mathcal{S}$. Else, do nothing.
3. Sample uniformly at random a string x from $\mathcal{S}$ and output it. If $\mathcal{S}$ is empty, sample $\tilde{x}$ uniformly at random from $\{0,1\}^n$.

Then, for any $s \in \{0,1\}^n$, we have that

$$\frac{\langle 1|\, C(s)\, |1\rangle}{1 + \sum_{x\in\{0,1\}^n\setminus\{s\}} \langle 1|\, C(x)\, |1\rangle} \leq \langle s|\, \tilde{C}(\varepsilon)\, |s\rangle\,. \tag{4.16}$$

Before proving Lemma 4.11, we highlight a technical fact. If a subset $\mathcal{S}$ of a finite set $\mathcal{X}$ is constructed by iterating over all elements $x \in \mathcal{X}$ and including this element in $\mathcal{S}$ with probability p_x, independently of all other choices, then the expected size of $\mathcal{S}$ is $\sum_{x\in\mathcal{X}} p_x$.

Fact 4.12. Let $\mathcal{X}$ be a finite set and, for every $x \in \mathcal{X}$, let $p_x \in [0,1]$ be a real number. Let $\mathcal{S}$ be a random variable distributed on $\mathcal{P}(\mathcal{X})$, the power set of $\mathcal{X}$ such that for all $\mathcal{Y} \in \mathcal{P}(\mathcal{X})$

$$\Pr[\mathcal{S} = \mathcal{Y}] = \left(\prod_{x\in\mathcal{Y}} p_x\right) \cdot \left(\prod_{x\in\mathcal{Y}\setminus\mathcal{S}} 1 - p_x\right). \tag{4.17}$$

Then, $\mathbb{E}\,|\mathcal{S}| = \sum_{x\in\mathcal{X}} p_x$.

Proof: For any $\mathcal{Y} \subseteq \mathcal{X}$, let $\delta_{\mathcal{Y}} : \mathcal{X} \to \{0,1\}$ be the characteristic function for the set $\mathcal{Y}$. We can then compute $\mathbb{E}\,|\mathcal{S}|$ as

$$\begin{aligned}\sum_{\mathcal{Y}\in\mathcal{P}(\mathcal{X})} |\mathcal{Y}| \cdot \Pr\left[\mathcal{S} = \mathcal{Y}\right] &= \sum_{\mathcal{Y}\in\mathcal{P}(\mathcal{X})} \sum_{x\in\mathcal{X}} \delta_{\mathcal{Y}}(x) \cdot \Pr\left[\mathcal{S} = \mathcal{Y}\right] \\ &= \sum_{x\in\mathcal{X}} \sum_{\mathcal{Y}\in\mathcal{P}(\mathcal{X})} \delta_{\mathcal{Y}}(x) \cdot \Pr\left[\mathcal{S} = \mathcal{Y}\right]\end{aligned} \tag{4.18}$$

which is precisely $\sum_{x\in\mathcal{X}} \Pr[x \in \mathcal{S}] = \sum_{x\in\mathcal{X}} p_x$, the desired result. ∎

We can now prove Lemma 4.11.

Proof: Let $\mathcal{S}$ be the random variable distributed on $\mathcal{P}\left(\{0,1\}^n\right)$ which models the contents of the set maintained by the circuit $\tilde{C}$ once it terminates step 2. Note that

$$\Pr\left[\mathcal{S} = \mathcal{X}\right] = \left(\prod_{x\in\mathcal{X}} \langle 1|\, C(x)\,|1\rangle\right)\left(\prod_{x\in\{0,1\}^n\setminus\mathcal{X}} 1 - \langle 1|\, C_x(x)\,|1\rangle\right). \tag{4.19}$$

We see that

$$\langle s|\, \tilde{C}(\varepsilon)\,|s\rangle \geq \sum_{\substack{\mathcal{X}\in\mathcal{P}(\{0,1\}^n)\\ s\in\mathcal{X}}} \frac{1}{|\mathcal{X}|} \cdot \Pr\left[\mathcal{S} = \mathcal{X}\right] \tag{4.20}$$

where the inequality is obtained by neglecting the case where $\mathcal{S}$ is empty.

Next, let $\mathcal{S}'$ be the random variable distributed on $\mathcal{P}\left(\{0,1\}^n \setminus \{s\}\right)$ such that

$$\Pr\left[\mathcal{S}' = \mathcal{X}\right] = \left(\prod_{x\in\mathcal{X}} \langle 1|\, C(x)\,|1\rangle\right)\left(\prod_{x\in(\{0,1\}^n\setminus\{s\})\setminus\mathcal{X}} 1 - \langle 1|\, C_x(x)\,|1\rangle\right) \tag{4.21}$$

and note that if $s \in \mathcal{X}$ then $\Pr[\mathcal{S} = \mathcal{X}] = \langle 1| C(s) |1\rangle \cdot \Pr[\mathcal{S}' = \mathcal{X} \setminus \{s\}]$. We then have that

$$\begin{aligned}
\sum_{\substack{\mathcal{X}\in\mathcal{P}(\{0,1\}^n)\\ s\in\mathcal{X}}} \frac{1}{|\mathcal{X}|} \cdot \Pr\left[\mathcal{S} = \mathcal{X}\right] &= \sum_{\substack{\mathcal{X}\in\mathcal{P}(\{0,1\}^n)\\ s\in\mathcal{X}}} \frac{1}{1+|\mathcal{X}\setminus\{s\}|} \cdot \Pr\left[\mathcal{S} = \mathcal{X}\right] \\
&= \sum_{\substack{\mathcal{X}\in\mathcal{P}(\{0,1\}^n)\\ s\in\mathcal{X}}} \frac{\langle 1| C(s) |1\rangle}{1+|\mathcal{X}\setminus\{s\}|} \cdot \Pr\left[\mathcal{S}' = \mathcal{X}\setminus\{s\}\right] \\
&= \langle 1| C(s) |1\rangle \sum_{\mathcal{X}'\in\mathcal{P}(\{0,1\}^n\setminus\{s\})} \frac{1}{1+|\mathcal{X}'|} \cdot \Pr\left[\mathcal{S}' = \mathcal{X}'\right] \qquad (4.22) \\
&= \langle 1| C(s) |1\rangle \cdot \mathbb{E}\, \frac{1}{1+|\mathcal{S}'|} \\
&\geq \langle 1| C(s) |1\rangle \cdot \frac{1}{1+\mathbb{E}\,|\mathcal{S}'|} \\
&= \langle 1| C(s) |1\rangle \cdot \frac{1}{1+\sum_{x\in\{0,1\}^n\setminus\{s\}} \langle 1| C(x) |1\rangle}
\end{aligned}$$

where the inequality is obtained by Jensen's inequality and the expectation is evaluated by Fact 4.12. The third equality can be obtained by noting that there is a canonical bijection from the elements $\mathcal{X} \in \mathcal{P}(\{0,1\}^n)$ which contain s to $\mathcal{P}(\{0,1\}^n \setminus \{s\})$: $\mathcal{X} \mapsto \mathcal{X} \setminus \{s\}$. Combining this with Equation 4.20 yields the desired result. ∎

We can now prove Theorem 4.10.

Proof: Let $(\tilde{C}_n)_{n\in\mathbb{N}}$ be a $(0,\ell)$-circuit family such that $\tilde{C}_n$ implements the algorithm described in Lemma 4.11 with respect to C_n for all $n \in \mathbb{N}$.

Note that if $(C_n)_{n\in\mathbb{N}}$ is an exponential-time family and $\ell \in \mathcal{O}(n^k)$, then $(\tilde{C}_n)_{n\in\mathbb{N}}$ is an exponential-time circuit family. Otherwise, if $(C_n)_{n\in\mathbb{N}}$ is a uniform family then so is $(\tilde{C}_n)_{n\in\mathbb{N}}$.

Lemma 4.11 then yields that

$$\langle 1| C_n(s_n) |1\rangle \leq \langle s_n| \tilde{C}_n(\varepsilon) |s_n\rangle \left(1 + \sum_{x\in\{0,1\}^{\ell(n)}\setminus\{s_n\}} \langle 1| C_n(x) |1\rangle\right) \qquad (4.23)$$

for all $n \in \mathbb{N}$. As $(s_n)_{n\in\mathbb{N}}$ is a computably (respectively, exponential-time) ingenerable ℓ-sequence, there exists a polynomial p such that, for all sufficiently large values of n, we

have that $\langle s_n| \tilde{C}_n(\varepsilon) |s_n\rangle \leq p(n) \cdot 2^{-\ell(n)}$. Thus,

$$\langle 1| C_n(s_n) |1\rangle \leq \frac{p(n)}{2^{\ell(n)}} \left(1 + \sum_{x \in \{0,1\}^{\ell(n)} \setminus \{s_n\}} \langle 1| C_n(x) |1\rangle\right) \tag{4.24}$$

for all sufficiently large values of n. Adding the $x = s_n$ term to the sum and distributing the $2^{-\ell(n)}$ factor yields the first desired result.

Furthermore, if $\ell \in \omega(\log)$, then $2^{-\ell}$ is negligible. If $\mathbb{E}_{x \leftarrow \{0,1\}^{\ell(n)}} \langle x| C(x) |x\rangle$ is also negligible, then the upper bound is a polynomial times the sum of two negligible functions and thus is itself negligible. ∎

4.3.3 Derandomizing a Cloning Task via Ingenerable Sequences

In this section, we construct a sequence of fixed states $(|\psi_n\rangle\langle\psi_n|)_{n\in\mathbb{N}}$ which cannot be cloned by any uniform family of quantum circuits. This will be a simple demonstration of the ideas we will be using in our constructions of uncloneable advice in Section 4.5.

The following lemma was first shown in [MVW13] to formalize the proof of security of Wiesner's quantum money scheme [Wie83]. It can be interpreted as a bound on how well a party can clone a single Wiesner state $|x^\theta\rangle$ selected uniformly at random.

Lemma 4.13. Let $n \in \mathbb{N}$ and let C be an $(n, 2n)$-circuit. Then,

$$\mathop{\mathbb{E}}_{\substack{x \leftarrow \{0,1\}^n \\ \theta \leftarrow \{0,1\}^n}} \langle x^\theta| \langle x^\theta| C\left(|x^\theta\rangle\langle x^\theta|\right) |x^\theta\rangle |x^\theta\rangle \leq \left(\frac{3}{4}\right)^n. \tag{4.25}$$

Note that the lemma holds for all quantum channels, but we restrict ourselves to circuits to maintain consistency with the rest of this work.

We can now "derandomize" Lemma 4.13 by replacing the uniformly random Wiesner state $|x^\theta\rangle$ with a fixed one chosen according to an ingenerable sequence. It then suffices to invoke Theorem 4.10 to obtain the desired result.

Theorem 4.14. Let $(s_n)_{n\in\mathbb{N}}$ be a computably ingenerable $2n$-sequence and parse each s_n as a pair of bit strings of length n, *i.e.*: $s_n = (x_n, \theta_n)$ for $x_n, \theta_n \in \{0,1\}^n$ for all $n \in \mathbb{N}$. Then, for any uniform family of $(n, 2n)$-circuits $(C_n)_{n\in\mathbb{N}}$, we have that

$$n \mapsto \langle x_n^{\theta_n}| \langle x_n^{\theta_n}| C_n\left(|x_n^{\theta_n}\rangle\langle x_n^{\theta_n}|\right) |x_n^{\theta_n}\rangle |x_n^{\theta_n}\rangle \tag{4.26}$$

is a negligible function.

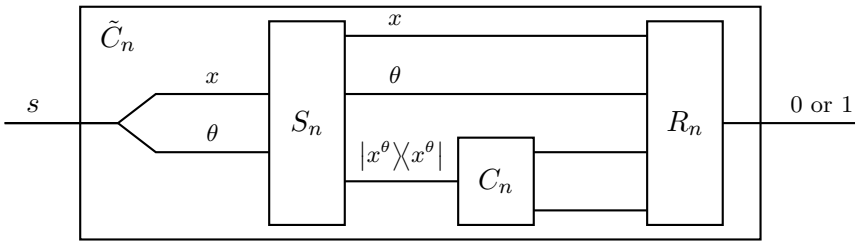


Figure 4.2: The $\tilde{C}_n$ circuits of the proof of Theorem 4.14.

Proof: Conceptually, it suffices to apply Theorem 4.10 to Lemma 4.13. However, a few technical details are needed to frame Lemma 4.13 in a way where Theorem 4.10 is applicable.

Let $(S_n)_{n\in\mathbb{N}}$ be a uniform family of $(2n, 3n)$-circuits such that on input of $x \otimes \theta$ they output the state $x \otimes \theta \otimes \left|x^\theta\right\rangle\!\left\langle x^\theta\right|$ for any $x, \theta \in \{0,1\}^n$. Let $(R_n)_{n\in\mathbb{N}}$ be a uniform family of $(4n, 1)$-circuits which, on input of $x \otimes \theta \otimes \rho$ for any strings $x, \theta \in \{0,1\}^n$ and any state ρ on $2n$ qubits, applies the unitary $H^\theta \otimes H^\theta$ to ρ, measures the resulting qubits in the computational basis, and outputs 1 if two copies of x are obtained and outputs 0 otherwise.

Now, consider the uniform family of $(2n, 1)$-circuits $(\tilde{C}_n)_{n\in\mathbb{N}}$ where $\tilde{C}_n$ is obtained by composing the S_n, C_n, and R_n circuits, in that order and where C_n acts on the last n qubits produced by S_n. This is illustrated in Figure 4.2. At the level of channels, we see that

$$\tilde{C}_n = R_n \circ (\mathrm{Id}_n \otimes \mathrm{Id}_n \otimes C_n) \circ S_n. \tag{4.27}$$

In particular, note that

$$\langle 1|\, \tilde{C}_n(x \otimes \theta)\, |1\rangle = \langle x^\theta|\, \langle x^\theta|\, C\left(|x^\theta\rangle\langle x^\theta|\right) |x^\theta\rangle\, |x^\theta\rangle \tag{4.28}$$

for all $x, \theta \in \{0,1\}^n$. Thus, by Lemma 4.13,

$$n \mapsto \mathop{\mathbb{E}}_{\substack{x\leftarrow\{0,1\}^n \\ \theta\leftarrow\{0,1\}^n}} \langle 1|\, \tilde{C}_n(x \otimes \theta)\, |1\rangle \tag{4.29}$$

is negligible. Hence, by Theorem 4.10 and the fact that $(s_n)_{n\in\mathbb{N}}$ is computably ingenerable, the map $n \mapsto \langle 1|\, \tilde{C}_n(x_n \otimes \theta_n)\, |1\rangle$ is also negligible. By Equation 4.28, this is the desired result. ∎

4.4 State and Cloning Complexity Classes

In this section, we define a notion of cloning complexity class by generalizing the notion of a state complexity class, as studied by Metger, Rosenthal, and Yuen [RY22, MY23]. Roughly speaking, these works define and study the complexity of generating a given sequence $(\rho_n)_{n\in\mathbb{N}}$ of quantum states. The main result of these works is showing that **stateQIP** = **statePSPACE**, which is to say that the sequences of states which can be efficiently generated with the help of a possibly malicious prover are precisely those which can be generated by polynomial-space quantum circuits. This is a state synthesis analogue to the celebrated result of **QIP** = **PSPACE** [JJUW11].

From our perspective, these prior works concern the difficulty of creating one copy of ρ_n starting from no copies; we propose that it is natural to also study the difficulty of creating two copies from one, or more generally, b copies starting from a copies.[10]

Definition 4.15. Let $\mathcal{C}$ be a class of circuit families, $\delta : \mathbb{N} \to \mathbb{R}$ be a function, and $a, b \in \mathbb{N}$ be two non-negative integers. A sequence of states $(\rho_n)_{n\in\mathbb{N}}$ is in the state complexity class $\mathbf{state}_{\delta}^{a\to b}(\mathcal{C})$ if there exists a circuit family $C \in \mathcal{C}$ and an $n' \in \mathbb{N}$ such that

$$n \geq n' \implies \frac{1}{2}\left\|\rho_n^{\otimes b} - C_n(\rho_n^{\otimes a})\right\|_1 \leq \delta(n). \tag{4.30}$$

We also define

$$\mathbf{state}^{a\to b}(\mathcal{C}) = \bigcap_{k\in\mathbb{N}} \mathbf{state}_{n^{-k}}^{a\to b}(\mathcal{C}). \tag{4.31}$$

If $a = 0$ and $b = 1$, we may omit the $a \to b$ superscript from all the above notation.

We note that in full generality, we could also have a and b in the definition be maps from $\mathbb{N}$ to $\mathbb{N}$, but this will not be necessary for our observations.

Naturally occurring classes of circuit families which may be of interest include the class of uniform circuit families, which we denote $\mathcal{C}_{\mathrm{UNIF}}$, and the class of polynomial-time circuit families, which we denote $\mathcal{C}_{\mathrm{POLY}}$. Letting $\mathcal{C}_{\mathrm{PSPACE}}$ denote the class of "space-uniform circuits" as in [MY23, Definition 2.2] and **statePSPACE** be as in [MY23, Definition 2.4], we find, as expected, that $\mathbf{state}(\mathcal{C}_{\mathrm{PSPACE}}) = \mathbf{statePSPACE}$.[11] We note that finding a class of circuit families, tentatively denoted $\mathcal{C}_{\mathrm{QIP}}$, such that $\mathbf{state}(\mathcal{C}_{\mathrm{QIP}}) = \mathbf{stateQIP}$, where the latter is as given in [MY23, Definition 5.1], appears to be a bit more involved

[10] Recent work has also studied the quantum *Kolmogorov* complexity of cloning a quantum state [LFM+24]. The authors obtain results showing that, from this perspective, cloning a quantum state is almost always essentially as hard as creating it.

[11] Note that [RY22, MY23] also require that a sequence $(\rho_n)_{n\in\mathbb{N}}$ in **statePSPACE** also satisfies the constraint that each ρ_n is on precisely n qubits. However, they state that this is merely for convenience. We drop this requirement from our definition.

and non-trivial. In short, it is unclear how to encode the required soundness property of **stateQIP** into a class of circuit families. Nonetheless, we believe Definition 4.15 to be a natural generalization of state complexity classes in a wide range of interesting settings.

One interesting result of formalizing cloning complexity classes is that it allows us to make more precise statements about the relations between the complexity of generating and of cloning quantum states. Specifically, we can now state and prove the three lemmas which were sketched in Figure 4.1 from Section 4.1.1.2.

The first lemma formalizes the idea that, for essentially all classes of circuit families, cloning a state cannot be more difficult than generating it. For technical reasons, this lemmas does not apply to certain pathological classes of circuit families (*e.g.*: those without the ability to apply the identity channel), or those representing extremely limited computational power (*e.g.*: those with a hard bound on the number of qubits they can process). The idea is that if one is able to generate a sequence of states, then one way to clone the same sequence is to ignore the input and simply generate another copy.

Lemma 4.16. Let $\mathcal{C}$ be a class of circuit families satisfying the following criteria:

- The class $\mathcal{C}$ can implement the identity on qubits it could generate. More precisely, if $\mathcal{C}$ includes a $(0,\ell)$-circuit family C for some $\ell : \mathbb{N} \to \mathbb{N}$, then $\mathcal{C}$ also includes an (ℓ,ℓ)-circuit family C^{id} where every circuit in this family implements the identity channel.
- The class $\mathcal{C}$ is closed under tensor products. More precisely, given two circuit families $(C_n)_{n\in\mathbb{N}}$ and $(C'_n)_{n\in\mathbb{N}}$ in $\mathcal{C}$, then the family $(C_n \otimes C'_n)_{n\in\mathbb{N}}$ is in $\mathcal{C}$.

Then, for all $\delta : \mathbb{N} \to \mathbb{R}$ and $a, b \in \mathbb{N}$ we have that

$$\mathbf{state}_\delta^{0\to 1}(\mathcal{C}) \subseteq \mathbf{state}_\delta^{1\to 2}(\mathcal{C}). \tag{4.32}$$

Proof: Let $(\rho_n)_{n\in\mathbb{N}} \in \mathbf{state}_\delta^{0\to 1}(\mathcal{C})$. Let $C \in \mathcal{C}$ and $n' \in \mathbb{N}$ be such that

$$n \geq n' \implies \frac{1}{2}\|\rho_n - C_n(\varepsilon)\|_1 \leq \delta(n). \tag{4.33}$$

Consider now the family $C' = (C_n^{\mathrm{id}} \otimes C_n)_{n\in\mathbb{N}}$ which, by our assumptions, is in $\mathcal{C}$. For all $n \geq n'$, we have that

$$\frac{1}{2}\left\|\rho_n^{\otimes 2} - C'_n(\rho_n)\right\|_1 = \frac{1}{2}\|\rho_n \otimes \rho_n - \rho_n \otimes C_n(\varepsilon)\|_1 \leq \frac{1}{2}\|\rho_n - C_n(\varepsilon)\|_1 \leq \delta(n) \tag{4.34}$$

and so $(\rho_n)_{n\in\mathbb{N}} \in \mathbf{state}_\delta^{1\to 2}(\mathcal{C})$. ∎

The second lemma formalizes the idea that there exists sequences of states which cannot be generated but which can be cloned, even with less computational power. It suffices here to consider sequences of classical information which cannot be consistently produced with high probability. Weakly computably ingenerable sequences satisfy these properties.

Lemma 4.17. Let $(s_n)_{n\in\mathbb{N}}$ be a weakly computably ingenerable n-sequence. Then, the sequence of states $(|s_n\rangle\langle s_n|)_{n\in\mathbb{N}}$ is in $\mathbf{state}^{1\to 2}(\mathcal{C}_{\mathrm{POLY}})$ but not in $\mathbf{state}(\mathcal{C}_{\mathrm{UNIF}})$.

Proof: By definition, $(|s_n\rangle\langle s_n|)_{n\in\mathbb{N}}$ is not in $\mathbf{state}(\mathcal{C}_{\mathrm{UNIF}})$ since every uniform circuit family will produce the string s_n with negligible probability for infinitely many values of n.

On the other hand, let $(C_n)_{n\in\mathbb{N}}$ be an $(n, 2n)$-circuit family where C_n simply appends n qubits in the $|0\rangle$ state and then applies a CNOT gate on the $n+k$ qubit conditioned on the k qubit for all $k \in \{1, ..., n\}$. We see that $C_n(|s_n\rangle\langle s_n|) = |s_n\rangle\langle s_n| \otimes |s_n\rangle\langle s_n|$. Since we have that $(C_n)_{n\in\mathbb{N}} \in \mathcal{C}_{\mathrm{POLY}}$, it holds that $(|s_n\rangle\langle s_n|)_{n\in\mathbb{N}} \in \mathbf{state}^{1\to 2}(\mathcal{C}_{\mathrm{POLY}})$. ∎

The third and final lemma states that there are sequences of states which cannot be cloned by uniform circuit families. It is a direct corollary ofTheorem 4.14 and can be interpreted as a cloning complexity analogue to the existence of uncomputable functions.

Lemma 4.18. Let $(s_n)_{n\in\mathbb{N}}$ be a computably ingenerable $2n$-sequence where we parse each s_n as (x_n, θ_n) for two strings $x_n, \theta_n \in \{0,1\}^n$. Then, $(\,|x_n^{\theta_n}\rangle\langle x_n^{\theta_n}|)_{n\in\mathbb{N}} \notin \mathbf{state}^{1\to 2}(\mathcal{C}_{\mathrm{UNIF}})$. By Lemma 4.16, this also implies that the sequence is not in $\mathbf{state}(\mathcal{C}_{\mathrm{UNIF}})$.

Note that the corollary is a strictly weaker statement than the theorem. Indeed, the theorem demonstrates that it is impossible to clone with *non-negligible* fidelity, via a uniform family of circuits, the Wiesner states described by a computably ingenerable sequence. The corollary only states that it is impossible to clone these states with an *overwhelming* fidelity. We emphasize that our results in Section 4.5 concerning uncloneable quantum advice will require the stronger guarantee of the form given by the theorem.

4.5 Uncloneable Advice

The main results of this section are to give a definition for the complexity class of problems which can be solved by polynomial-time quantum computations with negligible errors in the presence of advice that is uncloneable and to give examples of problems in this class. We denote this complexity class **neglQP**/upoly.

As we briefly discussed in Section 4.1.1.1, we do not denote this class **BQP**/upoly; this is because we lack a generic error reduction technique which can reduce bounded errors to negligible errors, all the while maintaining the uncloneability property of the advice states. In other words, it is possible that **neglQP**/upoly $\subseteq$ **BQP**/upoly is a strict containment.

Section overview. We begin by reviewing the basic notions of quantum copy-protection in Section 4.5.1. In Section 4.5.2, we formally define the notion of uncloneable advice and give some basic remarks on this definition. We also emphasize some parallels between quantum copy-protection and uncloneable advice. In Section 4.5.3, we describe a promise problem which unconditionally admits uncloneable advice. In Section 4.5.4, we describe a language with uncloneable advice, assuming the feasibility of copy-protecting any one sequence of maps satisfying fairly mild assumptions. In Section 4.5.5, we discuss one possible instantiation of the construction presented in Section 4.5.4.

4.5.1 Quantum Copy-Protection

As discussed in Section 4.1, quantum copy-protection is broadly the task of encoding a given function f as a quantum state ρ_f such that the following two conditions are satisfied:

- Correctness: A party with access to the quantum state ρ_f can correctly evaluate f on any input by interacting with this state in a prescribed manner.
- Security: A party with access to the quantum state ρ_f cannot create and share a bipartite quantum state with two other separated parties such that both of these parties could correctly compute f using any means available to them.

While correctness is required to hold for all inputs, security is defined with respect to a sequence of random variables $D = (D_n)_{n\in\mathbb{N}}$ jointly distributed on all functions considered by the copy-protection scheme and pairs of inputs to these functions. Security is attained if no efficient attacker can split the program state for a function f, allowing both subsequent parties to evaluate $f(x_B)$ and $f(x_C)$ with more than a negligible advantage when (f, x_B, x_C) is sampled from D.

We formalize the syntax and correctness guarantee of a copy-protection scheme in Definition 4.19 below. Security will then be formalized in Definition 4.24. Up to technical details of presentation, our definition of the syntax and correctness of a copy-protection scheme coincides with the definition given in [CLLZ21].[12]

Definition 4.19. Consider a sequence of maps

$$f = \left(f_n : \{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{c(n)}\right)_{n\in\mathbb{N}} \tag{4.35}$$

whose domains and codomains are parameterized by maps $\kappa, d, c : \mathbb{N} \to \mathbb{N}$. A *copy-protection scheme* for f with program length $q : \mathbb{N} \to \mathbb{N}$ is a pair (G, E) of efficient (κ, q)- and $(q+d, c)$-circuit families, respectively.

[12]The definitions given in [CLLZ21] states that the map f should be pseudorandom function. However, they are applicable to any such family of maps, not only pseudorandom functions.

A copy-protection scheme is *correct for* f if there exists a negligible function η such that

$$\langle f_n(k,x)|\, E_n(G_n(k)\otimes x)\,|f_n(k,x)\rangle \geq 1-\eta(n) \tag{4.36}$$

for all $n\in\mathbb{N}$, all $k\in\{0,1\}^{\kappa(n)}$, and all $x\in\{0,1\}^{c(n)}$.

Note that the existence of a copy-protection scheme for a sequence of maps f as defined above implies that the maps κ, d, and c are polynomially bounded and efficiently computable, as otherwise there would not exist efficient (κ,q)- and $(q+d,c)$-circuit families.

Remark 4.20. The above definition can also be easily adapted to sequences of maps $(f_n)_{n\in\mathbb{N}}$ where, for all $n\in\mathbb{N}$, the domain of f_n is a subset $S_n\subseteq\{0,1\}^{\kappa(n)}\times\{0,1\}^{d(n)}$. To do so, it suffices to only require the correctness condition, Equation 4.36, to hold for pairs $(k,x)\in S_n$.

Remark 4.21. It is possible for a copy-protection scheme to be correct for two different sequences of maps. Indeed, assume $f=(f_n)_{n\in\mathbb{N}}$ and $f'=(f'_n)_{n\in\mathbb{N}}$ are sequences of maps such that $f_n\neq f'_n$ for a non-zero but finite number of values of n. Then, it is straightforward to show that a copy-protection scheme (G,E) is correct for the sequence f if and only if it is correct for the sequence f'. In short, this is due to the fact that correctness is an asymptotic property and that $f_n=f'_n$ for all sufficiently large values of n.

However, this is the maximal possible discrepancy between f and f'. If (G,E) is correct for f and f', then there must exist a $\tilde{n}\in\mathbb{N}$ such that $n\geq\tilde{n}\implies f_n=f'_n$. Indeed, by the assumed correctness of the scheme for f and f', the triangle inequality for the trace distance, the Fuchs-van de Graaf inequalities, and basic properties of negligible functions, there exists a negligible function η such that

$$\begin{aligned}
&\frac{1}{2}\left\|f_n(k,x)-f'_n(k,x)\right\|_1\\
&\leq \frac{1}{2}\|f_n(k,x)-E_n(G_n(k)\otimes x)\|_1+\frac{1}{2}\left\|E_n(G_n(k)\otimes x)-f'_n(k,x)\right\|_1\\
&\leq \sqrt{1-\langle f_n(k,x)|\, E_n(G_n(k)\otimes x)\,|f_n(k,x)\rangle}+\sqrt{1-\langle f'_n(k,x)|\, E_n(G_n(k)\otimes x)\,|f'_n(k,x)\rangle}\\
&\leq \eta(n)
\end{aligned} \tag{4.37}$$

for all $n\in\mathbb{N}$, all $k\in\{0,1\}^{\kappa(n)}$, and all $x\in\{0,1\}^{d(n)}$. It follows that $f_n=f'_n$ for all sufficiently large n as $\eta(n)<1\implies f_n(k,x)=f'_n(k,x)$.

An attack against a copy-protection scheme is a triplet of efficient circuit families (A,B,C). We will sometimes refer to the first, A, as the *splitting adversary* and the other two as the *guessing adversaries.* This reflects their respective tasks in the game, described below, used to define and quantify the notion of security for a copy-protection scheme.

The Copy-Protection Game Let $f = (f_n)_{n\in\mathbb{N}}$ be a sequence of maps as in Definition 4.19, (G, E) be a copy-protection scheme for f, and $D = (D_n)_{n\in\mathbb{N}}$ be a sequence of random variables where each D_n is distributed on $\{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \times \{0,1\}^{d(n)}$. The copy-protection security game, for a given $n \in \mathbb{N}$ in addition to the parameters described above, is played by a Referee against collaborating Alice, Bob, and Charlie — collectively known as the adversaries — as follows:

1. The Referee samples a triplet $(k, x_B, x_C) \leftarrow D_n$.
2. The Referee prepares the state $\rho = E_n(k)$ and gives it to Alice.
3. Alice prepares a bipartite quantum state and gives one part to Bob and the other to Charlie. This is the only communication between Alice, Bob, and Charlie that occurs during the game.
4. The Referee gives x_B to Bob and x_C to Charlie. Note that Bob does not receive x_C and Charlie does not receive x_B.
5. Bob, with what they have received from the Referee and Alice, outputs a string y_B. Similarly, Charlie outputs a string y_C.
6. Alice, Bob, and Charlie win if and only if $f_n(k, x_B) = y_B$ *and* $f_n(k, x_C) = y_C$.

Formally, we model Alice, Bob, and Charlie as efficient circuit families $A = (A_n)_{n\in\mathbb{N}}$, $B = (B_n)_{n\in\mathbb{N}}$, and $C = (C_n)_{n\in\mathbb{N}}$ respectively. Note that, at this point, we do not impose any other type of computational restraints on the Referee. In particular, we do not require the random variables $D = (D_n)_{n\in\mathbb{N}}$ to admit an efficient sampling procedure. We formalize this game and the winning probability of the adversaries in the next definition.

Definition 4.22. Let (G, E) be a copy-protection scheme with program lengths $q : \mathbb{N} \to \mathbb{N}$ for a sequence of maps $f = (f_n)_{n\in\mathbb{N}}$ as given in Definition 4.19. An attack against this scheme is a triplet (A, B, C) of efficient $(q, q_B + q_C)$-, $(d + q_B, c)$-, and $(q_C + d, c)$- circuit families, respectively, for two maps $q_B, q_C : \mathbb{N} \to \mathbb{N}$.

For any $n \in \mathbb{N}$ and triplet $(k, x_B, x_C) \in \{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \times \{0,1\}^{d(n)}$, we define the state

$$\rho^{(A,B,C)}_{(G,E),n}(k, x_B, x_C) = (B_n \otimes C_n)\left(x_B \otimes (A_n \circ G_n)(k) \otimes x_C\right) \tag{4.38}$$

which models the joint output of the B and C circuits.

Let $D = (D_n)_{n\in\mathbb{N}}$ be a sequence of random variables where, for each $n \in \mathbb{N}$, D_n is distributed over the set $\{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \times \{0,1\}^{d(n)}$ and let $p_k^{x_B,x_C} = \Pr[D_n = (k, x_B, x_C)]$.

For any attack (A, B, C) against (G, E), we define its winning probability with respect to f and D, denoted by $w^{(A,B,C)}_{(G,E),f,D} : \mathbb{N} \to \mathbb{R}$, as the function

$$n \mapsto \sum_{(k,x_B,x_C)\in S'_n} p_k^{x_B,x_C} \langle f_n(k,x_B), f_n(k,x_C)| \rho^{(A,B,C)}_{(G,E),n}(k,x_B,x_C) |f_n(k,x_B), f_n(k,x_C)\rangle \tag{4.39}$$

where $S'_n \subseteq \{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \times \{0,1\}^{d(n)}$ is precisely the set of all triplets (k, x_B, x_C) where both $f_n(k, x_B)$ and $f_n(k, x_C)$ are defined.

If every map f_n in f is defined on the entire set $\{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)}$, then the summation over S'_n and the $\Pr[D_n = (k, x_B, x_C)]$ factor in Equation 4.39 is simply an alternative expression for the expectation over sampling (k, x_B, x_C) from D_n. Such an expectation is typically how the winning probability of the adversaries is presented in the literature. However, our definition is also applicable to cases where some maps f_n are defined only on a strict subset of $\{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)}$ and where the support of D_n is a strict superset of S'_n. In these cases, the expectation would not be well defined.

It may seem unnatural to consider $w^{(A,B,C)}_{(G,E),f,D}$ for sequences f and D where D_n occasionally yields pairs (k, x) outside of the domain of f_n, but our generalization allows us to easily make sense of the winning probability of the adversaries when $f_n : \varnothing \to \{0,1\}^{c(n)}$ is the empty map. Note that it is impossible to even define a random variable distributed over the domain of f_n or, more precisely, over $S'_n = \varnothing$, in this case. Note that we can interpret appearances of the empty map in a sequence $(f_n)_{n\in\mathbb{N}}$ as corresponding to cases where the map is undefined.[13] This will be useful when we later relate the notions of copy-protection and uncloneable advice.

In general, we will only be interested in sequences f and D where the support of D_n is a subset of S'_n, unless f_n is the empty map.

Remark 4.23. Let $n \in \mathbb{N}$ be such that $f_n : \varnothing \to \{0,1\}^{c(n)}$ is the empty map, i.e.: $S'_n = \varnothing$. Then, adopting the convention that a sum over an empty set is 0, we have that $\omega^{(A,B,C)}_{(G,E),f,D}(n) = 0$.

The fact that $\omega^{(A,B,C)}_{(G,E),f,D}(n) = 0$ whenever f_n is the empty map, meaning that it is "in practice undefined", implies that an adversary (A, B, C) breaking the potential security of

[13] While it may be tempting to simply remove all instances of empty maps in a sequence (f_n) and then re-index it, yielding a new sequence (g_n), this can can have a material implications to the study of copy-protection for these maps as it impacts the scaling of the computational power permitted to schemes and adversaries. Indeed, suppose that the map f_n is defined if and only if $n = 2^k$ for some $k \in \mathbb{N}$. Under the proposed re-indexing scheme, $g_n = f_{2^n}$ for all $n \in \mathbb{N}$. In particular, if we subject the circuit families of an efficient copy-protection scheme (G, E) for $(f_n)_{n\in\mathbb{N}}$ to the same re-indexing, which would be the naive way to try to obtain a scheme for $(g_n = f_{2^n})_{n\in\mathbb{N}}$, they may become exponential-time circuit families.

a copy-protection scheme must do so for values of n where f_n is "in practice defined". This will be made clearer with the following definition. We see this as a desirable property of our definition.

Following a standard cryptographic paradigm, we now wish to define the security for a copy-protection scheme as the property that all efficient adversaries have at most a negligible advantage over some trivially attainable success probability. The issue here is to identify what precisely is the trivial success probability in the above described game.

There are a few attacks against copy-protection schemes which cannot reasonably be prevented. One such attack, consists of having the splitting adversary A_n give the complete and unmodified program state $G_n(k)$ to the guessing party B_n and giving nothing to C_n. When the adversary B_n receives their challenge x_B, they can then honestly evaluate the program state $G_n(k)$ on x_B to obtain, with overwhelming probability assuming that the scheme (G, E) is correct for f, the proper answer. The C_n adversary, for their part, will simply output a uniformly random string sampled from $\{0,1\}^{c(n)}$, the codomain of the maps under consideration. It is easy to see that the winning probability of this trivial attack is $n \mapsto 2^{-c(n)} - \eta(n)$ for some negligible function η accounting for the possibility of an incorrect evaluation by B. Thus, we take $2^{-c(n)}$ as our trivial success probability.

Definition 4.24. Let $f = (f_n)_{n\in\mathbb{N}}$ be a sequence of maps as given in Definition 4.19, (G, E) be a copy-protection scheme for f, and $D = (D_n)_{n\in\mathbb{N}}$ be a sequence of random variables as given in Definition 4.22. The copy-protection scheme (G, E) is secure with respect to f and D if for every attack (A, B, C) against it, the following function is negligible:

$$n \mapsto w^{(A,B,C)}_{(G,E),f,D}(n) - 2^{-c(n)}\,. \tag{4.40}$$

Note that other works present definitions based on alternate notions of trivial success probabilities. For example, [CMP24] permits the splitting adversary A_n to optimally choose which of B_n or C_n should received the unmodified program state and further allows the non-receiving guessing adversary to learn their challenge value x before making their guess for $f_n(k, x)$. As this can only increase the value of the trivial winning probability above $2^{-c(n)}$, the security notion we present here is stronger. As another example, [CLLZ21] defines security with respect to a trivial winning probability which is 0 or negligible. While this is in agreement with our definition if $n \mapsto 2^{-c(n)}$ is negligible, it is an impractical definition to use when this is not the case, such as if c is constant function.

We leave to future work a more detailed comparison of existing security notions together with their notions of trivial success probabilities.

4.5.2 Defining Uncloneable Advice

Here, we briefly review the necessary preliminaries on complexity theory and then define the complexity classes **neglQP** and **neglQP**/upoly. We generally follow the definitions and conventions given in Watrous' survey of quantum complexity theory [Wat09].

First, we recall the definitions of a promise problem and of a language.

Definition 4.25. A *promise problem* $P = (P_0, P_1)$ is composed of a pair of disjoint subsets $P_0, P_1 \subseteq \{0,1\}^*$. The elements of P_1 are called the *yes instances* and those of P_0 are the *no instances*. Both *yes* and *no* instances are called *problem instances*. If $P_0 \cup P_1 = \{0,1\}^*$, we say that P is a *language* and the elements of P_1 are the *words* of this language. We also establish some additional notation by overloading the symbol P twice: first as a set, then as a map.

We let $P = P_0 \cup P_1$ be the set of all problem instances. For all $n \in \mathbb{N}$, we also define the sets $P^n = P \cap \{0,1\}^n$, $P_0^n = P_0 \cap \{0,1\}^n$, and $P_1^n = P_1 \cap \{0,1\}^n$ to be all problem instances of length n, all *no* instances of length n, and all *yes* instances of length n.

Second, we let $P : P_0 \cup P_1 \to \{0,1\}$ be the unique map satisfying $P(x) = 1 \iff x \in P_1$. We identify computing the map P with the ability to *solve* the problem P. For all $n \in \mathbb{N}$, we also define $P^n : P_0^n \cup P_1^n \to \{0,1\}$ to be the unique map satisfying $P^n(x) = 1 \iff x \in P_1^n$.

Note that, with the notation of Definition 4.25, a problem $P = (P_0, P_1)$ is completely characterized by the resulting sequence of maps $(P^n : P_0^n \cup P_1^n \to \{0,1\})_{n \in \mathbb{N}}$.

Before proceeding to defining our novel complexity classes, we recall the classes of problems which can be solved in quantum polynomial-time with bounded errors, with or without quantum advice. As usual, we denote these classes by **BQP** and **BQP**/qpoly.

Definition 4.26. Let $a, b : \mathbb{N} \to \mathbb{R}$ be two maps. A promise problem P is in the class $\mathbf{BQP}(a,b)$ if there exists a polynomial-time $(n,1)$-circuit family $(C_n)_{n \in \mathbb{N}}$ such that the following hold:

1. For all $x \in P_1$, we have that $\langle 1 | C_{|x|}(x) | 1 \rangle \geq a(|x|)$.
2. For all $x \in P_0$, we have that $\langle 1 | C_{|x|}(x) | 1 \rangle \leq b(|x|)$.

Definition 4.27. Let $a, b : \mathbb{N} \to \mathbb{R}$ be two maps. A promise problem P is in the class $\mathbf{BQP}(a,b)$/qpoly if there exists a sequence of states $(\rho_n)_{n \in \mathbb{N}}$, each on $q(n)$ qubits for a polynomially bounded $q : \mathbb{N} \to \mathbb{N}$, and a polynomial-time $(q+n,1)$-circuit family $(C_n)_{n \in \mathbb{N}}$ such that the following hold:

1. For all $x \in P_1$, we have that $\langle 1 | C_{|x|}(\rho_{|x|} \otimes x) | 1 \rangle \geq a(|x|)$.

2. For all $x \in P_0$, we have that $\langle 1| C_{|x|}(\rho_{|x|} \otimes x) |1\rangle \leq b(|x|)$.

By a standard error-reduction-by-repetition argument, we can show that there is a large flexibility in the choices of a and b without changing the underlying class, so long as these are sufficiently bounded. Thus, we define $\mathbf{BQP} = \mathbf{BQP}(2/3, 1/3)$ and $\mathbf{BQP}/\text{qpoly} = \mathbf{BQP}(2/3, 1/3)/\text{qpoly}$.

We are now ready to define our novel complexity classes. We begin by defining the class of problems which can be solved by polynomial-time quantum computations with negligible errors. We denote this class **neglQP** and immediately note that it is equivalent to **BQP** due to standard error reduction techniques for this latter class.

Definition 4.28. A problem P is in the complexity class **neglQP** if there exists a polynomial-time circuit family $(C_n)_{n\in\mathbb{N}}$ and a negligible function η such that

$$x \in P \implies \langle P(x)| C_{|x|}(x) |P(x)\rangle \geq 1 - \eta(|x|). \tag{4.41}$$

Lemma 4.29. neglQP = BQP.

Proof: The inclusion $\mathbf{BQP} \subseteq \mathbf{neglQP}$ follows directly by the standard error reduction technique for **BQP**. To show the other inclusion, consider a problem $P \in \mathbf{neglPQ}$ which is solved by a polynomial-time circuit family $(C_n)_{n\in\mathbb{N}}$ with negligible error η. In particular, there exists an $n_0 \in \mathbb{N}$ such that $n \geq n_0 \implies \eta(n) \leq \frac{1}{3}$. Thus, we can consider a new circuit family $(C'_n)_n$ where each C'_n for $n < n_0$ simply "hard codes" each solution to each problem instance. Otherwise, if $n \geq n_0$, we simply take $C'_n = C_n$. It follows that $(C'_n)_{n\in\mathbb{N}}$ is a polynomial-time circuit family which solves P with error at most $\frac{1}{3}$. Hence, we have that $P \in \mathbf{BQP}$ and so that $\mathbf{neglQP} \subseteq \mathbf{BQP}$. ∎

We can now state our definition for the complexity class of problems which can be solved with negligible error by polynomial-time quantum computation with the help of *uncloneable* advice, which we denote **neglQP**/upoly.

As previously discussed in Section 4.1.1.1, a problem P is in **neglQP**/upoly if there exists a sequence of advice states $(\rho_n)_{n\in\mathbb{N}}$ satisfying the following two criteria. First, an honest user must be able to solve problem instances in P with negligible errors when given a single copy of the advice state. Second, a malicious user given a single copy of the advice state cannot share this state between two other non-communicating malicious users such that they could both solve problem instances in P with more than a negligible advantage. We emphasize again that this is analogous to the security criteria of a copy-protection scheme as set out in Definition 4.24. The following definition formalizes the above.

Definition 4.30. A problem $P = (P_0, P_1)$ is in the complexity class **neglQP**/upoly if there exists a sequence of quantum states $(\rho_n)_{n\in\mathbb{N}}$, each on $q(n)$ qubits respectively for a polynomially bounded map $q : \mathbb{N} \to \mathbb{N}$, such that the following two conditions hold:

1. *Correctness.* There exists a polynomial-time $(q + n, 1)$-circuit family $(C_n)_{n\in\mathbb{N}}$ and a negligible function η such that

$$x \in P \implies \langle P(x)| C_{|x|} \left(\rho_{|x|} \otimes x\right) |P(x)\rangle \geq 1 - \eta(|x|). \tag{4.42}$$

2. *Uncloneability.* For each $n \in \mathbb{N}$, let

$$k_n = \begin{cases} 1 & \text{if either } P_0^n \text{ or } P_1^n \text{ is empty} \\ \frac{1}{2} & \text{else.} \end{cases} \tag{4.43}$$

and let D_n be a random variable distributed on $\{0,1\}^n$ such that for both $b \in \{0,1\}$ we have that

$$x \in P_b^n \implies \Pr[D_n = x] = k_n \cdot \frac{1}{|P_b^n|}. \tag{4.44}$$

Then, for all triplets (A, B, C) of polynomial-time $(q, q_B + q_C)$-, $(n + q_B, 1)$-, and $(q_C + n, 1)$-circuit families, respectively, there exists a negligible function η' such that for all $n \in \mathbb{N}$ satisfying $P^n \neq \varnothing$ we have that

$$\mathop{\mathbb{E}}_{(x_B,x_C)\leftarrow D_n\times D_n} \langle P(x_B), P(x_C)| \left(B_n \otimes C_n\right) \left(x_b \otimes A_n(\rho_n) \otimes x_c\right) |P(x_B), P(x_C)\rangle \tag{4.45}$$

is at most $\frac{1}{2} + \eta'(n)$.

Note that $\frac{1}{2}$ in the second condition plays the same role as the $2^{-c(n)}$ term in the definition of security for copy-protection: it captures the maximal winning probability of strategies which give the advice unmodified to one guessing party and has the other guessing party output an element uniformly at random from the appropriate codomain.

We give a few other remarks on this definition.

Remark 4.31. It would be possible to strengthen Definition 4.30 by weakening the computational constraints on the adversaries considered in uncloneability criterion. While we do not formally define these variations here, we note that the advice we construct in the proof of Theorem 4.36 would fulfill the uncloneability criterion against any uniform adversaries, not only polynomial-time adversaries.

Remark 4.32. For a given problem P, the sequence of random variables $(D_n)_{n\in\mathbb{N}}$ considered in point 2 of Definition 4.30 is not uniquely defined. Indeed, for a particular $n \in \mathbb{N}$, the

random variable D_n is uniquely defined if $P \cap \{0,1\}^n \neq \varnothing$ and is unconstrained otherwise. However, our definition is insensitive to D_n for the values of n where $P \cap \{0,1\}^n = \varnothing$ and so this ambiguity is not an issue.

Remark 4.33. Our definition of the sequence of random variables D in point 2 of Definition 4.30 could be changed, leading to possibly distinct complexity classes.

Note that while we define random variables D_n over P^n, in Equation 4.45 we actually use the random variables $D_n \times D_n$ over $\{0,1\}^n \times \{0,1\}^n$. In particular, any other definition of a sequence of random variables $(D'_n)_{n\in\mathbb{N}}$ on $\{0,1\}^n \times \{0,1\}^n$, provided that it satisfy

$$\forall n \in \mathbb{N} \qquad P^n \neq \varnothing \implies \Pr\left[D'_n \in P^n \times P^n\right] = 1, \tag{4.46}$$

could replace the sequence $(D_n \times D_n)_{n\in\mathbb{N}}$ in Definition 4.30 and the definition would remain coherent. We note, however, that the choice of k_n on the right-hand side of Equation 4.45 may no longer be the most appropriate for certain choices of random variables. Requiring the random variables to satisfy Equation 4.46 simply enforces the condition that the adversaries should be challenged on elements of P with certainty.

We emphasize here that the random variables D_n depend on the problem P. If we wish to consider promise problems which are not languages, then it seems natural that the random variables depend on P. Indeed, if the random variables did not depend on P, it is unclear what we should ask of the adversaries when they are challenged on inputs which are not in P. While there are sensible answers to this question, such as considering any output to be valid or not consider these cases when computing the success probability of the adversaries, we leave further discussions along these lines to future work.

We highlight three other natural ways in which we could have defined the random variables from which the problem instances (x_0, x_1) are sampled. These emerge from the four possible ways to answer the following two questions:

- Should x_0 and x_1 be equal, or sampled independently?
- Should the challenges be sampled uniformly at random from P^n, or in such a way which makes 0 and 1 equally likely to be the correct answers (when possible)?

Our choice for the random variables results from taking the latter answer to both questions.

Finally, we note that this discussion on which random variables to use when challenging the adversaries for the uncloneability criterion echoes extremely similar discussions pertaining to secure software leasing [AL21, BJL+21a] and copy-protected functions [CMP24].

With Definition 4.30 in hand, we can now formalize the relation between copy-protection and uncloneable advice. Let (G, E) be a pair of circuit families forming a copy-protection

scheme as defined in Definition 4.19, except that G need not be efficient, or even uniform: G can be an arbitrary circuit family. Call such a scheme a *copy-protection scheme with unconstrained generation*. Correctness and security for copy-protection schemes with unconstrained generation is exactly as defined for usual copy-protection schemes. The next theorem states that a problem P is in **neglQP**/upoly if and only if there exists a copy-protection with unconstrained generation (G, E) which is correct and secure for the sequence of maps $(P^n : P_0^n \cup P_1^n \to \{0,1\})_{n\in\mathbb{N}}$ with respect to random variables $D = (D_n)_{n\in\mathbb{N}}$ as described in Definition 4.30.

Theorem 4.34. Let P be a problem and let $D = (D_n)_{n\in\mathbb{N}}$ be a sequence of random variables as described in Definition 4.30. For all $n \in \mathbb{N}$, let $\tilde{P}^n : \{0,1\}^0 \times (P_0^n \cup P_1^n) \to \{0,1\}$ be the map defined by $\tilde{P}^n(\varepsilon, x) = P^n(x)$ for all $x \in P_0^n \cup P_1^n$ and let $\tilde{D}_n$ be the random variable defined by

$$\Pr\left[\tilde{D}_n = (\varepsilon, x_B, x_C)\right] = \Pr\left[D_n \times D_n = (x_B, x_C)\right]. \tag{4.47}$$

Then, P is in **neglQP**/upoly if and only if there exists a copy-protection scheme with unconstrained generation (G, E) for the sequence of maps $\tilde{P} = (\tilde{P}^n)_{n\in\mathbb{N}}$ which is correct and secure with respect to the random variables $\tilde{D} = (\tilde{D}_n)_{n\in\mathbb{N}}$.

Proof: Assume that P is in **nelgQP**/upoly by using the sequence of advice states $(\rho_n)_{n\in\mathbb{N}}$. Thus, there exists an efficient circuit family $C = (C_n)_{n\in\mathbb{N}}$ and a negligible function η such that $x \in P$ implies that $\langle P(x)| C_n(\rho_{|x|} \otimes x) |P(x)\rangle \geq 1 - \eta(|x|)$. Moreover, there exists a circuit family $G = (G_n)_{n\in\mathbb{N}}$ such that $\frac{1}{2}\|\rho_n - G_n(\varepsilon)\|_1 \leq 2^{-n}$.[14] Consider now (G, C) as a copy-protection scheme with unconstrained generation for $(\tilde{P}^n)_{n\in\mathbb{N}}$. We note two things:

1. The correctness of the advice directly implies the correctness of (G, C) for $(\tilde{P}^n)_{n\in\mathbb{N}}$.

2. The uncloneability of the advice implies that the scheme (G, C) is secure for $(\tilde{P}^n)_{n\in\mathbb{N}}$ with respect to $(\tilde{D}_n)_{n\in\mathbb{N}}$.

   Indeed, adversaries against the scheme (G, C) are precisely adversaries against the advice states $(\rho_n)_{n\in\mathbb{N}}$ for the uncloneability criterion of **neglQP**/upoly and the winning probability of such an adversary is identical in both scenarios for those values of n satisfying $P^n \neq \varnothing$. As the advice is uncloneable, we have that for all adversaries (A, B, C) there exists a negligible function η such that $P^n \neq 0$ implies that $\omega^{(A,B,C)}_{(G,C),\tilde{P},\tilde{D}}(n) \leq \frac{1}{2} + \eta(n)$. Moreover, by definition, we have that $\omega^{(A,B,C)}_{(G,C),\tilde{P},\tilde{D}}(n) = 0$ whenever $P^n = \varnothing$. It follows that $n \mapsto \omega^{(A,B,C)}_{(G,C),\tilde{P},\tilde{D}} - \frac{1}{2}$ is negligible and so (G, C) is secure for $\tilde{P}$ with respect to $\tilde{D}$.

[14] It may be that it is impossible to achieve $\rho_n = G_n(\varepsilon)$ for all n due to the underlying gate set from which the circuits in G are constructed. This negligible error is not an issue as both correctness and security of uncloneable advice is defined up to negligible errors.

For the other direction, assume that (G, E) is a copy-protection scheme with unconstrained generation for $(\tilde{P}^n)_{n\in\mathbb{N}}$ which is correct and secure with respect to $(\tilde{D}_n)_{n\in\mathbb{N}}$. For each $n \in \mathbb{N}$, take $\rho_n = G_n(\varepsilon)$. It follows that P is in **neglQP**/upoly with the help of the advice states $(\rho_n)_{n\in\mathbb{N}}$ since correctness and security of the copy-protection scheme directly imply correctness and security of the uncloneable advice. ∎

As a final remark in this section, we note that the intersection of **BQP** and **neglQP**/upoly can only contain essentially trivial problems.

Proposition 4.35. A problem P is in $\mathbf{BQP} \cap \mathbf{neglQP}$/upoly if and only if $|P|$ is finite.

The proof of this proposition reduces to two ideas. First, if $P \in \mathbf{BQP}$ then it can be solved with the empty state $1 \in \mathcal{D}(\mathbb{C})$ as advice. Second, the empty state is perfectly cloneable. Or, even more precisely, two copies of the empty state can be easily produced from any other state.

Proof: Assume that P is a problem such that $|P|$ is finite. Then, it is trivially true that $P \in \mathbf{BQP}$. This can be shown by considering a circuit family $(C_n)_{n\in\mathbb{N}}$ where each C_n has a "hard-coded" list of the elements of P_0^n which can then be used to solve all problem instances of length n by simply checking if the instance is in P_0^n or not. As $|P|$ is finite, this can be implemented efficiently. On the other hand, let $m \in \mathbb{N}$ be the maximal length of the elements in P, i.e.: $x \in P \implies |x| \leq m$. Consider the negligible function $\eta : \mathbb{N} \to \mathbb{R}$ defined by $\eta(n) = 1$ if $n \leq m$ and $\eta(n) = 0$ if $n > m$ and the sequence of empty states $(\rho_n = 1)_{n\in\mathbb{N}}$. Correctness and uncloneability as given in Definition 4.30 holds with respect to this sequence of states and this negligible function. Thus, $P \in \mathbf{neglQP}$/upoly.

Now, for the other direction, assume that $P \in \mathbf{BQP}$ and $P \in \mathbf{neglQP}$/upoly where the second inclusion is obtained with respect to the sequence of advice states $(\rho_n)_{n\in\mathbb{N}}$. Let $C = (C_n)_{n\in\mathbb{N}}$ be an efficient $(n, 1)$-circuit family and η be a negligible function such that $x \in P \implies \langle P(x)| C_{|x|}(x) |P(x)\rangle \geq 1 - \eta(|x|)$. Such a negligible function and circuit family exists by virtue of P being in **BQP**. Consider now the triplet of efficient circuit families (A, C, C) where each circuit A_n simply discards its input. Finally, let $(D_n)_{n\in\mathbb{N}}$ be a sequence of random variables as defined in Definition 4.30. As the circuit family C solves P with negligible error on all inputs and P is assumed to be in **neglQP**/upoly, there exists another negligible function η' such that for all $n \in \mathbb{N}$ satisfying $P^n \neq \varnothing$ we have

$$\mathop{\mathbb{E}}_{(x_B,x_C)\leftarrow D_n\times D_n} \langle P(x_B), P(x_C)| (C_n \otimes C_n) (x_B \otimes A_n(\rho_n) \otimes x_C) |P(x_B), P(x_C)\rangle \tag{4.48}$$

is lower bounded by $1 - 2\eta(n)$ and upper bounded by $\frac{1}{2} + \eta'(n)$. Thus, $\frac{1}{2} \leq \eta'(n) + 2\eta(n)$. Since η' and η are negligible, this inequality can hold for at most finitely many values of n.

It follows that $P^n = \varnothing$ for all sufficiently large values of n, implying that $|P|$ is finite. ∎

4.5.3 A Promise Problem with Uncloneable Advice

In this section, we formally describe the promise problem with uncloneable advice which was discussed in Section 4.1.1.4. The goal is to prove the following theorem.

Theorem 4.36. Let $(s_\lambda)_{\lambda\in\mathbb{N}}$ be an exponential-time ingenerable 2λ-sequence and parse each s_λ as a pair of strings of length λ, *i.e.:* $s_\lambda = (x_\lambda, \theta_\lambda)$ for $x_\lambda, \theta_\lambda \in \{0,1\}^\lambda$ for all $\lambda \in \mathbb{N}$. For each $b \in \{0,1\}$, let

$$P_b = \bigcup_{\lambda\in\mathbb{N}} \left\{ (\theta_\lambda, y) \ : \ y \in \{0,1\}^\lambda \wedge x_\lambda \cdot y = b \right\}. \tag{4.49}$$

Then, the promise problem $P = (P_0, P_1)$ is in **neglQP**/upoly.

We prove this theorem by first collecting three lemmas, each encapsulating one step of the argument we made in Section 4.1.1.4. We begin by recalling a lemma concerning the monogamy-of-entanglement game of [TFKW13].[15]

Lemma 4.37 ([TFKW13])**.** Let $n, a_B, a_C \in \mathbb{N}$ be three non-negative integers and let (A, B, C) be a triplet of $(n, a_B + a_C)$-, $(a_B + n, n)$-, and $(a_C + n, n)$-circuits, respectively. Then, we have that

$$\mathop{\mathbb{E}}_{\substack{x\leftarrow\{0,1\}^n \\ \theta\leftarrow\{0,1\}^n}} \langle x, x| (B \otimes C) \left(\theta \otimes A\left(|x^\theta\rangle\langle x^\theta|\right) \otimes \theta\right) |x, x\rangle \leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^n < 0.86^n. \tag{4.50}$$

As for Lemma 4.13, this lemma holds for any triplet of channels, but we frame it in terms of circuits to maintain consistency with the rest of this work and, in particular, this section. Now, we "derandomize" the above with ingenerable sequences, analogously to how we previously obtained Theorem 4.14.

Lemma 4.38. Let $(s_\lambda)_{\lambda\in\mathbb{N}}$ be an exponential-time ingenerable 2λ-sequence and parse each s_λ as a pair of strings of length λ, *i.e.:* $s_\lambda = (x_\lambda, \theta_\lambda)$ for $x_\lambda, \theta_\lambda \in \{0,1\}^\lambda$ for all $\lambda \in \mathbb{N}$.

For any triplet $((A_\lambda)_{\lambda\in\mathbb{N}}, (B_\lambda)_{\lambda\in\mathbb{N}}, (C_\lambda)_{\lambda\in\mathbb{N}})$ of exponential-time $(\lambda, b+c)$-, $(\lambda+b, \lambda)$-, and $(c+\lambda, \lambda)$-circuit families for maps $b, c : \mathbb{N} \to \mathbb{N}$, respectively, we have that

$$\lambda \mapsto \langle x_\lambda, x_\lambda| (B_\lambda \otimes C_\lambda) \left(\theta_\lambda \otimes A_\lambda\left(|x_\lambda^{\theta_\lambda}\rangle\langle x_\lambda^{\theta_\lambda}|\right) \otimes \theta_\lambda\right) |x_\lambda, x_\lambda\rangle \tag{4.51}$$

[15] This particular form of the lemma was first stated, essentially, in [BL20]. However, it is an immediate corollary of a more general result found in [TFKW13] obtained by exploiting the operational relation between measuring half an EPR pair [EPR35] and sending a random Wiesner state.

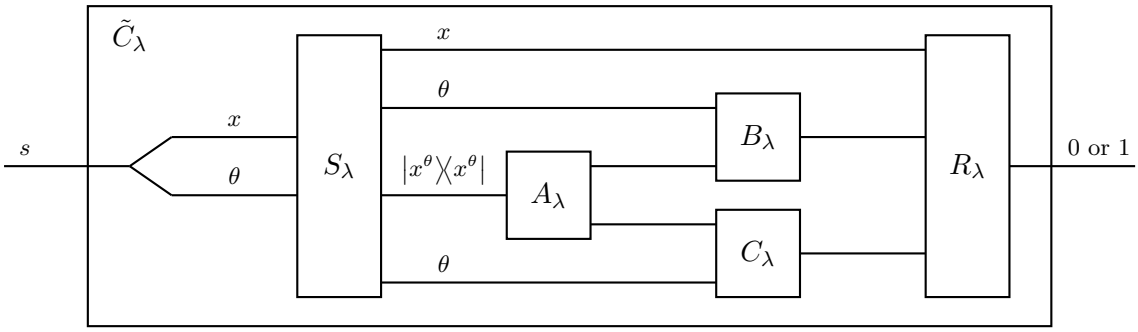


Figure 4.3: A representation of the $\tilde{C}_\lambda$ circuit constructed in the proof of Lemma 4.38

is negligible.

Proof: It suffices to apply Theorem 4.10 to Lemma 4.37. We follow the same general ideas as in the proof of Theorem 4.14.

Let $(S_\lambda)_{\lambda\in\mathbb{N}}$ be a polynomial-time family of $(2\lambda, 4\lambda)$-circuits such that each circuit, on input of $x \otimes \theta$, outputs the state $x \otimes \theta \otimes |x^\theta\rangle\langle x^\theta| \otimes \theta$ for any $x, \theta \in \{0,1\}^\lambda$. Let $(R_\lambda)_{\lambda\in\mathbb{N}}$ be a polynomial-time family of $(3\lambda, 1)$-circuits where each circuit measures all of its input qubits in the computational bases, parses it as three strings of length λ, and outputs 1 if all three are equal and 0 otherwise.

We now consider the exponential-time family of $(2\lambda, 1)$-circuits $(\tilde{C}_\lambda)_{\lambda\in\mathbb{N}}$ such that, for all $\lambda \in \mathbb{N}$, $\tilde{C}_\lambda$ is the composition of the A_λ, B_λ, C_λ, S_λ, and R_λ circuits such that the resulting channel is given by

$$\tilde{C}_\lambda = R_\lambda \circ (\mathrm{Id}_\lambda \otimes B_\lambda \otimes C_\lambda) \circ (\mathrm{Id}_\lambda \otimes \mathrm{Id}_\lambda \otimes A_\lambda \otimes \mathrm{Id}_\lambda) \circ S_\lambda. \tag{4.52}$$

This composition is illustrated in Figure 4.3. For all $x, \theta \in \{0,1\}^\lambda$ we have that

$$\langle 1|\, \tilde{C}_\lambda(x \otimes \theta)\, |1\rangle = \langle x, x|\, (B_\lambda \otimes C_\lambda) \left(\theta \otimes A_\lambda(\, |x^\theta\rangle\langle x^\theta|) \otimes \theta\right) |x, x\rangle \tag{4.53}$$

In particular, by Lemma 4.37,

$$\mathop{\mathbb{E}}_{\substack{x\leftarrow\{0,1\}^\lambda \\ \theta\leftarrow\{0,1\}^\lambda}} \langle 1|\, \tilde{C}_\lambda(x \otimes \theta)\, |1\rangle \leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^\lambda \tag{4.54}$$

is a negligible function. Thus, by Theorem 4.10 and the fact that $(s_\lambda)_{\lambda\in\mathbb{N}}$ is an exponential-time ingenerable 2λ-sequence, the function $\lambda \mapsto \langle 1|\, \tilde{C}_\lambda(x_\lambda \otimes \theta_\lambda)\, |1\rangle$ is also negligible which, by Equation 4.53, is the desired result. ∎

Finally, we can apply Lemma 4.1 to show that any exponential-time adversary will have a negligible advantage determining the inner product of x_λ with uniformly random string y.

Lemma 4.39. Let $(s_\lambda)_{\lambda\in\mathbb{N}}$ be an exponential-time ingenerable 2λ-sequence and parse each s_λ as a pair of strings of length λ, *i.e.:* $s_\lambda = (x_\lambda, \theta_\lambda)$ for two strings $x_\lambda, \theta_\lambda \in \{0,1\}^\lambda$ for all $\lambda \in \mathbb{N}$. Then, for any triplet $((A_\lambda)_{\lambda\in\mathbb{N}}, (B_\lambda)_{\lambda\in\mathbb{N}}, (C_\lambda)_{\lambda\in\mathbb{N}})$ of exponential-time $(\lambda, b+c)$-, $(2\lambda+b, 1)$-, and $(c+2\lambda, 1)$-circuit families, respectively and for maps $b, c : \mathbb{N} \to \mathbb{N}$, there exists a negligible function η such that

$$\mathop{\mathbb{E}}_{\substack{y_B\leftarrow\{0,1\}^\lambda\\ y_C\leftarrow\{0,1\}^\lambda}} \langle x_\lambda\cdot y_B, x_\lambda\cdot y_C| \left(B_\lambda\otimes C_\lambda\right)\left(y_B\otimes\theta_\lambda\otimes\sigma_\lambda\otimes\theta_\lambda\otimes y_C\right)|x_\lambda\cdot y_B, x_\lambda\cdot y_C\rangle = \frac{1}{2}+\eta(\lambda) \tag{4.55}$$

where $\sigma_\lambda = A_\lambda\left(|x_\lambda^{\theta_\lambda}\rangle\langle x_\lambda^{\theta_\lambda}|\right)$.

Proof: In the notation of Lemma 4.1, we let ξ_λ be the random variable distributed on the set $\{0,1\}^\lambda\times\{0,1\}^\lambda$ such that $\Pr[\xi_\lambda = (x_\lambda, x_\lambda)] = 1$ and we let $\rho_{\lambda,x_\lambda,x_\lambda} = \theta_\lambda\otimes\sigma_\lambda\otimes\theta_\lambda$.

By Lemma 4.38, for every pair $((B'_\lambda)_{\lambda\in\mathbb{N}}, (C'_\lambda)_{\lambda\in\mathbb{N}})$ of exponential-time $(\lambda+b,\lambda)$- and $(c+\lambda,\lambda)$-circuits, respectively, the function

$$\lambda \mapsto \langle x_\lambda, x_\lambda| \left(B'_\lambda\otimes C'_\lambda\right)(\rho_{\lambda,x_\lambda,x_\lambda})|x_\lambda, x_\lambda\rangle \tag{4.56}$$

is negligible. Since the support of ξ_λ is $\{(x_\lambda, x_\lambda)\}$, the premise of Lemma 4.1 is satisfied. Thus,

$$\lambda \mapsto \mathop{\mathbb{E}}_{\substack{y_B\leftarrow\{0,1\}^\lambda\\ y_C\leftarrow\{0,1\}^\lambda}} \langle x_\lambda\cdot y_B, x_\lambda\cdot y_C| \left(B_\lambda\otimes C_\lambda\right)\left(y_B\otimes\rho_\lambda\otimes y_C\right)|x_\lambda\cdot y_B, x_\lambda\cdot y_C\rangle - \frac{1}{2} \tag{4.57}$$

is also negligible. Taking η to be this map and expanding the definition of ρ_λ yields the result. ∎

We give a final technical lemma concerning exponential-time ingenerable sequences before proving the theorem.

Lemma 4.40. Let $(s_\lambda)_{\lambda\in\mathbb{N}}$ be an exponential-time ingenerable 2λ-sequence and parse each s_λ as $(x_\lambda, \theta_\lambda)$ for two strings $x_\lambda, \theta_\lambda \in \{0,1\}^\lambda$. Then, there exists a $\tilde{\lambda}$ such that $\lambda \geq \tilde{\lambda}$ implies that $x_\lambda \neq 0^\lambda$.

Proof: Assume this was not the case and $x_\lambda = 0^\lambda$ infinitely often. Consider now the polynomial-time circuit family $(C_\lambda)_{\lambda\in\mathbb{N}}$ where C_λ samples a uniformly random $y \in \{0,1\}^\lambda$

and outputs $(0^\lambda, y)$. Then, $\langle s_\lambda | C_\lambda(\varepsilon) | s_\lambda \rangle = 2^{-\lambda}$ infinitely often. But, as $(s_\lambda)_{\lambda\in\mathbb{N}}$ is exponential-time ingenerable, there is a polynomial p such that $\langle s_\lambda | C_\lambda(\varepsilon) | s_\lambda \rangle < p(n)2^{-2\lambda}$ for all sufficiently large values of λ. This implies that $2^\lambda < p(\lambda)$ infinitely often, a contradiction. Hence, $x_\lambda = 0^\lambda$ only finitely often. ∎

We can now give the proof of Theorem 4.36. Note that we will use ν as our main indexing variable and set $\nu = 2\lambda$ whenever ν is even. This will make the notation pertaining to λ a bit more consistent when invoking the previous lemma.

Proof: Define the sequence of states $(\rho_\nu)_{\nu\in\mathbb{N}}$ as

$$\rho_\nu = \begin{cases} |x_\lambda^{\theta_\lambda}\rangle\langle x_\lambda^{\theta_\lambda}| & \text{if} \quad \nu = 2\lambda \text{ for } \lambda \in \mathbb{N} \\ \varepsilon & \text{else.} \end{cases} \tag{4.58}$$

for all $\nu \in \mathbb{N}$. These will be our advice states for P. Let $q : \mathbb{N} \to \mathbb{N}$ be such that $q(\lambda)$ is the number of qubits in the advice state ρ_ν. Note that $q(\nu) = \lambda$ if $\nu = 2\lambda$ and $q(\nu) = 0$ otherwise.

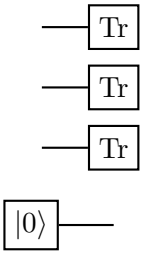
Correctness. To show the correctness criterion, we give an explicit polynomial-time circuit family which solves P with certainty when given these advice states. Let $(C_\nu)_{\nu\in\mathbb{N}}$ be the polynomial-time $(q+\nu, 1)$-circuit family which is described below:

- If ν is odd, C_ν discards the input and outputs a single qubit in the 0 state.
- If $\nu = 2\lambda$ is even, C_ν does the following:
  1. Parse the $q(\nu) + \nu = 3\lambda$ input qubits as three registers $\mathsf{A} \otimes \mathsf{B} \otimes \mathsf{S}$ of λ qubits each.
  2. For each $i \in [\lambda]$, apply a Hadamard gate H on the i-th qubit of A conditioned on the i-th qubit of B.
  3. Using an extra qubit initialized in the 0 state, which we will call the R register, compute the inner product of the A and S registers and store the result in the R register. We do this by applying λ Toffoli gates on R, each conditioned on the i-th qubits of A and S, respectively, for each $i \in [\lambda]$.
  4. Discard the A, B, and S registers and output the R register.

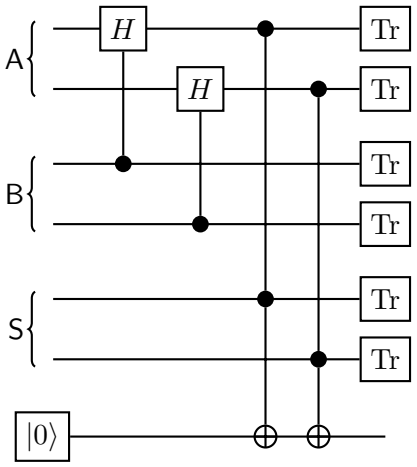
Examples of these circuits, for $\nu = 3$ and $\nu = 4$, are illustrated in Figure 4.4.

For all $\lambda \in \mathbb{N}$ and all strings $y \in \{0,1\}^\lambda$, a direct calculation yields

$$C_{2\lambda}\left(\rho_{2\lambda} \otimes \theta_\lambda \otimes y\right) = C_{2\lambda}\left(|x_\lambda^{\theta_\lambda}\rangle\langle x_\lambda^{\theta_\lambda}| \otimes \theta_\lambda \otimes y\right) = x_\lambda \cdot y = P(\theta_\lambda, y) \tag{4.59}$$



(a) The circuit C_3.



(b) The circuit C_4. The advice state is inputted in the A register and the problem instance in the B and S registers.

Figure 4.4: Two of the C_ν circuits used to show correctness in the proof of Theorem 4.36.

Thus, for any $z \in P$, which must be of the form $z = (\theta_\lambda, y)$ for some $y \in \{0,1\}^\lambda$, we have that $C_{|z|}$ correctly computes $P(z)$ with certainty when also given the advice state $\rho_{|z|}$. Hence, the correctness criterion is satisfied.

Uncloneability. Our first step in showing that these advice states satisfy the uncloneability criterion is to gain a better understanding of the sequence of random variables $(D_\nu \times D_\nu)_{\nu\in\mathbb{N}}$, as given in Definition 4.30, for this promise problem.

Recall that by Lemma 4.40, there are only finitely many instances where $x_\lambda = 0^\lambda$. This implies that for each $b \in \{0,1\}$ and all sufficiently large values of λ, the set $P_b^{2\lambda}$ is exactly half of $P^{2\lambda}$. Indeed, this follows from the fact that $\left|\{y \in \{0,1\}^\lambda \,:\, x \cdot y = b\}\right|$ is $2^{\lambda-1}$ precisely when $x \neq 0^\lambda$. This is to say that, for all sufficiently large λ, there are always as many yes instances as no instances of length 2λ. In particular, for all sufficiently large values of λ, the random variable $D_{2\lambda}$ is uniformly random over $P^{2\lambda} = \{\theta_\lambda\} \times \{0,1\}^\lambda$.

Now, consider a triplet $((A_\nu)_{\nu\in\mathbb{N}}, (B_\nu)_{\nu\in\mathbb{N}}, (C_\nu)_{\nu\in\mathbb{N}})$ of polynomial-time $(q, b+c)$-, $(2\lambda+b, 1)$-, and $(c+2\lambda, 1)$-circuit families, respectively and for maps $b, c : \mathbb{N} \to \mathbb{N}$. Let $v : \mathbb{N} \to \mathbb{R}$ be the function given by

$$v(\lambda) = \mathop{\mathbb{E}}_{(z_B,z_C)\leftarrow D_{2\lambda}\times D_{2\lambda}} \langle P(z_B), P(z_C)| \left(B_{2\lambda} \otimes C_{2\lambda}\right) \left(z_B \otimes A_{2\lambda}\left(\rho_{2\lambda}\right) \otimes z_C\right) |P(z_B), P(z_C)\rangle \,. \tag{4.60}$$

In other words, $v(\lambda)$ is the probability that the triplet of circuit families split and successfully

use the advice state to solve problem instances in $P^{2\lambda}$ sampled according to $D_{2\lambda} \times D_{2\lambda}$. To show that the uncloneability criterion is satisfied, it suffices so show the existence of a negligible function η such that $v(\lambda) \leq \frac{1}{2} + \eta(\lambda)$ as, in the notation of Definition 4.30, it will then suffice to take η' to be the function such that $\eta'(\nu) = 0$ if ν is odd and $\eta'(\nu) = \eta(\lambda)$ if $\nu = 2\lambda$ is even. Clearly, this η' will be negligible if η is negligible.

For $L \in \{A, B, C\}$, let $L'_\lambda = L_{2\lambda}$ and note that $(L'_\lambda)_{\lambda \in \mathbb{N}}$ is a polynomial-time family of circuits. As the random variable $D_{2\lambda}$ is uniformly distributed on $\{(\theta_\lambda, y) : y \in \{0,1\}^\lambda\}$ for all sufficiently large values of λ, we have that

$$v(\lambda) = \mathop{\mathbb{E}}_{\substack{y_B \leftarrow \{0,1\}^\lambda \\ y_C \leftarrow \{0,1\}^\lambda}} \langle x_\lambda \cdot y_B, x_\lambda \cdot y_C | \left(B'_\lambda \otimes C'_\lambda\right) \left(\theta_\lambda \otimes y_B \otimes A'_\lambda(\rho_{2\lambda}) \otimes \theta_\lambda \otimes y_C\right) | x_\lambda \cdot y_B, x_\lambda \cdot y_C \rangle \tag{4.61}$$

for all sufficiently large values of λ. Up to performing the swapping map $\theta_\lambda \otimes y_B \mapsto y_B \otimes \theta_\lambda$ on the first 2λ qubits given to B'_λ, an operation which can easily be incorporated into this circuit, Lemma 4.39 implies the existence of a negligible function $\tilde{\eta}$ such that the right-hand side of the above equation is precisely $\frac{1}{2} + \tilde{\eta}(\lambda)$. Since $v(\lambda)$ is equal to $\frac{1}{2} + \tilde{\eta}(\lambda)$ for all sufficiently large values of λ, there exists a function $\eta : \mathbb{N} \to \mathbb{R}$ which differs from $\tilde{\eta}$ on only finitely many inputs such that $v = \frac{1}{2} + \eta$. Since η differs only finitely often from a negligible function, it is itself negligible. ∎

Remark 4.41. If the language in Theorem 4.36 is instantiated with an exponential-time ingenerable sequence $(s_n)_{n \in \mathbb{N}}$ which satisfies the stronger notion of being *computably* ingenerable, then the advice states given in the proof also satisfy uncloneability against uniform adversaries.

On the other hand, if the language is instantiated with an exponential time ingenerable sequences which can be computed in triple-exponential time, then we note that the advice states can be generated by quantum circuits described by a Turing machine running in triple-exponential time. By Theorem 4.9, such sequences exist.

Note that the advice states which we give in the proof of Theorem 4.36 are quantum but, in some sense, the advice is actually the *classical* strings $(x_n)_{n \in \mathbb{N}}$. We encode this classical advice into a quantum state only to achieve uncloneability, not additional computational power. With this in mind, the following proposition formalizes the idea that uncloneable advice need not be more powerful than classical advice.

Proposition 4.42. The intersection of **neglQP**/upoly and **P**/poly, where the latter is the class of promise problems which can be solved in polynomial-time by a classical deterministic Turing machine with access to polynomially many classical bits of advice, is non-empty.

Proof: The promise problems constructed in Theorem 4.36, which states that these problems are in **neglQP**/upoly, can also be seen to be in **P**/poly. Indeed, if the problem is instantiated with the exponential-time ingenerable $2n$-sequence $(s_n = (\theta_n, x_n))_{n\in\mathbb{N}}$, then the classical advice strings $(s'_n)_{n\in\mathbb{N}}$ defined by

$$s'_n = \begin{cases} x_n & \text{if} \quad n \text{ is even} \\ \varepsilon & \text{else} \end{cases} \tag{4.62}$$

are sufficient to show that the problem constructed in Theorem 4.36 is also in **P**/poly. ∎

Similarly, it is clear that if a classical party has enough computational power to generate the sequence $(s_n)_{n\in\mathbb{N}}$ or, more precisely, the sequence $(x_n)_{n\in\mathbb{N}}$, then it can solve the promise problem constructed in Theorem 4.36 from $(s_n)_{n\in\mathbb{N}}$.

Proposition 4.43. The intersection of **neglQP**/upoly and **EEEXP**, where the latter is the class of promise problems which can be solved in triple-exponential time by a classical deterministic Turing machine, is non-empty.

Proof: Certain instantiations of the promise problems constructed in Theorem 4.36, which states that these problems are in **neglQP**/upoly, can be seen to be in **EEEXP**. Indeed, if the exponential-time ingenerable sequence $(s_n)_{n\in\mathbb{N}}$ used in the construction is computable in triple-exponential time by a classical deterministic Turing machine, then the resulting promise problem is in **EEEXP**. By Theorem 4.9, such sequences exist. ∎

Propositions 4.42 and 4.43 immediately point to an open question which we believe is of interest: Are there non-trivial problems, i.e.: with infinitely many *yes* and *no* instances, in the intersection of **neglQP**/upoly and **EEXP**, the class of problems which can be solved by classical deterministic Turing machines in double exponential time? What about in the intersection of **neglQP**/upoly and **EXP** or any other classical deterministic super-polynomial time complexity classes? These questions appear to reflect those considered in Section 4.4 about gaps in the complexity of generating and cloning sequences of states.

4.5.4 A Language with Uncloneable Advice

In this section, we describe a class of languages which admit uncloneable advice under the assumption that it is possible to copy-protect certain families of maps. We proceed in two steps. First, we give a general template in Definition 4.44 for the class of languages we consider. Each language from this template is parameterized by a sequence of maps and is illustrated in Figure 4.5. Second, in Theorem 4.46, we establish sufficient conditions

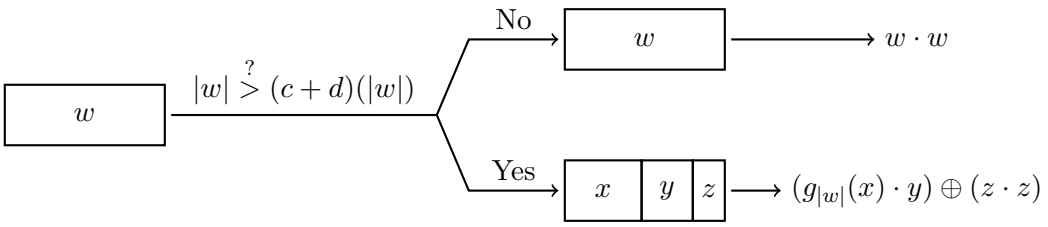


Figure 4.5: An illustration of an algorithm showing how to compute the bit determining if a given string $w \in \{0,1\}^*$ is in the set L_0^g or the set L_1^g as given in Definition 4.44.

on the parametrizing maps which ensure that the resulting language admits uncloneable advice. We comment, in the next section, on certain specific instantiations of this language admitting uncloneable advice based on existing constructions for the copy-protection of pseudorandom functions.

Definition 4.44. Let $g = (g_n : \{0,1\}^{d(n)} \to \{0,1\}^{c(n)})_{n\in\mathbb{N}}$ be a sequence of maps whose domains and codomains are parameterized by $d, c : \mathbb{N} \to \mathbb{N}$. For any such g, we define the language $L^g = (L_0^g, L_1^g)$ as follows:

- If $w \in \{0,1\}^*$ is such that $|w| \leq (d+c)(|w|)$, then

$$w \in L_b^g \iff w \cdot w = b. \tag{4.63}$$

- If $w \in \{0,1\}^*$ is such that $|w| > (d+c)(|w|)$, we parse w as (x,y,z) where x is composed of the first $d(|w|)$ bits of w, y of the next $c(|w|)$ bits, and z of the remaining bits. Then,

$$w = (x,y,z) \in L_b^g \iff (g_{|w|}(x) \cdot y) \oplus (z \cdot z) = b. \tag{4.64}$$

We note a nice property of L^g which is independent of the choice of g, namely that exactly half of all bit strings of a given non-zero length are in the language.

Lemma 4.45. Let $L^g = (L_0^g, L_1^g)$ be as given in Definition 4.44. Then, for all $n \geq 1$ we have that

$$|L_0^g \cap \{0,1\}^n| = |L_1^g \cap \{0,1\}^n| = 2^{n-1}. \tag{4.65}$$

In other words, for any given non-zero length n, there are as many *yes* instances as there are *no* instances of length n in L^g.

Proof: This follows immediately from the fact that for all $n \geq 1$ and $b \in \{0,1\}$ we have that $|\{s \in \{0,1\}^n \ : \ s \cdot s = b\}| = 2^{n-1}$. ∎

We now establish conditions under which the language L^g defined above admits uncloneable advice.

Theorem 4.46. Let $f = (f_n : \{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{c(n)})_{n\in\mathbb{N}}$ be a sequence of maps whose domains and codomains are parameterized by maps $\kappa, d, c : \mathbb{N} \to \mathbb{N}$ such that the following conditions are satisfied:

- There exists an $n' \in \mathbb{N}$ such that $n \geq n' \implies d(n) + c(n) < n$ and $c \in \omega(\log)$.
- There exists a copy-protection scheme (G, E) for f which is correct and secure.

Let $k = (k_n)_{n\in\mathbb{N}}$ be an exponential-time ingenerable κ-sequence and let $g = (g_n)_{n\in\mathbb{N}}$ be the sequence of maps where $g_n = f_n(k_n, \cdot)$ for all $n \in \mathbb{N}$. Then, L^g is in **neglQP**/upoly.

At a high level, the proof proceeds as follows: The advice states will be the program states generated by the copy-protection scheme for the maps in the sequence g. Correctness follows trivially from the correctness of the copy-protection scheme. Uncloneability is a bit more involved. Consider only the cases where $n \geq n'$ and so any string w will be parsed as (x, y, z). The first step is to argue that the z component does not really contribute to the difficulty of determining if w is in the language. The only difficult part is for both guessers to be able to correctly and simultaneously compute $g_n(x) \cdot y$. Since the y's are independently sampled for both guessers, the result of Kundu and Tan (Lemma 4.1) tells us that if both parties can only compute $g_n(x)$ with a negligible probability of success, then they can simultaneously correctly compute this inner product with at most a negligible advantage above $\frac{1}{2}$. How can we show that this probability of simultaneously guessing $g_n(x)$ on independent values of x is negligible? It suffices to derandomize the copy-protection game with the exponential-time ingenerable sequence $(k_n)_{n\in\mathbb{N}}$. More precisely, since the advice states are produced from a secure copy-protection scheme for maps where the trivial guessing probability 2^{-c} is negligible (as $c \in \omega(\log)$), we know that any efficient adversaries will fail to simultaneously and correctly compute $f_n(k', x)$ with more than a negligible probability if the key k' is uniformly random (but identical for both adversaries) and where x is uniformly random and independent for both adversaries. By Theorem 4.10, the probability will remain negligible even when we do not sample k' uniformly at random, but rather fix it to the n-th element of an exponential-time ingenerable κ-sequence.

Proof: Assume the copy-protection scheme (G, E) has program length q. We will consider the sequence of states

$$\rho = (\rho_n = G_n(k_n))_{n\in\mathbb{N}}\,, \tag{4.66}$$

which are the program states produced by the copy-protection scheme (G, E) for the sequence of maps $(g_n)_{n\in\mathbb{N}}$, as our advice for this language.

Correctness. We first show correctness. This follows directly from the assumed correctness of the copy-protection scheme (G, E). More explicitly, let $C = (C_n)_{n\in\mathbb{N}}$ be the efficient $(q+n, 1)$-circuit family where each C_n implements the following algorithm on input of $\rho\otimes w$:

1. Verify if $n \leq d(n) + c(n)$.
   - If this inequality holds, output the bit $w \cdot w$ and terminate.
   - If this inequality does not hold, execute steps 2 to 4 below.
2. Parse w as a triplet (x, y, z) where x is the first $d(n)$ bits, y the subsequence $c(n)$ bits, and z the remaining $n - d(n) + c(n)$ bits.
3. Run $E_n(\rho_n \otimes x)$ and measure the output in the computational basis. Let y' denote this result.
4. Output the bit $(y' \cdot y) \oplus (z \cdot z)$ and terminate.

If $n \leq d(n) + c(n)$, then $C_n(\rho_n \otimes w)$ outputs the correct bit, namely $w \cdot w$, with certainty. Else, $C_n(\rho_n \otimes w)$ outputs the correct bit with a probability at least

$$\langle g_n(x)|\, E_n(\rho_n \otimes x)\, |g_n(x)\rangle = \langle f_n(k_n, x)|\, E_n\left(G_n(k_n) \otimes x\right) |f_n(k_n, x)\rangle \tag{4.67}$$

which is the probability that step 3 correctly evaluates to $g_n(x)$. Since (G, E) is correct, there exists a negligible function η_c such that this probability is at least $1 - \eta_c(n)$. Hence, the correctness criterion is satisfied.

Uncloneability. First, let's set up an adversary and state what is sufficient to show to demonstrate the uncloneability criterion.

Let (A, B, C) be a triplet of polynomial-time $(q, q_B + q_C)$-, $(d + q_B, 1)$-, and $(q_C + d, 1)$-circuit families, respectively, for maps $q_B, q_C : \mathbb{N} \to \mathbb{N}$. Let D_n be the random variable distributed on the set $\{0, 1\}^n$ as defined in Definition 4.30 for the language L^g. Note that, by Lemma 4.45, D_n is uniformly distributed. Let

$$v(n) = \mathop{\mathbb{E}}_{w_B, w_C \leftarrow D_n} \langle L^g(w_B), L^g(w_C)|\, (B_n \otimes C_n)\, (w_B \otimes A_n(\rho_n) \otimes w_C)\, |L^g(w_B), L^g(w_C)\rangle \tag{4.68}$$

be the probability that the adversaries (A, B, C) succeed in sharing the advice state ρ_n and use it to correctly determine the membership of two independent problem instances of length n sampled according to D_n. It suffices to show the existence of a negligible function η such that $v \leq \frac{1}{2} + \eta$. In fact, it suffices to show that this inequality holds for all $n \geq n'$ where, we recall, n' is a threshold after which $n > d(n) + c(n)$ always holds. Assume from

now on that $n \geq n'$ is always satisfied and parse every $w \in \{0,1\}^n$ as a triplet of strings (x, y, z) where x is the first $d(n)$ bits, y the subsequent $c(n)$ bits, and z the remaining bits.

Now, as a first step, we argue that the z component of a problem instance essentially does not matter. In the process, we will express $v(n)$ in a form which will be closer to something to which we can apply Lemma 4.1.

Let B' be a circuit family such that for all $n \geq n'$, the circuit B'_n implements the following algorithm on input of $\rho \otimes x \otimes y$:

- Sample a uniformly random $z \leftarrow \{0,1\}^{n-d(n)-c(n)}$.
- Run $B_n(\rho \otimes x \otimes y \otimes z)$ and measure the output in the computational basis. Let b denote the result of this measurement.
- Output the bit $b \oplus (z \cdot z)$ and terminate.

(For completeness, we can assume that if $n < n'$ then B'_n simply discards its input and outputs 0. These cases do not matter in our analysis.) Note that B' is an efficient family of circuits. Define the circuit family $C' = (C'_n)_{n\in\mathbb{N}}$ analogously with respect to the family C. Note that B'_n outputs the bit b' if and only if its execution of B_n outputs the bit $b' \oplus (z \cdot z)$ and similarly for C'_n. Hence, since D_n is uniformly distributed, we have that $v(n)$ can also be expressed as

$$\mathop{\mathbb{E}}_{\substack{x_B,x_C\\y_B,y_C}} \langle g_n(x_B)\cdot y_B, g_n(x_C)\cdot y_C| \, (B'_n \otimes C'_n)(y_B \otimes \sigma_n^{x_B,x_C} \otimes y_C) \, |g_n(x_B)\cdot y_B, g_n(x_C)\cdot y_C\rangle \quad (4.69)$$

where $\sigma_n^{x_B,x_C} = x_B \otimes A_n(\rho_n) \otimes x_C$, x_B and x_C are sampled independently and uniformly at random from $\{0,1\}^{d(n)}$, and y_B and y_C similarly from $\{0,1\}^{c(n)}$. To show that v is at most negligibly more than $\frac{1}{2}$, it now suffices by Lemma 4.1 to show that for all pairs (B'', C'') of efficient $(d + q_B, c)$- and $(q_C + d, c)$-circuit families, respectively, we have that

$$n \mapsto \mathop{\mathbb{E}}_{x_B,x_C} \langle g_n(x_B), g_n(x_C)| \, (B''_n \otimes C''_n)(\sigma_n^{x_B,x_C}) \, |g_n(x_B), g_n(x_C)\rangle \quad (4.70)$$

is a negligible map when x_B and x_C are sampled independently and uniformly at random from $\{0,1\}^{d(n)}$.

Since (G, E) is a secure copy-protection scheme and 2^{-c} is negligible, we have that

$$\mathop{\mathbb{E}}_{k,x_B,x_C} \langle f_n(k, x_B), f_n(k, x_C)| \, (B''_n \otimes C''_n)(x_B \otimes A_n(G_n(k)) \otimes x_C) \, |f_n(k, x_B), f_n(k, x_C)\rangle \quad (4.71)$$

is negligible in n when x_B and x_C are independently and uniformly sampled at random from $\{0,1\}^{d(n)}$ and similarly for k from $\{0,1\}^{\kappa(n)}$. So, by Theorem 4.10 and the fact that

$\kappa \in \omega(\log)$,[16] we can fix the keys k to be the elements of the exponential-time ingenerable sequence $(k_n)_{n\in\mathbb{N}}$ and obtain that

$$\mathop{\mathbb{E}}_{x_B,x_C} \langle f_n(k_n,x_B), f_n(k_n,x_C)| \, (B''_n \otimes C''_n)(x_B \otimes A_n(G_n(k_n)) \otimes x_C) \, |f_n(k_n,x_B), f_n(k_n,x_C)\rangle \tag{4.72}$$

is also negligible in n when x_B and x_C are sampled independently and uniformly at random from $\{0,1\}^{d(c)}$. Recalling that $f_n(k_n,\cdot) = g_n$, this is precisely what is needed.

Note that we neglect in this proof to give an explicit construction of a $(\kappa,1)$-circuit which models the complete set-up and attack of the copy-protection scheme on input of a given key k, as is technically required to apply Theorem 4.10. However, this construction would simply follow the same ideas as the ones presented in the proofs of Theorem 4.14 and Lemma 4.38: The attacking circuits from B'' and C'' are placed between an initial set-up circuit, which creates and distributes the program state and challenges, and a final referee circuit, which receives the guesses and outputs 1 if and only if they are correct. This composition would yield the necessary circuits.

Recalling that $\sigma_n^{x_B,x_C} = x_B \otimes A_n(G_n(k')) \otimes x_C$ is sufficient to complete the proof. ∎

4.5.5 Instantiating a Language with Uncloneable Advice

Our construction of a language with uncloneable advice in the previous section did not consider an important question: Is it possible to copy-protect a sequence of functions satisfying the necessary conditions? The following theorem gives one positive answer.

Theorem 4.47. If there exists a correct and secure copy-protection scheme for a pseudorandom function $f' = (f'_n)_{n\in\mathbb{N}}$ with outputs of super-logarithmic length, then there exists a sequence of maps $f = (f_n)_{n\in\mathbb{N}}$ satisfying the assumptions of Theorem 4.46.

Recall that a pseudorandom function $f = (f_n : \{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{c(n)})_{n\in\mathbb{N}}$ is a collection of keyed functions such that $f_n(k,\cdot)$, for uniformly random k, is computationally indistinguishable from a truly random function when given only oracle access. We will not actually use the property of being pseudorandom in this work, but they have been the subject of other works in quantum copy-protection.

Before proving this theorem, we recall a result from [CLLZ21]. It establishes sufficient conditions, which we do not formalize, for the copy-protection of a certain construction of

[16] If $\kappa \notin \omega(\log)$, then $2^{-\kappa}$ is non-negligible. Thus, a triplet of adversaries can break the assumed security of the copy-protection scheme (G,E) by sampling a key uniformly at random and using it, with the honest generation and evaluation algorithms, to correctly evaluate the function correctly with non-negligible probability. Indeed, they will have a non-negligible probability of having guessed the correct key.

pseudorandom functions presented in that work. By the previous theorem, these are also sufficient conditions for the existence of a language with uncloneable advice.

Theorem 4.48 ([CLLZ21])**.** Assuming the existence of post-quantum indistinguishable obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, there exists a secure and correct copy-protection scheme for a specific construction of pseudorandom functions where the length of the keys, inputs, and outputs are non-constant polynomials.

Note that the PRF f' assumed in Theorem 4.47 does not necessarily satisfy the first condition of Theorem 4.46, i.e., there is no guarantee that $d(n)+c(n)$ will eventually always be strictly smaller than n. In fact, this is never satisfied for the PRF considered in [CLLZ21]. Thus, we need to show that we can "slow down" the growth of this value, while keeping the correctness and, more importantly, the security of the scheme. The way we will do this is by repeating certain elements of the sequence $f' = (f'_n)_{n\in\mathbb{N}}$ multiple times in succession to obtain a new sequence of maps $f = (f_n)_{n\in\mathbb{N}}$. This will be parametrized by a re-indexing map $\gamma : \mathbb{N} \to \mathbb{N}$ which will, in general, *not* be injective and where $f_n = f'_{\gamma(n)}$. In some sense, the map γ "slows down" the growth of the security parameter for the maps. The following lemma gives conditions for this transformation to preserve correctness and security.

Lemma 4.49. Let (G, E) be a copy-protection scheme for a sequence of maps f, as defined in Equation 4.35, which is correct and secure and where 2^{-c} is a negligible map. Let $\gamma : \mathbb{N} \to \mathbb{N}$ be a non-decreasing computable map in $\Omega(\lambda^{r_l})$ and $\mathcal{O}(\lambda^{r_u})$ for some $r_l, r_u \in \mathbb{R}^+$. Let $G^\gamma = (G^\gamma_\lambda = G_{\gamma(\lambda)})_{\lambda\in\mathbb{N}}$, $E^\gamma = (E^\gamma_\lambda = E_{\gamma(\lambda)})_{\lambda\in\mathbb{N}}$, and $f^\gamma = (f^\gamma_\lambda = f_{\gamma(\lambda)})_{\lambda\in\mathbb{N}}$. Then, (G^γ, E^γ) is a copy-protection scheme for f^γ which is correct and secure.

Moreover, $2^{-c\circ\gamma}$ is also negligible, as it is simply $2^{-c} \circ \gamma$ and $\gamma \in \Omega(\lambda^{r_l})$.

Proof: First, note that since γ is efficiently computable and upper-bounded by a polynomial, as it is in $\mathcal{O}(\lambda^{r_u})$, then G^γ and E^γ are efficient circuit families. The correctness of (G^γ, E^γ) follows trivially from the correctness of (G, E) and the fact that the map $\eta \circ \gamma$ is negligible if η is negligible as $\gamma \in \Omega(\lambda^{r_l})$. We move on to showing the security.

Let (A', B', C') be an attack against (G^γ, E^γ). We construct an attack (A, B, C) against the original scheme (G, E) as follows. For completion, assume we have an existing attack $(\tilde{A}, \tilde{B}, \tilde{C})$ against (G, E). The specifics of this attack do not matter for this proof; it will only be used to fill in some gaps and ensure that all of our objects are well defined. We use n to index the security parameter of (A', B', C') and λ to index the one of (A, B, C).

For all $\lambda \in \mathbb{N}$, define A_λ as follows:

- If $\gamma^{-1}(\lambda) = \varnothing$, then $A_\lambda = \tilde{A}_\lambda$.

- If $\gamma^{-1}(\lambda) \neq \varnothing$, then A_λ is a circuit which samples uniformly at random an $n' \in \gamma^{-1}(\lambda)$, executes $A'_{n'}$ and then gives both B_λ and C_λ a copy of n' in addition to their respective share of the output of $A'_{n'}$.

For all $\lambda \in \mathbb{N}$, we define B_λ as follows:

- If $\gamma^{-1}(\lambda) = \varnothing$, then $B_\lambda = \tilde{B}_\lambda$.
- If $\gamma^{-1}(\lambda) \neq \varnothing$, then B_λ is a circuit which reads the n' value received from A_λ and then executes the $B'_{n'}$ circuit on the remainder of the input received from A_λ.

The C_λ circuits are defined analogously to the B_λ circuits. Before proceeding, we should ensure that the (A, B, C) defined here is a triplet of *efficient* circuit families.

First, we show that the set $\gamma^{-1}(\lambda)$ is efficiently computable from λ, in the sense that there exists a polynomial-time Turing machine which, on input of 1^λ, outputs an encoding of $\gamma^{-1}(\lambda)$.

By our assumptions on the map γ, there exists a n_γ and $k_u, k_l \in \mathbb{R}^+$ such that $n \geq n_\gamma$ implies that $k_l n^{r_l} \leq \gamma(n) \leq k_u n^{r_u}$. Thus, for any $n \geq n_\gamma$,

$$n \in \gamma^{-1}(\lambda) \implies k_l \lambda^{r_l} \leq \gamma(n) = \lambda \leq k_u n^{r_u} \implies (\lambda/k_u)^{\frac{1}{r_u}} \leq n \leq (\lambda/k_l)^{\frac{1}{r_l}}. \tag{4.73}$$

This implies that there are at most $\max\{n_\gamma, (\lambda/k_l)^{\frac{1}{\lambda_\ell}}\} \in \mathcal{O}(\gamma^{\frac{1}{r_\ell}})$ values in $\gamma^{-1}(\lambda)$. Specifically, the possible elements of $\gamma^{-1}(\lambda)$ are the non-negative integers which are no greater than $\max\{n_\gamma, (\lambda/k_l)^{\frac{1}{r_\ell}}\}$. Thus, one efficient way to compute $\gamma^{-1}(\lambda)$ is to simply compute γ on each of these candidate integers and verify if the result is n. Since each computation can be done in polynomial-time and there are at most a polynomial number of candidate values, this is an efficient computation of $\gamma^{-1}(\lambda)$. Note that this reasoning also implies that $\left|\gamma^{-1}(\lambda)\right|$ is upper bounded by a polynomial in λ.

It then follows that the circuit families (A, B, C) described above are efficient. Indeed, checking if $\gamma^{-1}(\lambda)$ is empty or not can be done efficiently. If it is empty, A_λ, B_λ and C_λ are simply circuits which are already assumed to be from an efficient family. If $\gamma^{-1}(\lambda)$ is not empty, then A_λ, B_λ, and C_λ is simply $\left|\gamma^{-1}(\lambda)\right|$ circuits in parallel with a minimal overhead to route the inputs to the right circuit. Since $\left|\gamma^{-1}(\lambda)\right|$ is polynomially bounded, this remains efficient.

Thus, by the security of (G, E) and the fact that 2^{-c} is negligible, there exists a negligible function η such that $\eta(\lambda)$ upper bounds

$$\mathop{\mathbb{E}}_{\substack{k \leftarrow \{0,1\}^{\kappa(\lambda)} \\ x_B, x_C \leftarrow \{0,1\}^{d(\lambda)}}} \langle f_\lambda(k, x_B), f_\lambda(k, x_C)| \left(B_\lambda \otimes C_\lambda\right) \left(x_B \otimes A_\lambda(G_\lambda(k)) \otimes x_C\right) |f_\lambda(k, x_B), f_\lambda(k, x_C)\rangle \tag{4.74}$$

Let $v(\lambda)$ denote the left-hand side of the above. By our construction, the attack (A, B, C) averages multiple instances of the attack (A', B', C') when possible. More precisely, if we let $v'(n)$ denote the success probability of the attack (A', B', C') for the security parameter n, we have that

$$\mathop{\mathbb{E}}_{n\in\gamma^{-1}(\lambda)} v'(n) = v(\lambda) \leq \eta(\lambda) \tag{4.75}$$

if $\gamma^{-1}(\lambda) \neq \varnothing$. In particular, this implies that $v'(n) \leq \left|\gamma^{-1}(\gamma(n))\right| \cdot \eta \circ \gamma(n)$. By our assumption on the map γ, $\gamma(n)$ is upper bounded by a polynomial in n, which by our previous remarks also implies that $\left|\gamma^{-1}(\gamma(n))\right|$ is upper bounded by a polynomial in n. Since γ is in $\Omega(n^{r_l})$ and is non-decreasing, we have that $\eta \circ \gamma$ is negligible. It follows that $v'(n)$ is negligible, which is the desired result. ∎

While the previous lemma gives sufficient condition for when a "γ slowdown" maintains the security and correctness guarantees of copy-protected functions, it does not establish the existence of a suitable γ. This is done, implicitly, by the following lemma.

Lemma 4.50. Let $g : \mathbb{N} \to \mathbb{N}$ be a map such that $g \in \mathcal{O}(n^d)$ for a $d \in \mathbb{N}^+$. Then, there exists an efficiently computable non-decreasing map $f : \mathbb{N} \to \mathbb{N}$ such that $g \circ f$ is eventually strictly smaller than n and $f \in \Omega(\lambda^r) \cap \mathcal{O}(\lambda^r)$ for a $r \in \mathbb{R}^+$.

Proof: Let $k, n_g \in \mathbb{N}$ be such that $n \geq n_g \implies g(\lambda) \leq kn^d$. Now, let f be defined by

$$f(n) = \begin{cases} 0 & \text{if} \quad n < \sqrt[d]{k} n_g^{2d} \\ \left\lfloor \frac{1}{\sqrt[d]{k}} n^{\frac{1}{2d}} \right\rfloor & \text{else.} \end{cases} \tag{4.76}$$

Trivially, f is non-decreasing and efficiently computable. Thus, for all $n > \max\{\sqrt[d]{k} n_g^{2d}, 1\}$,

$$g \circ f(n) = g\left(\left\lfloor \frac{1}{\sqrt[d]{k_u}} n^{\frac{1}{2d}} \right\rfloor\right) \leq n^{\frac{1}{2}} < n. \tag{4.77}$$

It now remains to show that there exists an $r \in \mathbb{R}^+$ such that $f \in \Omega(\lambda^r)$. Note that for all sufficiently large values of λ we have that $f(\lambda) \geq \frac{1}{\sqrt[d]{k}} \lambda^{\frac{1}{2d}} - 1 \geq \frac{1}{2\sqrt[d]{k}} \lambda^{\frac{1}{2d}}$. Thus, $f \in \Omega(\lambda^{\frac{1}{2d}})$. We similarly find that $f \in \mathcal{O}(\lambda^{\frac{1}{2d}})$. ∎

Pulling these lemmas together, we can prove Theorem 4.47.

Proof: Let $f' = (f'_n : \{0,1\}^{\kappa(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{c(n)})_{n\in\mathbb{N}}$ be the copy-protected PRF. We know that $d + c \in \mathcal{O}(n^r)$ for some $r \in \mathbb{R}^+$ else it would not be possible to efficiently compute this PRF. Then, by Lemma 4.50, there exists a non-decreasing efficiently computable γ in both $\Omega(\lambda^{\frac{1}{2d}})$ and $\mathcal{O}(\lambda^{\frac{1}{2d}})$ such that $(c+d)\circ\gamma(n)$ is eventually strictly smaller

than n. Thus, by applying the transformation of Lemma 4.49 with this γ to f', we obtain a sequence of maps f satisfying the assumptions of Theorem 4.46. ∎

Chapter 5

Conclusion

In this thesis we have studied various cryptographic applications of the no-cloning and information-disturbance principles offered by quantum mechanics. Our contributions touched upon established notions, contemporary ideas, as well as novel topics.

We began in Chapter 2 by revisiting Gottesman's twenty-year old idea of tamper-evident encryption. We helped situate this notion within the landscape of cryptography by demonstrating various generic implications and separations between it and other concepts. In particular, we answered a question left open by Gottesman by showing that tamper evidence implied encryption.

In Chapter 3, we participated in the ongoing effort to study uncloneable functionalities via copy-protection and secure software leasing. Our main contribution was to construct, for the first time, schemes achieving these security notions in an information-theoretic setting.

Finally, in Chapter 4, we broke new ground by demonstrating the feasibility of copy-protecting *advice*. We achieved this by defining a complexity class which captures this notion, **neglQP**/upoly, and proving that it is non-empty via the use of our novel concept of ingenerable sequences. We hope that this will give another lens under which to study the notion of uncloneability.

We believe that uncloneable cryptography — roughly defined as cryptography where the security notions can only be achieved due to the no-cloning principle — will remain a fruitful area of research for the near future.

As we have implicitly shown, there is work to be done in identifying relations amongst uncloneable cryptographic primitives, as well as with non-uncloneable primitives. There is work to be done in constructing additional schemes with better performance profiles or requiring fewer assumptions for established notions of uncloneability. And there is work to be done in pushing the frontiers of which cryptographic and information processing concepts can be made uncloneable in one sense or another.

5.1 Open Questions

We conclude with a few additional open questions, collected under two themes, which did not appear in prior chapters. These questions touch on subjects pertaining to more than one chapter and answering them was beyond the scope of our endeavours. However, in some cases, we sketch potential avenues for solutions. We believe inquiries along these lines to be of foundational importance for the further development of uncloneable cryptography.

Theme 1: What is the proper way to pick challenge distributions and trivial winning probabilities for quantum copy-protection and quantum advice? Security for a quantum copy-protection scheme and uncloneable advice is defined with the help of a *trivial winning probability* and a *challenge distribution*. The challenge distribution is a joint probability distribution on the functions considered by the scheme — which reduces to a single possibility in the case of advice — and pairs of inputs to these functions. There have been many proposals in the literature on how to select these parameters. While sensible options have been put forward, we believe it remains an open question to provide a rigorous theoretical justification for these choices.

Recall that adversaries for these schemes are given as triplets: one splitting adversary who distributes the honest program state amongst two guessing adversaries who try to use it to compute the function on the challenge input they receive.

Concerning the trivial winning probability, we took the approach in Chapter 4 to define it using the weakest viable adversaries: the splitting adversary gives the advice state to one guessing adversary, who we assume will use it to correctly compute their answer, while the other guessing adversary picks their answer uniformly at random. This is also the approach taken by [ALL+21]. More fine-grained approaches, such as the ones of [AKL23] and [CMP24], allow the splitting adversary to make the optimal choice, based on the challenge distribution, in determining which guessing adversary receives the unmodified program state and which one does not. They also allow the guessing adversary who did not receive the program state to provide an answer not picked uniformly at random. This is essentially the approach we took in Chapter 3, but our challenge distribution was such that this trivial winning probability was equivalent to the one obtained by guessing at random.

The trivial winning probabilities of [AKL23] and [CMP24] do not consider the complexity or computational cost for the splitting adversary to determine the optimal choice of which guessing adversary should receive the program state. This leads to a few interesting questions for definitions of this sort. When can this optimal decision be made efficiently? Are there distributions where this trivial winning probability cannot actually be achieved by efficient adversaries? Does this yield schemes secure only in a pathological sense?

A related question that should also be answered is whether the splitting adversary should be allowed to *use* the program state — perhaps modeled as queries to an oracle for the encoded function — before making their choice.

Pushing this last idea further, we could define the trivial winning probability by considering the maximal winning probability of any triplet of efficient adversaries in the following game played with an oracle instead of an actual program state: A triplet (f, x_B, x_C) is sampled according to the challenge distribution and the splitting adversary is given oracle access to f. The splitting adversary then outputs $L \in \{B, C\}$ as well as any additional information σ. The guessing adversary corresponding to L is then given σ and x_L and wins if and only if they correctly output $f(x_L)$.

Conceptually, this perspective consists in identifying an uncloneable program state with an oracle for the encoded function which can be freely queried, but is only accessible to one party at a time. The non-L guessing adversary is assumed to have been given access to the oracle and hence can correctly compute the answer to their challenge.

Interestingly, adopting this model might require us to recontextualize Aaronson's original result that learnable functions cannot be copy-protected [Aar09]. Instead, it would state that copy-protecting learnable functions is a vacuous task since the corresponding trivial winning probability is 1.

We now move on to the question of challenge distributions. While every challenge distribution yields a *valid* security game, it is unclear which, if any, is the *right* challenge distribution to choose. So far, works have considered distributions in a more-or-less *ad hoc* fashion. Distributions appear to be picked because they capture, in one sense or another, some interesting behaviour of the functions under consideration and admit schemes for which a proof of security can be obtained. It would be beneficial for the theory of copy-protection if better justifications for the choice of challenge distribution could be given.

A starting point would be to identify criteria allowing us to state that security with respect to one distribution implies security with respect to another. This would yield a preorder on challenge distributions allowing us to identify which distributions yield stronger security notions, and which yield weaker notions. Proving security with respect to maximal elements of this preorder would be a natural goal. Ideally, we would then prove security with respect to *all* maximal elements, implying security with respect to *all* distributions.

Notable work in this direction has been done by Ananth, Kaleogu, and Liu, who showed that under certain conditions security against independently sampled inputs implies security against identically sampled inputs [AKL23]. However, a limitation of their result is that it only holds when the trivial winning probability is negligible. In particular, this does not occur in the context of uncloneable advice where the trivial winning probability is one half.

Theme 2: Can we formalize the idea that similar security notions occur at various levels of the uncloneability hierarchy? At the beginning of Chapter 3, we highlighted the existence of a hierarchy of resources which can be made uncloneable: authenticity,[1] information, and functionality. As we have seen throughout this work, there are often many distinct ways to formalize a notion of uncloneability for each of these types of resources.

Sattath has observed that "tamper-evident encryption is to uncloneable encryption what [secure software leasing] is to copy-protection" [Sat23]. We largely agree with this assessment, but we would have perhaps used revocation instead of tamper evidence in this analogy. However, this may be a small quibble in light of our result from Chapter 2 that tamper evidence and revocation are extremely closely related security notions.

Nonetheless, Sattath's remark highlights that certain security notions appear to be applicable across the layers of the uncloneability hierarchy. Both quantum copy-protection and uncloneable encryption prevent adversaries from proliferating a resource they have gained from the honest parties. More precisely, they prevent the adversaries from sharing an *operational capability granted by the resource*: the ability to learn the plaintext when given the key in the first, or the ability to correctly compute a function in the second. Tamper-evidence, revocation, certified deletion, and secure software leasing, for their part, all allow an honest party to detect an attempt by an adversary to covertly keep an operational capability which they are no longer intended to have. This is done via a mechanism allowing the honest party to verify the proper return of a resource. We would also add *quantum money*, an element from the lowest rung of the uncloneability hierarchy, to this list of primitives allowing such a verification.

This leaves us to wonder if this observation can be properly formalized and, if so, how? This question may be appropriate to study in the "cloning games" model [AKL23] or some suitable generalization of this framework.

Tangentially, this also raises the question of whether it would be possible to construct a secure software leasing scheme for quantum advice. Clearly, uncloneable advice for a problem in **neglQP**/upoly cannot be efficiently *generated*. However, would it be possible to efficiently *verify* its return? We note that such a verification may require the verifier to have access to some information unknown to the user of the advice state. Would it be possible for the verifier — while in possession of the advice state — to efficiently generate this information before handing the state over to the user?

[1] We could have also labelled this layer of the hierarchy as "tokens", as the objects here carry no meaningful information beyond the fact that they are authentic.

Bibliography

[Aar05] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, vol. 1, no. 1, pp. 1–28, 2005.
doi:10.4086/toc.2005.v001a001

[Aar09] S. Aaronson. Quantum copy-protection and quantum money. In *24th Annual Conference on Computational Complexity (CCC 2009)*, pp. 229–242, 2009.
doi:10.1109/CCC.2009.42

[Aar16a] S. Aaronson. The complexity of quantum states and transformations: From quantum money to black holes. Eprint arXiv:1607.05256 [quant-ph], 2016.
doi:10.48550/arXiv.1607.05256

[Aar16b] S. Aaronson. The no-cloning theorem and the human condition. After-diner talk given at the QCRYPT 2016 conference, 2016.
https://www.scottaaronson.com/blog/?p=2903

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ABF+16] G. Alagic, A. Broadbent, B. Fefferman, T. Gagliardoni, C. Schaffner, and M. St. Jules. Computational security of quantum encryption. In *Information Theoretic Security (ICITS 2016)*, pp. 47–71, 2016.
doi:10.1007/978-3-319-49175-2_3

[AC02] M. Adcock and R. Cleve. A quantum Goldreich-Levin theorem with cryptographic applications. In *19th Annual Symposium on Theoretical Aspects of Computer Science — STACS 2002*, pp. 323–334, 2002.
doi:10.1007/3-540-45841-7_26

[AC12] S. Aaronson and P. Christiano. Quantum money from hidden subspaces. In *STOC '12: Proceedings of the fourty-fourth ACM symposium on Theory of computing*, pp. 41–60, 2012.
doi:10.1145/2213977.2213983

[AD14] S. Aaronson and A. Drucker. A full characterization of quantum advice. *SIAM Journal on Computing*, vol. 43, no. 3, pp. 1131–1183, 2014.
doi:10.1137/110856939

[AKL23] P. Ananth, F. Kaleoglu, and Q. Liu. Cloning games: A general framework for unclonable primitives. In *Advances in Cryptology – CRYPTO 2023*, vol. 5, pp. 66–98, 2023.
doi:10.1007/978-3-031-38554-4_3

[AL21] P. Ananth and R. L. La Placa. Secure software leasing. In *Advances in Cryptology — EUROCRYPT 2021*, vol. 2, pp. 501–530, 2021.
doi:10.1007/978-3-030-77886-6_17

[ALL+21] S. Aaronson, J. Liu, Q. Liu, M. Zhandry, and R. Zhang. New approaches for quantum copy-protection. In *Advances in Cryptology — CRYPTO 2021*, vol. 1, pp. 526–555, 2021.
doi:10.1007/978-3-030-84242-0_19

[AM17] G. Alagic and C. Majenz. Quantum non-malleability and authentication. In *Advances in Cryptology — CRYPTO 2017*, vol. 2, pp. 310–341, 2017.
doi:10.1007/978-3-319-63715-0_11

[AMTdW00] A. Ambainis, M. Mosca, A. Tapp, and R. de Wolf. Private quantum channels. In *41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*, pp. 547–553, 2000.
doi:10.1109/SFCS.2000.892142

[Bar20] E. Barker. Recommendation for key management: Part 1 – general. Technical Report SP 800-57 Part 1 Rev. 5, National Institute of Standards and Technology, 2020.
doi:10.6028/NIST.SP.800-57pt1r5

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computers, Systems and Signal Processing*, pp. 175–179, 1984.

[BCG+02] H. Barnum, C. Crépeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of quantum messages. In *43rd Annual Symposium on Foundations of Computer Science (FOCS 2002)*, pp. 449–485, 2002.
doi:10.1109/SFCS.2002.1181969

[BDE+98] D. Bruß, D. P. DiVincenzo, A. Ekert, C. A. Fuchs, C. Macchiavello, and J. A. Smolin. Optimal universal and state-dependent quantum cloning. *Physical Review A*, vol. 57, no. 4, p. 2368, 1998.
doi:10.1103/PhysRevA.57.2368

[Bel02] M. Bellare. A note on negligible functions. *Journal of Cryptology*, vol. 15, no. 4, pp. 271–284, 2002.
doi:10.1007/s00145-002-0116-x

[BGPS24] A. Broadbent, A. B. Grilo, S. Podder, and J. Sikora. The role of piracy in quantum proofs. Eprint arXiv:2410.02228 [quant-ph], 2024.
doi:10.48550/arXiv.2410.02228

[BGT21] A. Bouland and T. Giurgică-Tiron. Efficient universal quantum compilation: An inverse-free Solovay-Kitaev algorithm. Eprint arXiv:2112.02040 [quant-ph], 2021.
doi:10.48550/arXiv.2112.02040

[BH96] V. Bužek and M. Hillery. Quantum copying: Beyond the no-cloning theorem. *Physical Review A*, vol. 54, no. 3, pp. 1844–1852, 1996.
doi:10.1103/PhysRevA.54.1844

[BI20] A. Broadbent and R. Islam. Quantum encryption with certified deletion. In *Theory of Cryptography (TCC 2020)*, vol. 3, pp. 92–122, 2020.
doi:10.1007/978-3-030-64381-2_4

[Bil95] P. Billingsley. *Probability and Measure*. John Wiley & Sons, 3rd edition, 1995.

[BJL+21a] A. Broadbent, S. Jeffery, S. Lord, S. Podder, and A. Sundaram. Secure software leasing without assumptions. In *Theory of Cryptography (TCC 2021)*, vol. 1, pp. 90–120, 2021.
doi:10.1007/978-3-030-90459-3_4

[BJL+21b] A. Broadbent, S. Jeffery, S. Lord, S. Podder, and A. Sundaram. Secure software leasing without assumptions. Eprint arXiv:2101.12739 [quant-ph], 2021.
doi:10.48550/arXiv.2101.12739

[BK23] J. Bartusek and D. Khurana. Cryptography with certified deletion. In *Advances in Cryptology — CRYPTO 2023*, vol. 5, pp. 192–223, 2023.
doi:10.1007/978-3-031-38554-4_7

[BKL23] A. Broadbent, M. Karvonen, and S. Lord. Uncloneable quantum advice. Eprint arXiv:2309.05155 [quant-ph], 2023.
doi:10.48550/arXiv.2309.05155

[BKL24] A. Broadbent, M. Karvonent, and S. Lord. Uncloneable quantum advice. *IACR Communications in Cryptography*, vol. 1, no. 3, 2024.
doi:10.62056/abe0fhbmo

[BL20] A. Broadbent and S. Lord. Uncloneable quantum encryption via oracles. In *15th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC 2020)*, art. 4, 2020.
doi:10.4230/LIPIcs.TQC.2020.4

[Bra85] G. Brassard. Crusade for a better notation. *ACM SIGACT News*, vol. 17, no. 1, pp. 60–64, 1985.
doi:10.1145/382250.382808

[Bra05] G. Brassard. Brief history of quantum cryptography: A personal perspective. In *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*, pp. 19–23, 2005.
doi:10.1109/ITWTPI.2005.1543949

[BS16] A. Broadbent and C. Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, vol. 78, no. 1, pp. 351–382, 2016.
doi:10.1007/s10623-015-0157-4

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, vol. 96, no. 5, pp. 1411–1473, 1997.
doi:10.1137/S0097539796300921

[CBD+24] A. S. Cacciapuoti, A. Broadbent, E. Diamanti, J. Romero, and S. Wehner. The quantum internet: Principles, protocols, and architectures. *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 7, pp. 1719–1722, 2024.
doi:10.1109/JSAC.2024.3379106

[CJL+16] L. Chen *et al.* Report on post-quantum cryptography. Technical Report IR 8105, National Institute of Standards and Technology, 2016.
doi:10.6028/NIST.IR.8105

[CLLW16] R. Cleve, D. Leung, L. Liu, and C. Wang. Near-linear constructions of exact unitary 2-designs. *Quantum Information & Computation*, vol. 16, no. 9-10, pp. 721–756, 2016.
doi:10.26421/QIC16.9-10-1

[CLLZ21] A. Coladangelo, J. Liu, Q. Liu, and M. Zhandry. Hidden cosets and applications to unclonable cryptography. In *Advances in Cryptology — CRYPTO 2021*, pp. 556–584, 2021.
doi:10.1007/978-3-030-84242-0_20

[CMP24] A. Coladangelo, C. Majenz, and A. Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *Quantum*, vol. 8, p. 1330, 2024.
doi:10.22331/q-2024-05-02-1330

[CRW19] X. Coiteux-Roy and S. Wolf. Proving erasure. In *IEEE International Symposium on Information Theory—ISIT 2019*, pp. 832–836, 2019.
doi:10.1109/ISIT.2019.8849661

[DCEL09] C. Dankert, R. Cleve, J. Emerson, and E. Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, vol. 80, no. 1, p. 012304, 2009.
doi:10.1103/PhysRevA.80.012304

[Die82] D. Dieks. Communication by EPR devices. *Physics Letters A*, vol. 92, no. 6, pp. 271–272, 1982.
doi:10.1016/0375-9601(82)90084-6

[Dir39] P. A. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 35, no. 3, pp. 416–418, 1939.
doi:10.107/S0305004100021162

[DN06] C. M. Dawson and M. A. Nielsen. The Solovay-Kitaev algorithm. *Quantum Information & Computation*, vol. 6, no. 1, pp. 81–95, 2006.
doi:10.26421/QIC6.1-6

[DNS12] F. Dupuis, J. B. Nielsen, and L. Salvail. Actively secure two-party evaluation of any quantum operation. In *Advances in Cryptology — CRYPTO 2012*, pp. 794–811, 2012.
doi:10.1007/978-3-642-32009-5_46

[DS18] Y. Dulek and F. Speelman. Quantum ciphertext authentication and key recycling with the trap code. In *13th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC 2018)*, art. 1, 2018.
doi:10.4230/LIPIcs.TQC.2018.1

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review Letters*, vol. 47, no. 10, pp. 777–780, 1935.
doi:10.1103/physrev.47.777

[FdG99] C. A. Fuchs and J. V. de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1216–1227, 1999.
doi:10.1109/18.761271

[FS17] S. Fehr and L. Salvail. Quantum authentication and encryption with key recycling. In *Advances in Cryptology — EUROCRYPT 2017*, pp. 311–338, 2017.
doi:10.1007/978-3-319-56617-7_11

[Fuc96] C. A. Fuchs. Information gain vs. state disturbance in quantum theory. Eprint arXiv:quant-ph/9611010, 1996.
doi:10.48550/arXiv.quant-ph/9611010

[Gao15] J. Gao. Quantum union bounds for sequential projective measurements. *Physical Review A*, vol. 92, no. 5, p. 052331, 2015.
doi:10.1103/PhysRevA.92.052331

[GMR23] V. Goyal, G. Malavolta, and J. Raizes. Unclonable commitments and proofs. IACR Cryptology ePrint Archive, Paper 2023/1538, 2023.
https://eprint.iacr.org/2023/1538

[Got03] D. Gottesman. Uncloneable encryption. *Quantum Information & Computation*, vol. 3, no. 6, pp. 581–602, 2003.
doi:10.26421/QIC3.6-2

[Got11] D. Gottesman. Answer to "Rigorous security proof for Wiesner's quantum money?". Theoretical Computer Science Stack Exchange, 2011.
https://cstheory.stackexchange.com/q/11364

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the twenty-eighth ACM symposium on Theory of computing*, pp. 212–219, 1996.
doi:10.1145/237814.237866

[GYZ17] S. Garg, H. Yuen, and M. Zhandry. New security notions and feasibility results for authentication of quantum data. In *Advances in Cryptology — CRYPTO 2017*, vol. 2, pp. 342–371, 2017.
doi:10.1007/978-3-319-63715-0_12

[Her82] N. Herbert. FLASH —– A superluminal communicator based upon a new kind of quantum measurement. *Foundations of Physics*, vol. 12, no. 12, pp. 1171–1179, 1982.
doi:10.1007/BF00729622

[HMNY21] T. Hiroka, T. Morimae, R. Nishimaki, and T. Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In *Advances in Cryptology — ASIACRYPT 2021*, pp. 606–636, 2021.
doi:10.1007/978-3-030-92062-3_21

[JJUW11] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP = PSPACE. *Journal of the ACM*, vol. 58, no. 6, p. 30, 2011.
doi:10.1145/2049697.2049704

[JK23] R. Jawale and D. Khurana. Unclonable non-interactive zero-knowledge. IACR Cryptology ePrint Archive, Paper 2023/1532, 2023.
https://eprint.iacr.org/2023/1532

[Kit97] A. Y. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, vol. 52, no. 6, pp. 1191–1249, 1997.
doi:10.1070/RM1997v052n06ABEH002155

[KL15] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. CRC Press, 2nd edition, 2015.

[Knu76] D. E. Knuth. Big omicron and big omega and big theta. *ACM SIGACT News*, vol. 8, no. 2, pp. 18–24, 1976.
doi:10.1145/1008328.1008329

[KNY21] F. Kitagawa, R. Nishimaki, and T. Yamakawa. Secure software leasing from standard assumptions. In *Theory of Cryptography (TCC 2021)*, vol. 1, pp. 31–61, 2021.
doi:10.1007/978-3-030-90459-3_2

[KT22] S. Kundu and E. Y.-Z. Tan. Device-independent uncloneable encryption. Eprint arXiv:2210.01058 [quant-ph], 2022.
doi:10.48550/arXiv.2210.01058

[KT23] S. Kundu and E. Y.-T. Tan. Composably secure device-independent encryption with certified deletion. *Quantum*, vol. 7, no. 1047, 2023.
doi:10.22331/q-2023-07-06-1047

[Lan96] R. Landauer. The physical nature of information. *Physics Letters A*, vol. 217, no. 4-5, pp. 188–193, 1996.
doi:10.1016/0375-9601(96)00453-7

[LFM+24] M. Lemus, R. Faleiro, P. Mateus, N. Paunković, and A. Souto. Quantum Kolmogorov complexity and quantum correlations in deterministic-control quantum Turing machines. *Quantum*, vol. 8, p. 1230, 2024.
doi:10.22331/q-2024-01-18-1230

[Lor24] S. Lord. Relating quantum tamper-evident encryption to other cryptographic notions. Eprint arXiv:2411.02742 [quant-ph], 2024.
doi:10.48550/arXiv.2411.02742

[LŠ20] D. Leermaker and B. Škorić. Qubit-based unclonable encryption with key recycling. Eprint arXiv:2004.04084 [quant-ph], 2020.
doi:10.48550/arXiv.2004.04084

[Lut10] A. Lutomirski. An online attack against Wiesner's quantum money. Eprint arXiv:1010.0256 [quant-ph], 2010.
doi:10.48550/arXiv.2004.04084

[LV19] M. Li and P. Vitányi. *An Intorduction to Kolmogorov Complexity and Its Applications*. Springer, fourth edition, 2019.
doi:10.1007/978-3-030-11298-1

[Mae13] R. Maes. *Physically Unclonable Functions*. Springer, 2013.
doi:10.1007/978-3-642-41395-7

[MS10] M. Mosca and D. Stebila. Quantum coins. In *Error-Correcting Codes, Finite Geometries and Cryptography*, pp. 35–47. American Mathematical Society, 2010.

[MVW13] A. Molina, T. Vidick, and J. Watrous. Optimal counterfeiting attacks and generalizations for Wiesner's quantum money. In *Theory of Quantum Computation, Communication, and Cryptography (TQC 2012)*, pp. 45–64, 2013. doi:10.1007/978-3-642-35656-8_4

[MW05] C. Marriott and J. Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, vol. 14, no. 2, pp. 122–152, 2005. doi:10.1007/s00037-005-0194-x

[MY23] T. Metger and H. Yuen. stateQIP = statePSPACE. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1349–1356, 2023. doi:10.1109/FOCS57990.2023.00082

[NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010. doi:10.1017/CBO9780511976667

[NR99] M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby—rackoff revisited. *Journal of Cryptology*, vol. 12, no. 1, pp. 29–66, 1999. doi:10.1007/PL00003817

[NSBU16] D. Nagaj, O. Sattath, A. Brodutch, and D. Unruh. An adaptive attack on Wiesner's quantum money. *Quantum Information & Computation*, vol. 16, pp. 1048–1070, 2016. doi:10.26421/QIC16.11-12-7

[NW06] T. Neary and D. Woods. Small fast universal turing machines. *Theoretical Computer Science*, vol. 362, no. 1, pp. 171–195, 2006. doi:10.1016/j.tcs.2006.06.002

[NY04] H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Information Processing Letters*, vol. 90, no. 4, pp. 195–204, 2004. doi:10.1016/j.ipl.2004.02.005

[OED22] Clonable, adj. In *Oxford English Dictionary Online*. 2022. www.oed.com/view/Entry/272601

[Ort18] J. Ortigoso. Twelve years before the quantum no-cloning theorem. *American Journal of Physics*, vol. 86, no. 3, pp. 201–205, 2018.
doi:10.1119/1.5021356

[OV22] R. O'Donnell and R. Venkateswaran. The quantum union bound made easy. In *Symposium on Simplicity in Algorithms (SOSA 2022)*, pp. 314–320, 2022.
doi:10.1137/1.9781611977066.25

[Par70] J. L. Park. The concept of transition in quantum mechanics. *Foundations of Physics*, vol. 1, no. 1, pp. 23–33, 1970.
doi:10.1007/BF00708652

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
doi:10.1145/359340.359342

[RY22] G. Rosenthal and H. Yuen. Interactive Proofs for Synthesizing Quantum States and Unitaries. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pp. 112:1–112:4, 2022.
doi:10.4230/LIPIcs.ITCS.2022.112

[Sat23] O. Sattath. Uncloneable cryptography. *Communications of the ACM*, vol. 66, no. 11, pp. 78 – 86, 2023.
doi:10.1145/3576897

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
doi:10.1145/359168.359176

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science (FOCS 1994)*, pp. 124–134, 1994.
doi:10.1109/SFCS.1994.365700

[SP00] P. W. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, vol. 85, no. 2, pp. 441–444, 2000.
doi:10.1103/physrevlett.85.441

[TFKW13] M. Tomamichel, S. Fehr, J. Kaniewski, and S. Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, vol. 15, no. 10, p. 103002, 2013.
doi:10.1088/1367-2630/15/10/103002

[Unr15] D. Unruh. Revocable quantum timed-release encryption. *Journal of the ACM*, vol. 62, no. 6, p. 49, 2015.
doi:10.1145/2817206

[vDLL+24] W. van Dam *et al.* End-to-end quantum simulation of a chemical system. Eprint arXiv:2409.05835 [quant-ph], 2024.
doi:10.48550/arXiv.2409.05835

[vdVCRŠ22] B. van der Vect, X. Coiteux-Roy, and B. Škorić. Can't touch this: Unconditional tamper evidence from short keys. *Quantum Information & Computation*, vol. 22, no. 5&6, pp. 385–407, 2022.
doi:10.26421/QIC22.5-6-1

[VV14] U. Vazirani and T. Vidick. Fully device-independent quantum key distribution. *Physical Review Letters*, vol. 113, no. 14, p. 140501, 2014.
doi:10.1103/PhysRevLett.113.140501

[WAB21] S. Wang, C. Adams, and A. Broadbent. Password authentication schemes on a quantum computer. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE 2021)*, pp. 346–350, 2021.
doi:10.1109/QCE52317.2021.00051

[Wat09] J. Watrous. Quantum computational complexity. In *Encyclopedia of complexity and systems science*, pp. 7174–7201. Springer, 2009.
doi:10.1007/978-3-642-27737-5_428-3

[Wat18] J. Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.
doi:10.1017/9781316848142

[Wer98] R. F. Werner. Optimal cloning of pure states. *Physical Review A*, vol. 58, no. 3, p. 1827, 1998.
doi:10.1103/PhysRevA.58.1827

[Wie83] S. Wiesner. Conjugate coding. *ACM SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983.
doi:10.1145/1008908.1008920

[Wil17] M. M. Wilde. *Quantum Information Theory*. Cambridge University Press, second edition, 2017.
doi:10.1017/9781316809976

[Win99] A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2481–2485, 1999.
doi:10.1109/18.796385

[WZ82] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, vol. 299, pp. 802–803, 1982.
doi:10.1038/299802a0