Unbounded Leakage-Resilience and Intrusion-Detection in a Quantum World

Alper Çakan* Vipul Goyal† Chen-Da Liu-Zhang‡ João Ribeiro§

Abstract

Can an adversary hack into our computer and steal sensitive data such as cryptographic keys? This question is almost as old as the Internet and significant effort has been spent on designing mechanisms to prevent and detect hacking attacks. Once quantum computers arrive, will the situation remain the same or can we hope to live in a better world?

We first consider ubiquitous side-channel attacks, which aim to leak side information on secret system components, studied in the *leakage-resilient* cryptography literature. Classical leakage-resilient cryptography must necessarily impose restrictions on the type of leakage one aims to protect against. As a notable example, the most well-studied leakage model is that of *bounded leakage*, where it is assumed that an adversary learns at most ℓ bits of leakage on secret components, for some leakage bound ℓ. Although this leakage bound is necessary, many real-world side-channel attacks cannot be captured by bounded leakage. In this work, we design cryptographic schemes that provide guarantees against *arbitrary* side-channel attacks:

- Using techniques from unclonable quantum cryptography, we design several basic leakage-resilient primitives, such as public- and private-key encryption, (weak) pseudorandom functions, and digital signatures which remain secure under (polynomially) *unbounded* classical leakage. In particular, this leakage can be much longer than the (quantum) secret being leaked upon. In our view, leakage is the result of observations of quantities such as power consumption and hence is most naturally viewed as classical information. Notably, the leakage-resilience of our schemes holds even in the stronger adaptive "LOCC leakage" model where the main adversary and the leakage adversary can cooperate via arbitrary local quantum operations and two-way classical communication in multiple rounds.
- What if the adversary simply breaks in and obtains unbounded *quantum* leakage (thus making leakage-resilience impossible)? Going beyond leakage, what if the adversary can even tamper with the data arbitrarily? We initiate the study of *intrusion-detection* in the quantum setting, where one would like to detect if security has been compromised even in the face of an arbitrary intruder attack which can leak and tamper with classical as well as quantum data. We design cryptographic schemes supporting intrusion detection for a host of primitives such as public- and private-key encryption, digital signature, functional encryption, program obfuscation and software protection. Our schemes are based on techniques from cryptography with secure key leasing and certified deletion.

*Carnegie Mellon University. acakan@andrew.cmu.edu.

†NTT Research & Carnegie Mellon University. vipul@cmu.edu

‡Luzern University of Applied Sciences and Arts & Web3 Foundation. chen-da.liuzhang@hslu.ch. Part of the work was done while at NTT Research.

§NOVA LINCS & NOVA School of Science and Technology. joao.ribeiro@fct.unl.pt. Part of the work was done while at Carnegie Mellon University.

Contents

1 Introduction

A central problem in the area of computer security is to store sensitive data securely. This could mean we would like to make any intrusion into the system harder to realize, as well as detect if an intrusion did occur. This is a notoriously hard problem with significant resources spent on preventing and mitigating such attacks. Indeed, in the classical setting all information can theoretically be copied, and so we can only rely on heuristic countermeasures for such attacks. We study this question in the quantum setting and show that the quantum world might offer certain advantages - at least as far as protecting cryptographic secrets such as decryption or signing keys is concerned. Indeed, securely storing cryptographic keys has proven to be a notoriously hard problem.

We first consider side-channel (also known as *leakage*) attacks. Real-world implementations of cryptographic schemes are often vulnerable to side-channel attacks, which allow an adversary to obtain side information from secret components such as a secret key. This can be achieved, for example, by measuring the time elapsed or the electromagnetic radiation emitted during computations – such simple practical attacks stretch back some decades [Koc96, QS01, AARR03] and have proven catastrophic for textbook versions of several well known schemes. As a response to this, *leakage-resilient cryptography*, the study of cryptographic schemes resilient against many types of side-channel attacks, has received significant interest. The survey of Kalai and Reyzin [KR19] is an excellent source for many of the developments in this area.

Arguably the most well studied leakage model is that of *bounded leakage*. In this model, it is assumed that the adversary may not leak more than ℓ bits of leakage from a secret component, where ℓ is some leakage bound that is smaller than the secret length k. For example, in the setting of secret-key encryption with a secret key $\mathsf{sk} \in \{0,1\}^k$, the adversary chooses an arbitrary function $f : \{0,1\}^k \to \{0,1\}^\ell$, where $\ell < k$ represents the leakage bound, and learns the bounded leakage $f(\mathsf{sk})$.

Is a Leakage Bound Justified? Generally, the justification for a leakage bound is that in the absence of one, the adversary can just leak the whole secret and no security guarantees are possible. However, it is quite often the case that real world side channels attacks do not adhere to any a priori bounded leakage limit [BFO+21]. Moreover, even choosing a leakage bound entails predicting adversarial capabilities, and these predictions may be wildly incorrect. Nonetheless, the study of bounded leakage-resilient cryptography has given rise to a beautiful and highly successful area of research. It has been impactful not just in leakage-resilience but even in other seemingly unrelated areas in cryptography.

Leakage-Resilience in a Quantum World. What if the secret being leaked on is quantum? Quantum information behaves in a fundamentally different way compared to its classical counterpart. While classical schemes can only tolerate a bounded amount of leakage, the same may not be true for quantum schemes. This raises the following tantalizing question:

> Is it possible to design cryptographic schemes based on the laws of quantum mechanics which can tolerate any arbitrary unbounded leakage?

We answer the above question in the affirmative by proposing a host of cryptographic schemes resilient to arbitrary classical leakage. In our view, leakage is the result of *observations* of quantities such as power consumption, time elapsed, and temperature fluctuations. Hence, most types of leakage are naturally viewed as classical information. Therefore, *our schemes can even be seen*

as leakage-proof rather than just leakage-resilient. The leakage could even be a result of possibly adaptive measurements informed by various rounds of feedback between adversaries. This motivates our LOCC (local operations and classical communication) leakage model as we will explain later.

We design a host of cryptographic schemes such as public- and private-key encryption schemes, digital signatures, and (weak) pseudorandom functions which are resilient to unbounded (polynomial) classical leakage. All our schemes are secure in the general LOCC leakage model. For example, in the public-key encryption setting, we can envision a "main" adversary who tries to win the ciphertext indistinguishability game, and a leakage adversary who has access to the secret key. We wish to construct public-key encryption schemes with quantum secret keys that remain secure even when the main adversary can communicate with the leakage adversary via any LOCC protocol. In particular, such a scheme would tolerate any (polynomially) unbounded and even adaptive classical leakage.

Remarkably, for the specific case of secret sharing schemes, the above question is equivalent to the problem of *quantum data hiding* [TDL01, DLT02]. Even though it was studied with a different motivation and used a different terminology, quantum data hiding schemes have been constructed unconditionally and lead to leakage-resilient secret sharing schemes (albeit with imperfect reconstruction). Please see Section 1.2 for more details.

Intrusion-Detection in a Quantum World. What if an adversary simply breaks in and can get unbounded *quantum* leakage after all? In this case, the adversary can just leak the whole secret state and, analogously to the classical setting, all bets are now off. Even worse, what if the adversary can even tamper with the stored data in an arbitrary manner? Indeed, in the quantum setting the boundary between leakage and tampering is somewhat blurred. This raises the following question:

> Can we still achieve meaningful security guarantees in the face of arbitrary (QPT) operations including unbounded quantum leakage as well as unbounded tampering with quantum as well as classical data?

We refer to the above setting as *intrusion-detection* in the quantum world. While intrusion-detection is fundamentally impossible in the classical setting (since an adversary may just clone secret system components without causing any changes to the system's state), it has nonetheless been widely studied in practice and is considered a highly desirable security goal. For example, tamper-proof audit logs have been extensively studied which, under certain assumptions, can detect if a machine has been broken into [SYC04, SJEL14, ALP22].

Based on principles of quantum mechanics, we are able to design many primitives, including public-key encryption and digital signature schemes supporting intrusion-detection. More precisely, our schemes provide the following guarantee. Suppose that an adversary was able to arbitrarily act on the quantum secret, tampering it and leaking sufficient information to break the security of the primitive (e.g., break indistinguishability in the case of public-key encryption). Then, a procedure called TestIntrusion outputs INTRUSION with overwhelming probability, indicating that an attack occurred and security has been compromised. TestIntrusion takes as input the residual secret (e.g., the residual secret key in the case of public-key encryption), and *a classical public verification key.* A copy of this public verification key could be stored offline or anywhere outside the machine under attack. Note that the notion of intrusion-detection is meaningless in the absence of such a public verification key since in that case what constitutes an attack and what is a valid modification to the quantum secret is not well-defined.

We require that if the procedure TestIntrusion outputs NO INTRUSION, then, with overwhelming probability, *either there has been no attack, or any possible attack was not successful in breaking the*

scheme's security! All of our results in this direction are obtained via a connection to cryptography with secure key leasing [AL21, BGG+23] and cryptography with certified deletion [BK23, BGG+23].

1.1 Our Results

1.1.1 Unbounded Leakage-Resilience Against LOCC Protocols

We design schemes for public-key encryption (PKE), signature schemes, and pseudorandom functions (PRFs) that tolerate LOCC leakage against polynomial adversaries. For all tasks mentioned here, we consider a *main* adversary and a leakage adversary, the latter holding the quantum secret key. The adversaries are allowed to interact via an LOCC protocol, i.e., to communicate classically and also act on their local quantum states. After their LOCC protocol is over, the main adversary participates in the respective security game (e.g., in case of PKE, receives a challenge ciphertext and guesses the plaintext).

We present an informal description of our results below. More details can be found in Section 2.

Theorem (informal)**.** *Assuming the existence of post-quantum sub-exponentially secure iO, one-way functions and the quantum hardness of LWE, there exist LOCC-leakage-resilient schemes for public-key encryption, digital signatures, and weak PRFs.*

Unclonability Implies Non-Adaptive Unbounded Classical Leakage-Resilience. While we are able to construct LOCC leakage-resilient schemes for various primitives using techniques from unclonable cryptography, it is still an interesting question to investigate the relationship between unclonability (i.e., anti-piracy security) and LOCC leakage-resilience in general. We show that in many cryptographic settings unclonability implies leakage-resilience against LOCC protocols with 1 or 2 rounds.

Theorem (informal)**.** *Let X be a {public-key encryption, digital signature, PRF} scheme that satisfies anti-piracy (i.e., unclonable) security. Then, X also satisfies non-adaptive unbounded classical (i.e., 1-round LOCC) leakage-resilience.*

Moreover, let X be a {public-key encryption, digital signature, PRF} scheme that satisfies anti-piracy security against adversaries with non-uniform quantum advice. Then, X also satisfies 2-round LOCC leakage-resilience.

Theorem (Theorem 18, informal)**.** *Let* PKE *be a public-key encryption scheme with key protection that satisfies CPA-style anti-piracy security. Then, either* PKE *is CPA-style 1-round LOCC leakage-resilient, or* PKE *can be used to build* weak *quantum lightning.*

While this might lead one to think that LOCC leakage-resilience in general is weaker than and implied by anti-piracy (i.e. unclonability), we show a justification that this is not the case. If quantum lightning [LAF+10, Zha19] and virtual black-box obfuscation for quantum circuits exist, then we show that unclonability *does not* imply even 2-round[1] LOCC leakage-resilience. While virtual black-box obfuscation for general circuits has been shown to be impossible, still our construction serves as a heuristic justification for a separation between anti-piracy and LOCC leakage-resilience.

Theorem (informal)**.** *Suppose quantum lightning and virtual black-box obfuscation for quantum circuits exist. Then, there exists a PKE scheme that satisfies anti-piracy security (against adversaries* without *non-uniform quantum advice) but does not satisfy 2-round LOCC leakage-resilience.*

[1] Or, 3-round depending on the underlying advice model for adversaries

Similarly, there exists a public-key encryption scheme that satisfies anti-piracy security (against adversaries with *non-uniform quantum advice) but does not satisfy* 3*-round LOCC leakage-resilience.*

1.1.2 Intrusion-Detection

As pointed out in the introduction, tolerating unbounded *quantum* leakage is impossible, since the adversary can leak the whole secret. Therefore, we aim to achieve intrusion-detection instead. More specifically, we design cryptographic primitives with an intrusion-detection algorithm which can detect whether *useful* leakage has been obtained on the secret key. These results are obtained by establishing a connection to cryptographic schemes with publicly verifiable secure leasing [BGG+23].

Theorem (informal)**.** *Suppose that there exists a* [public-key encryption, digital signature, functional encryption, obfuscation, software protection] *scheme with publicly verifiable secure leasing. Then, there exists a* [public-key encryption, digital signature, functional encryption, obfuscation, software protection] *scheme with intrusion-detection.*

We show that the other direction is also true: intrusion-detection implies secure key-leasing with quantum certificates for the primitives listed above. We refer the reader to Section 8 for details.

[BGG+23] constructs schemes with secure leasing based on indistinguishability obfuscation and [LLQZ22] constructs a digital signature scheme with unclonable signing key (which implies secure key leasing). Hence, we get the following corollaries.

Corollary. *Assuming post-quantum indistinguishability obfuscation and injective one-way functions, there exists a public-key encryption and functional encryption scheme with intrusion-detection.*

Corollary. *Assuming post-quantum subexponentially secure indistinguishability obfuscation, one-way functions, and quantum hardness of LWE, there exists a digital signature scheme with intrusion-detection.*

Corollary. *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists a differing-inputs obfuscation and software protection scheme with intrusion-detection.*

While post-quantum indistinguishability obfuscation is a strong assumption, we show that public-key quantum money is implied by intrusion-detection. Since the only known construction of public-key quantum money in the plain model [Zha19] is based on post-quantum $i\mathcal{O}$, our assumption can be considered unavoidable until a breakthrough is achieved in the construction of public-key quantum money.

Theorem (informal)**.** *Suppose there exists a* {*public-key encryption, digital signature, functional encryption, obfuscation*} *scheme with intrusion-detection. Then, public-key quantum money exists.*

Through our result showing that key leasing with public verification implies intrusion detection, we also prove that key leasing with public verification implies public-key quantum money.

1.2 Related Work

Classical Leakage-Resilience. The topic of leakage-resilience has witnessed significant interest over the past few decades in the classical setting. We highlight some of these developments here and place our work in context.

As mentioned before, the most commonly studied leakage model is *bounded leakage*, which corresponds to the adversary learning a bounded-output function of secret data. This notion

has been realized in various related ways throughout the literature. The setting of cryptography resilient to *memory attacks*, introduced by Akavia, Goldwasser, and Vaikuntanathan [AGV09], corresponds to the scenario where the leakage function is efficiently computable and the leakage bound grows with the size of the secret data. Akavia, Goldwasser, and Vaikuntanathan [AGV09] constructed leakage-resilient public-key cryptography based on LWE. Later works in this direction constructed leakage-resilient versions of many basic cryptographic primitives in the computational setting, such as weak pseudorandom functions, symmetric-key encryption, signatures, and message authentication codes from general and minimal assumptions (e.g., see [KV09, NS09, FKPR10, BSW11, FNV15, HLAWW16]), and also designed schemes secure against continual memory attacks (e.g., see [DHLW10, BKKV10]). Another related notion is the *bounded retrieval model*, introduced in [DLW06, Dzi06]. The difference with respect to the memory attacks setting is that here the leakage bound is an absolute quantity. Bounded leakage-resilience has also been studied in depth for information-theoretic primitives, assuming that the bounded leakage functions can be computationally-unbounded. Most relevant to our work, a long line of research has studied several notions of classical bounded leakage-resilient secret sharing [DP07, BDIR18, GK18, ADN+19, SV19, CGG+20, CKOS22].

Other lines of research have studied the question of whether the bounded leakage assumption can be weakened or replaced by other reasonable assumptions. This has led to the development of *cryptography with auxiliary input* (e.g., see [DKL09, DGT+10]), where the adversary can learn unbounded leakage from secret data provided that it is *computationally* infeasible to gain information about the secret data from the leakage and any public information (in other words, the leakage is in some sense "one-way"). In parallel, other works have studied *noisy leakage-resilience* [NS09, PR13, DFS15, HLAWW16, DDF19, PGMP19, BFO+21], where the adversary learns unbounded but highly noisy leakage from secret data.

The works discussed above consider leakage from memory or encoded storage. A parallel and closely related line of research has studied leakage-resilience in settings where the adversary has some access to computational processes. For example, the adversary may adaptively probe circuit wires while computation is taking place [CJRR99, ISW03], or may have access to the entire state of computation save for some leak-free hardware component [MR04].

It is clear that classical leakage-resilient cryptographic schemes can exist only if we impose restrictions on the type of allowed leakage, such as the ones above (boundedness, one-way-ness, or noisiness), or if we allow regular refreshing of secret data (as in the setting of continual memory attacks). In contrast, we show that there exist quantum cryptographic schemes for several basic tasks which are resilient to *arbitrary* classical leakage. Moreover, even if the leakage is quantum instead of classical, in which case leakage-*resilience* becomes impossible, then we can still *detect* harmful quantum leakage attacks, a property that is also not achievable in the classical setting.

Quantum Data Hiding. The problem of quantum data hiding, which corresponds to LOCC-leakage-resilient secret sharing, was introduced by Terhal, DiVincenzo, and Leung [TDL01]. Another motivation for this problem was to understand the power of LOCC versus unconstrained protocols for distinguishing quantum states [DLT02]. Initial works [TDL01, DLT02] provided constructions in the 2-party setting. This was then extended to more than 2 parties and to general access structures by Eggeling and Werner [EW02], although the secret sharing schemes provided there have a small probability of reconstruction error. Improved parameters were achieved in later work [HLSW04, HLS05], albeit via non-explicit schemes. Other works exploring quantum data hiding and distinguishability of quantum states include [MWW09, LPW18]. In Appendix B.3, we showcase a different approach based on BB84 states towards designing explicit secret sharing

schemes for general access structures resilient against unbounded non-adaptive classical leakage.

2 Technical Overview

A common theme across several of our results is that they are based on techniques from quantum copy-protection/unclonability and secure key leasing (i.e., keys with certified deletion). The connection between leakage resilience and copy-protection is as follows. Suppose one can obtain classical leakage on a quantum secret satisfying copy-protection security, which is "functionally equivalent" to the secret itself (e.g., this leakage allows one to decrypt in case the quantum *secret* is a secret key of an encryption scheme). Since any classical information can be cloned, this gives us a way of essentially obtaining multiple states having the same functionality as the original quantum secret. Since we assumed the quantum secret was "unclonable", we arrive at a contradiction, which should yield leakage-resilience security. While this basic observation is indeed our starting point, this is not enough due to new challenges in the setting of leakage-resilience. We highlight some challenges, taking public-key encryption as our running example.

Challenge 1: Main Adversary's State Cannot be Cloned. In the context of public-key encryption, the leakage game consists of an interaction between the main adversary $\mathcal{A}_{\mathsf{Main}}$ (which will attempt to decrypt the challenge ciphertext) and the leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ (which has the quantum secret key). The implicit assumption in the observation from the previous paragraph is that there is a single classical message from $\mathcal{A}_{\mathsf{Leak}}$ which is sufficient to decrypt the challenge ciphertext. However, note that, in general, the leakage game will have multiple rounds of interaction between $\mathcal{A}_{\mathsf{Leak}}$ and $\mathcal{A}_{\mathsf{Main}}$, and the decryption of the challenge ciphertext will utilize the final (quantum) internal state of $\mathcal{A}_{\mathsf{Main}}$. In general, this state cannot be cloned even if the messages sent by $\mathcal{A}_{\mathsf{Leak}}$ are classical, and this holds true even if one started with multiple copies of $\mathcal{A}_{\mathsf{Main}}$ as non-uniform advice. This is because each copy of $\mathcal{A}_{\mathsf{Main}}$ could produce a different message to be sent to $\mathcal{A}_{\mathsf{Leak}}$ (e.g., because of a measurement). However, only a single copy of $\mathcal{A}_{\mathsf{Leak}}$ can be run (because we only have a single copy of the secret key), resulting in the inability to answer multiple different messages from multiple copies of $\mathcal{A}_{\mathsf{Main}}$. This challenge becomes particularly interesting in the computational setting.

Challenge 2: One Adversary vs Two Adversaries. In PKE with unclonable secret keys [CLLZ21], an adversary in possession of the decryption key R_{dec} splits it across two adversaries in an arbitrary manner, and two challenge ciphertexts are then sent to these adversaries. Afterwards, we require that the probability that *both* adversaries can simultaneously correctly decrypt their ciphertexts is negligibly close to $1/2$. This means that the adversaries' baseline success probability in the unclonable decryption game is $1/2$ (since one of the adversaries can simply keep the original decryption key R_{dec} and correctly distinguish its encoded message with probability 1 and the other one can randomly guess with probability $1/2$). On the other hand, the guarantee in leakage-resilient PKE requires that the probability of correctly decrypting *one* ciphertext given the leakage is negligibly close to $1/2$. This means that the probability of correctly decrypting *two* ciphertexts, as in the unclonable decryption game, should be close to $1/4$, instead of close to $1/2$. This means that a direct reduction to the unclonable decryption game cannot be obtained. We run into a similar issue while considering unclonable PRFs as well.

Connection to Quantum Lightning. Despite the above differences, the notion of leakage-resilience and unclonability are intimately connected. A natural question is to study a formal

relationship between these notions. We show that for non-adaptive leakage (i.e., the leakage consists of only a single message from $\mathcal{A}_{\mathsf{Leak}}$ to $\mathcal{A}_{\mathsf{Main}}$), unclonability implies leakage-resilience. However for adaptive leakage (where $\mathcal{A}_{\mathsf{Leak}}$ and $\mathcal{A}_{\mathsf{Main}}$ can interact in multiple rounds), such an implication is unlikely to hold. In particular, somewhat surprisingly, we show that this question is intimately connected to the existence of quantum lightning. We show that if quantum lightning and quantum VBB obfuscation exist, then there exists a PKE with unclonable decryption keys which is provably not LOCC leakage resilient.

2.1 LOCC Leakage-Resilience Property for Coset States

We first start by discussing coset states, previously studied by Coladangelo, Liu, Liu, and Zhandry [CLLZ21] and Vidick and Zhang [VZ21]. We show a LOCC leakage-resilience property for such states and using this property, we are able to prove that various existing anti-piracy secure primitives based on coset states are also LOCC leakage-resilient.

A *coset state* is a state of the form $\sum_{a\in A}(-1)^{\langle a,s'\rangle}|a+s\rangle$ where $A \subseteq \mathbb{F}_2^\lambda$ is a subspace of dimension $\dim(A) = \lambda/2$ and $s, s' \in \mathbb{F}_2^\lambda$. Coset states, when A, s, s' as above are randomly sampled, satisfy an important security notion called *monogamy-of-entanglement* (MoE) [CLLZ21, VZ21].

Monogamy-of-Entanglement Property of Coset States. In a monogamy-of-entanglement game, the adversary is presented with a randomly sampled coset state, and it is required to output two (possibly entangled) adversaries. Then, these adversaries are given the description of A, and they are required to *simultaneously* output vectors v, w such that $v \in A+s$ and $w \in A^\perp + s'$. Note that the initial adversary can simply measure the state in either computational or Hadamard basis to obtain such v or w, but not both of them at the same time since such a measurement irreversibly destroys the information in the other basis. More generally, Vidick and Zhang [VZ21] show that no (unbounded) adversary can win the game above except with subexponentially small probability. Further, [CLLZ21] also show a so-called computational MoE property: based on computational assumptions, the winning probability of any (polynomial time) adversary in the game above is still negligible even when it is presented at the beginning of the game with an obfuscated program that allows it to query for membership in $A + s$ and $A^\perp + s'$. A variation implicitly used in [CLLZ21, LLQZ22], which we formally prove secure in our paper, presents the initial adversary with a *tuple* of coset states (along with the membership checking programs) and requires the created two adversaries to output vectors in either $A_i + s_i$ or $A_i^\perp + s_i'$ depending on the bits of the random challenge strings r_1 and r_2 presented to them.

LOCC Leakage-Resilience Property for Coset States. We raise the question of whether coset states also satisfy some form of leakage-resilience, and if such a property can be used to construct leakage-resilient schemes for various cryptographic functionalities. Our first technical contribution is to answer these questions affirmatively. We show a leakage-resilience property for coset states and, using this result, we show that the coset state based copy-protection schemes of [CLLZ21] and [LLQZ22] for public-key encryption, pseudorandom functions, and digital signatures also satisfy LOCC leakage-resilience. We discuss the former here and the latter will be discussed in the upcoming sections.

We define a leakage-resilience game for coset states as follows. Consider a leakage adversary $\mathcal{A}_{\mathsf{Leak}}$, in possession of a tuple of coset states, and a main adversary $\mathcal{A}_{\mathsf{Main}}$, and assume that these adversaries do not share entanglement. The two adversaries execute their LOCC protocol. That is, they are allowed to communicate classically and apply local quantum operations to their states. After their protocol is over, $\mathcal{A}_{\mathsf{Main}}$ is also given the descriptions of the subspaces A_i. Then, $\mathcal{A}_{\mathsf{Main}}$ is

presented with a random challenge string r, and is required to output vectors in $A_i + s_i$ or $A_i^\perp + s'_i$ depending on the i-th challenge bit, $(r)_i$. We show that any unbounded LOCC adversary wins this game with subexponentially small probability.

Connections to the Monogamy-of-Entanglement Property. While LOCC leakage-resilience property might seem to be implied by the monogamy-of-entanglement property in a straightforward manner, in reality this does not immediately follow. Consider the following natural proposal for a reduction: in the MoE game, the initial adversary simulates the leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ and the main adversary $\mathcal{A}_{\mathsf{Main}}$. However, observe that in the MoE game we need to output two adversaries that will need to answer the vector outputting challenges, while in LOCC leakage-resilience there is only one such adversary. Therefore, we would need to create two copies of the final local state of $\mathcal{A}_{\mathsf{Main}}$ to succeed in the MoE game. Indeed, our proof precisely manages to do this. Observe that during each round $\mathcal{A}_{\mathsf{Main}}$ takes as input its previous internal state and the latest leakage messages from $\mathcal{A}_{\mathsf{Leak}}$ and it produces a state for the next round and a classical message to $\mathcal{A}_{\mathsf{Leak}}$. If we have sufficiently many (i.e., exponential) copies of its previous state, then by repeatedly running $\mathcal{A}_{\mathsf{Main}}$ on these copies we can obtain another copy of its next state conditioned on it producing the same message to $\mathcal{A}_{\mathsf{Leak}}$. Note that we want to obtain a copy conditioned on producing the same message, since $\mathcal{A}_{\mathsf{Leak}}$ starts with a single copy of the coset state tuple and so we can run $\mathcal{A}_{\mathsf{Leak}}$ only once each round. Then, we show that the multi-copy generation procedure described above succeeds in some finite amount of time, and so, since we are working with unbounded adversaries, the reduction succeeds. See Section 4 for further details.

Moving to the Computational Setting. While the above result is a step forward, as we will later discuss all coset state-based constructions of [CLLZ21] and [LLQZ22] crucially rely on the membership checking programs for $A_i + s_i$ and $A_i^\perp + s'_i$ for their correctness, and hence they also rely on the computational MoE property for their security. Therefore, analogous to MoE, we define a computational LOCC leakage-resilience game where the now computationally-bounded adversary $\mathcal{A}_{\mathsf{Leak}}$ is also presented with obfuscated programs that allow it to query for membership in $A_i + s_i$ and $A_i^\perp + s'_i$. Note that our reduction above from LOCC leakage-resilience to MoE might take exponentially long in general. Therefore, we are not able to utilize the same idea here to reduce the computational LOCC leakage-resilience to computational MoE. However, we show that the reduction of [CLLZ21] from computational MoE to (information-theoretic) MoE that utilizes *subspace hiding* obfuscation shO [Zha19] generalizes to the leakage-resilience setting. The argument mainly relies on using subspace hiding obfuscation to implement the membership checking programs, which can then be replaced with such programs for random superspaces of A_i by the security guarantee of shO. This eventually allows us to remove the membership checking programs, and hence the security reduces to the regular LOCC leakage-resilience game. Therefore, we are able to obtain a computational LOCC leakage-resilience property for coset states.

2.2 LOCC Leakage-Resilient PKE Using Coset States

We introduce the model of LOCC leakage-resilience for public-key encryption. In this model, we consider two adversaries $\mathcal{A}_{\mathsf{Leak}}$ and $\mathcal{A}_{\mathsf{Main}}$ that do no share entanglement. At the beginning of the game $\mathcal{A}_{\mathsf{Leak}}$ is in the possession of the public key and the quantum/protected secret key of the scheme. Then, the two adversaries are allowed to communicate classically and apply quantum operations to their local states, for any polynomial number of rounds. After the protocol is over, the main adversary $\mathcal{A}_{\mathsf{Main}}$ is presented with a challenge ciphertext that it needs to decrypt. Here, we can define two variations: random challenge message and CPA-style. In the former, the challenge

ciphertext is the encryption of a randomly sampled (long) message, and the adversary needs to output the full message to win the game. We require that any adversary wins with at most negligible probability. In the latter, the adversary outputs two messages m_0, m_1 and the challenge ciphertext is the encryption of m_b where $b \leftarrow \{0,1\}$. The adversary wins if it can correctly guess the bit b, and we require that any adversary wins with probability at most $1/2 + \mathsf{negl}(\lambda)$, where λ is the security parameter.

Construction. Now, we move onto our construction. We show (via a new proof) that the coset state-based construction of [CLLZ21] of a public-key encryption with copy-protection also satisfies LOCC leakage-resilience. Let us first informally discuss this construction. We sample a tuple of coset states, and let this be the protected secret key. We also sample obfuscated programs that allow one to query if $v \in A_i + s_i$ and $v \in A_i^{\perp} + s'_i$, as in the computational LOCC leakage-resilience game. We set the public-key to be the tuple of these programs. To encrypt a message m, we sample a random string r and let the ciphertext be $(i\mathcal{O}(P), r)$ where P is the following program: It takes as input some vectors, and it checks if these vectors are in $A_i + s_i$ or $A_i^{\perp} + s'_i$ according to the bit $(r)_i$. If all vectors are in the correct cosets, then it outputs m; otherwise, it outputs $\perp$. It is easy to see that we can indeed construct this program using the public-key defined above.

Proving Security. To argue security, first imagine that $i\mathcal{O}$ was instead a virtual black-box obfuscation scheme. If an LOCC leakage adversary pair $(\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$ is able to correctly decrypt the challenge ciphertext, then $\mathcal{A}_{\mathsf{Main}}$ must be taking r as input and querying the program P at a tuple of vectors that pass the checks corresponding to r as described above. Hence, using this LOCC adversary, we can obtain vectors that are in correct cosets with respect to r. Observe that the checks above correspond exactly to winning condition of the computational LOCC leakage-resilience game for coset states, and therefore we can win that game, which is a contradiction. Therefore, we conclude that the adversary cannot decrypt the challenge ciphertext.

Now we go back to the actual scheme, where $i\mathcal{O}$ is only an indistinguishability obfuscator. In this case, we first replace the ciphertext program with another one that computes the canonical versions of the input vectors and compares them to the canonical vectors of the correct vectors $A_i + s_i$ or $A_i^{\perp} + s'_i$ according to $(r)_i$. Since this is a compute-and-compare (CC) program, we can further replace it with its (distributionally) *virtual black-box* obfuscated version (Definition 10). Finally, by using the CC obfuscation security guarantee, we are able to extract vectors in the correct cosets if the adversary is successfully decrypting the ciphertext. This allows us to win the computational LOCC leakage-resilience game for the coset states, a contradiction. We have thus established the LOCC leakage-resilience of the public-key encryption scheme.

2.3 LOCC Leakage-Resilient Digital Signatures and PRFs Using Coset States

In a similar vein to our results above, we are able to show that the coset state-based digital signatures and PRF family constructions of [LLQZ22] and [CLLZ21] satisfying copy-protection also satisfy LOCC leakage-resilience. We only sketch the approach towards LOCC leakage-resilient digital signatures, since in this construction the signature of a message m will be the evaluation of a PRF on (part of) m. Then, the PRF family with key protection will have the same construction, except that it will not feature a verification key, and so its security is implied[2] by the security of the digital signature scheme.

[2]Almost. There are still some subtleties since the PRF security can be defined based on indistinguishability whereas security for signature scheme is based on unpredictability of the signature. See Section 7.3 for details.

Construction. The construction from [LLQZ22] relies on a complex punctured programming approach. Our quantum signing key is a coset state with its membership checking programs and also an obfuscated signing program. The verification key is again an obfuscated program. The signature of a message m will be $F_1(K_1, m_0)$ where F_1 is a puncturable PRF with some extra properties and m_0 is some part of the message. We will also have two extra PRFs that will only be used in the puncturing argument. Our signing program, on some input m and some vectors, first checks if m satisfies a hidden trigger condition. If not, it verifies if the given vectors are in the correct cosets according to m_0. It outputs the signature $F_1(K_1, m_0)$ if the vectors pass the checks. The verification program similarly first checks if the input x satisfies the hidden trigger condition. If not, it simply verifies the claimed signature. In the hidden trigger mode, both programs interpret their input x as a program of some special form, and run this program.

Proving Security. In the security argument, we first rely on the result of [LLQZ22] that shows that we can replace a uniformly random challenge input x with a hidden trigger input that is specially programmed to also perform coset membership checks, but it instead releases a uniformly random output when the checks pass. Recall that the ciphertexts in our public-key encryption scheme were also obfuscated programs that performed membership checks with respect to a random challenge and released the hidden message when the checks pass. Therefore, after this point, through a similar argument to the proof of LOCC leakage-resilience for PKE, we complete the security proof. See Section 6.2 for details.

2.4 Establishing the Relationship Between Leakage-Resilience and Unclonability

Our discussion will center around public-key encryption, but other primitives also use similar ideas. Let us first recall the definition of unclonability/anti-piracy security for public-key encryption. In an anti-piracy game, the adversary $\mathcal{A}$ is presented with the public key and the protected/quantum secret key. Then, it is required to produce two (possibly entangled) adversaries $\mathcal{A}_1, \mathcal{A}_2$. We present each adversary with independent challenge ciphertexts, and $\mathcal{A}$ wins if both adversaries correctly decrypt simultaneously. Similar to leakage-resilience, we have two variants: CPA-style and random challenge message. However, note that for the CPA-style security game, even though there are two adversaries that need to predict their challenge bit, we still require that the adversaries *simultaneously* correctly predict their independent challenges $b_1, b_2 \leftarrow \{0,1\}$ with probability at most $1/2 + \mathsf{negl}(\lambda)$.

Naive Reduction. Now, consider the naive reduction idea from leakage-resilience to unclonability: $\mathcal{A}$, which has the quantum secret key, simulates $\mathcal{A}_{\mathsf{Leak}}$ and $\mathcal{A}_{\mathsf{Main}}$. However, similarly to the problem of reducing LOCC leakage-resilience for coset states to monogamy-of-entanglement, we run into the following problem: LOCC leakage-resilience has one adversary that needs to decrypt the challenge ciphertext where the unclonability adversary $\mathcal{A}$ needs to output two adversaries that are capable of decrypting the challenge ciphertext. Since we are also making computational assumptions, we cannot use the repeated sampling idea we used before to create two copies of the internal state of $\mathcal{A}_{\mathsf{Main}}$, since it takes exponentially long. Therefore, this particular reduction does not work in general.

When the Naive Reduction Does Work. However, it turns out that the idea above does work in the simpler case of 1-round LOCC leakage-resilience (in the case of a random challenge message). In this model, there is a single classical message in the leakage game, from $\mathcal{A}_{\mathsf{Leak}}$ to $\mathcal{A}_{\mathsf{Main}}$. Since we

can easily *clone* this classical message, the adversary $\mathcal{A}$ for the anti-piracy game can simply create this leakage ℓ on the quantum secret key, then output $(\mathcal{A}_{\mathsf{Main}}, \ell), (\mathcal{A}_{\mathsf{Main}}, \ell)$ as its two adversaries. When $(\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$ is capable of winning the leakage-resilience game with probability $1/\mathrm{poly}(\lambda)$, then $\mathcal{A}$ can win the anti-piracy game with probability $(1/\mathrm{poly}(\lambda))^2$, by a simple argument based on Jensen's inequality. Thus, we get the following result.

Theorem (Theorem 16)**.** *Let* PKE *be a public-key encryption scheme with key protection that satisfies anti-piracy security with random challenge messages. Then, it also satisfies* 1*-round random challenge message LOCC leakage-resilience.*

Note that this argument only works in the case of random challenge messages. In the CPA-style security case, if $(\mathcal{A}_{\mathsf{Leak}}\mathcal{A}_{\mathsf{Main}})$ wins with probability $1/2 + 1/\mathrm{poly}(\lambda)$, then $\mathcal{A}$ will win with probability $(1/2 + 1/\mathrm{poly}(\lambda))^2 \approx 1/4$, while we needed $1/2 + \mathrm{poly}(\lambda)$ to break anti-piracy security.

Reductions Using Non-Uniform Quantum Advice We also consider the case of 2-round LOCC leakage-resilience. In this model, first there is a classical message from $\mathcal{A}_{\mathsf{Main}}$ to $\mathcal{A}_{\mathsf{Leak}}$, and then a classical message from $\mathcal{A}_{\mathsf{Leak}}$ to $\mathcal{A}_{\mathsf{Main}}$. When we try to use the same reduction as above, we run into the problem that we only have a single copy of the state that $\mathcal{A}_{\mathsf{Main}}$ maintains. However, if we assume that the PKE scheme is anti-piracy secure against adversaries with nonuniform quantum advice, we can indeed make the reduction go through as follows. Since $\mathcal{A}_{\mathsf{Main}}$ outputs a classical message to $\mathcal{A}_{\mathsf{Leak}}$ and a state at the beginning, we can write

$$\sum_{x\in\{0,1\}^{k(\lambda)}} q_{x,\lambda}|x\rangle\langle x| \otimes \xi_{x,\lambda} = \mathcal{A}_{\mathsf{Main}}(1^\lambda, \sigma_\lambda),$$

where $k(\lambda)$ is the length of the first message. Then, we can define the following state which automatically gives us two copies of the state of $\mathcal{A}_{\mathsf{Main}}$: $\xi'_\lambda = \sum_{x\in\{0,1\}^{k(\lambda)}} q_{x,\lambda}|x\rangle\langle x| \otimes \xi_{x,\lambda} \otimes \xi_{x,\lambda}$. Crucially, observe that the state that $\mathcal{A}_{\mathsf{Main}}$ after its first message does not actually depend on the PKE scheme instantiation. Therefore, we can use ξ'_λ as the non-uniform quantum advice of $\mathcal{A}$, in which case it will indeed be able to obtain two copies of the final state of $\mathcal{A}_{\mathsf{Main}}$ and the reduction goes through as above. Thus, we get the following result.

Theorem (Theorem 17)**.** *Let* PKE *be a public-key encryption scheme with key protection that satisfies anti-piracy security with random challenge messages against adversaries with non-uniform quantum advice. Then, it also satisfies* 2*-round random challenge message LOCC leakage-resilience against adversaries with non-uniform quantum advice.*

Win-Win Result for CPA-style Security and Quantum Lightning While are not able to directly show that CPA-style unclonability implies CPA-style 1-round LOCC leakage-resilience, we show the win-win result that a public-key encryption scheme that satisfies CPA-style anti-piracy security either also satisfies CPA-style 1-round LOCC leakage-resilience, or can be used to build quantum lightning. The main challenge here is the squaring of $1/2$ issue we have when producing two decryptors in the anti-piracy game. We show that it is either possible to side-step this issue, or we can build quantum lightning. Suppose that PKE is anti-piracy secure but not 1-round LOCC leakage-resilient. We let the public key of the quantum lightning scheme be pk (the public key of PKE) and $\ell \leftarrow \mathcal{A}_{\mathsf{Leak}}(pk, R_{\mathsf{dec}})$. Then, a quantum lightning bolt is the state ρ_{m_0,m_1} of the main adversary, and its serial number is (m_0, m_1), the challenge messages it chooses. Our verification runs $\mathcal{A}_{\mathsf{Main}}$ on the claimed bolt ρ_{m_0,m_1}, the leakage ℓ, and a challenge ciphertext $\mathsf{PKE.Enc}(pk, m_b)$. Verification succeeds if $\mathcal{A}_{\mathsf{Main}}$ correctly predicts b. Now, suppose that is possible to produce two

bolts ρ and σ with the same serial number (m_0, m_1). Then, running the verification algorithm above simultaneously on the bolts corresponds exactly to the CPA-style anti-piracy game. This means that verification of the bolts succeeds with probability $1/2 + \mathsf{negl}(\lambda)$. Correctness follows by the 1-round LOCC leakage-resilience insecurity of PKE, since a valid bolt will succeed with probability $1/2 + 1/\mathrm{poly}(\lambda)$. In summary, we were able to side-step the "one adversary versus two adversaries" problem that plagues the LOCC leakage-resilience to unclonability reduction precisely because quantum lightning also involves a bipartite simultaneous verification. Thus, we obtain the following result.

Theorem (Theorem 18, informal)**.** *Let* PKE *be a public-key encryption scheme with key protection that satisfies CPA-style anti-piracy security. Then, either* PKE *is CPA-style* 1*-round LOCC leakage-resilient, or* PKE *can be used to build* weak *quantum lightning.*

Unclonability Does Not Imply General LOCC Leakage-Resilience. Finally, one might wonder if the fact that we can only show that unclonability implies LOCC leakage-resilience in a limited setting is a weakness of our proof techniques. We also show that this is likely not the case – these limitations are an inherent property of these models. More precisely, we construct a public-key encryption scheme, based on quantum lightning and virtual black box obfuscation, that satisfies anti-piracy but does not satisfy even 2-round LOCC leakage-resilience. We construct such a scheme as follows. Let $d(\lambda)$ denote the length of the serial numbers of the quantum lightning scheme QL. The proof utilizes the fact that in the anti-piracy game we need to come up with two decrypting adversaries, as opposed to a single adversary in the LOCC leakage-resilience game. In our construction, our secret key is a coset state, and we require that, to be able to decrypt a ciphertext, the adversary needs to have a valid quantum lightning bolt, and it needs to output some vectors in either $A_i + s_i$ or $A_i^{\perp} + s'_i$ depending on the serial number of the bolt. Since the 2-round LOCC adversary can sample a bolt, send the serial number to the leakage adversary, and measure the coset state accordingly, this scheme is not LOCC leakage-resilient. However, the anti-piracy adversary will need to produce two valid bolts, which will necessarily have different serial numbers. Therefore, it will need to obtain vectors in both $A_i + s_i$ and $A_i^{\perp} + s'_i$ to decrypt, but this is not possible by monogamy-of-entanglement (or an even simpler version called *direct product hardness theorem* for coset states [CLLZ21]). Hence, it will not be able to produce two freeloaders that can both decrypt. Therefore, we obtain the following result.

Theorem (Theorem 20, informal)**.** *Suppose quantum lightning and virtual black-box obfuscation scheme for quantum circuits exist. Then, there exists a public-key encryption scheme that satisfies anti-piracy security but does not satisfy* 2*-round LOCC leakage-resilience.*

2.5 Intrusion-Detection

We discuss how to construct an intrusion-detection scheme from any publicly verifiable key leasing (i.e., certified key deletion) scheme for a primitive. As an example, we will elaborate on public-key encryption; see Section 8 for the other primitives.

One might ask whether we could just digitally sign the quantum secret under this public key and store it along with the secret itself. The $\mathsf{TestIntrusion}$ procedure would then just check the validity of this signature. However, there are multiple problems with this approach. First, digital signatures for signing quantum states do not exist (and quantum MACs necessarily also encrypt the quantum state, causing loss of functionality). Secondly, an adversary may just try to clone the secret without modifying it (which gives an impossibility result in our model in the classical setting).

We discuss how to construct public-key encryption schemes that support intrusion-detection for unbounded quantum leakage attacks on the decryption key R_{dec}. More precisely, the PKE scheme generates a public key pk, a test key tk (used to test whether an intrusion occurred), and a (quantum) decryption key R_{dec}. An adversary is given $(pk, tk, R_{\mathsf{dec}})$, and it can arbitrarily act on the quantum part to produce a quantum leakage R_{leak} and two challenge messages m_0 and m_1. Note that this may change the state in register R_{dec}. Before the distinguishing game proceeds, a intrusion-detection step is run and the adversary automatically loses if its presence is detected, i.e., if $\mathsf{TestIntrusion}(tk, R_{\mathsf{dec}}) = \mathsf{INTRUSION}$. If no intrusion is detected, we want to guarantee that it is not possible to distinguish between $\mathsf{Enc}(pk, m_0)$ and $\mathsf{Enc}(pk, m_1)$ given $(R_{\mathsf{leak}}, m_0, m_1, tk, pk)$ with probability negligibly close to the baseline

$$\frac{1}{2}\Pr[\mathsf{TestIntrusion}(tk, R_{\mathsf{dec}}) = \mathsf{NO\ INTRUSION}].$$

We construct PKE schemes with these guarantees by establishing a connection to secure key leasing [AL21, KN22, BK23, BGG+23]. We start with the notion of a PKE scheme with secure key leasing, which features an additional *deletion procedure* that, given the secret decryption key R_{dec}, produces a certificate *cert* which should certify that this key was indeed deleted. Roughly speaking, this scheme satisfies the property that an adversary which is able to produce a valid certificate *cert* based on R_{dec}, (validity of *cert* is checked by a Verify procedure using a certificate validation key cvk) cannot distinguish between the ciphertexts $\mathsf{Enc}(pk, m_0)$ and $\mathsf{Enc}(pk, m_1)$ using the leftover state. PKE schemes with secure key leasing have been recently constructed from any post-quantum PKE scheme [AKN+23][3] or post-quantum indistinguishability obfuscation [BGG+23].

We show that we can construct a PKE scheme that supports intrusion-detection from a PKE scheme with secure key leasing. Starting with a PKE scheme with secure key leasing, we construct a $\mathsf{TestIntrusion}$ procedure which essentially tries to produce a deletion certificate for the secret decryption key R_{dec}, and outputs $\mathsf{NO\ INTRUSION}$ if it succeeds. Intuitively, we can argue intrusion-detection security as follows: If an adversary has obtained a leakage that allows it to distinguish ciphertexts, then we should fail to produce a valid deletion certificate using our leftover state. Otherwise, one can create a lessee attacker against the key leasing security that simulates the intrusion adversary on their key, produces a valid deletion certificate using the leftover state, and still succeeds in distinguishing ciphertexts using the leakage. However, the major problem with this approach is reusability: even when there is no attack, we destroy our key when we test for leakage, since we produce a deletion certificate.

Crucially, note that producing a valid deletion certificate using an undisturbed key succeeds with overwhelming probability. Therefore, using the gentle measurement lemma (see Lemma 1), we are able to construct an algorithm for producing a deletion certificate in such a way that we can rewind our algorithm afterwards. While seemingly contradictory, this is not a violation of lessor security. Indeed, in the lessor security game the certificate generation circuit will end with a measurement, while our intrusion-detection procedure will skip this measurement and will instead run the verification procedure coherently. Furthermore, the intrusion-detection procedure will not trace out the garbage registers that are created while producing a certificate or testing for certificate validity, which we then use to rewind the algorithm.

Using similar techniques, we can also build digital signatures, PRFs, functional encryption, and indistinguishability obfuscation schemes supporting intrusion-detection. More generally, we show that the notion of intrusion-detection is *equivalent* to key leasing/certified deletion. See Section 8 for more details.

[3]This scheme unfortunanely lacks public verifiability, which is crucial for intrusion detection since the adversary gets the complete state of the honest party, including the verification key.

Finally, we show that intrusion detection implies public-key quantum money. Let's take the case of public-key encryption with intrusion detection as an example. Our banknote is a quantum secret key, along with the serial number which is the intrusion detection key. Our banknote verification procedure checks for intrusion on the quantum key using the intrusion detection key. First, note that without loss generality, we can assume that our intrusion detection algorithm first checks if the key can successfully decrypt a the encryption of a random message. This check only makes intrusion detection stronger. Then, it is easy to see that simultaneous verification of two banknotes produced by an adversary given a single banknote corresponds to the intrusion detection game. We refer the reader to Section 8 for more details.

3 Notation and Preliminaries

3.1 Notation and Computational Model

We write λ to denote a security parameter. We denote classical sets, random variables and quantum registers by uppercase letters such as X, Y, and Z. We will write $|X|$ to denote the size of the alphabet associated with a register X. Similarly, we denote both classical sets, ensembles and Hilbert spaces by calligraphic letters such as $\mathcal{X}$ and $\mathcal{Y}$. The distinctions will always be clear from context. We write $[n] = \{1, \ldots, n\}$. Given a string $s \in \mathcal{S}^n$ and a set $\mathcal{T} \subseteq [n]$, we denote the projection of s to the coordinates in $\mathcal{T}$ by $s_\mathcal{T} = (s_i)_{i\in\mathcal{T}}$. We write $\mathcal{S}^*$ for $\bigcup_{i=0}^{\infty} \mathcal{S}^i$. For two operators ρ, σ, writing $\rho \geq \sigma$ will mean that $\rho - \sigma$ is positive semi-definite. U_n denotes the uniform distribution over the set $\{0,1\}^n$, and in the same expression all occurrences of U_n will refer to the same sample rather than independent samples, except when differentiated, such as $U_n^{(1)}$ and $U_n^{(2)}$. For a joint state ρ of some quantum registers $R = \{R_1, \ldots, R_n\}$, we will use ρ or ρ^R to denote the joint state and $\rho^{(R_i)_{i\in\mathcal{T}}}$ or $\rho_\mathcal{T}$ to denote the state of the subsystem $(R_i)_{i\in\mathcal{T}}$ alone for some $\mathcal{T} \subseteq [n]$, given by $\mathrm{Tr}_{R\setminus\{R_i\}_{i\in\mathcal{T}}}(\rho)$. Similarly, for a quantum operation Φ, we will sometimes use a superscript to denote the registers to which it is applied, such as Φ^X. We will use $\mathcal{H}$ to denote the Hilbert space associated with a single qubit, that is, $\mathcal{H} = \mathbb{C}^{\{0,1\}}$. $E_{a,b}$ denotes the matrix that has 1 in the entry (a,b) and zeroes in all other entries, and its dimensions will be clear from the context. For a distribution D, we will write $x \leftarrow D$ to mean x is sampled from D. Similarly for a mixed state ρ, we will write $R \leftarrow \rho$ to mean that the register R is initialized to the state ρ. $x \leftarrow \mathcal{A}(a)$ means sample x from the distribution induced by the randomized algorithm $\mathcal{A}$ run on input a. For a tuple or string x, $(x)_i$ will mean the i-th element or character. $\mathsf{U}_{\mathsf{univ}}$ denotes the universal quantum circuit that takes in the description of a quantum circuit and an input, and evaluates the circuit on this input.

Unless otherwise explicitly specified, we will make the following implicit assumptions. All of our cryptographic assumptions will be against non-uniform QPT adversaries, i.e., QPT algorithms (Definition 1) with non-uniform quantum advice. In particular, our assumptions and reductions will be implicitly *post-quantum*. Algorithm will mean a quantum algorithm, and our schemes will be uniform QPT algorithms. Separate adversaries will unentangled. In the computational setting, negligible means negligible in the security parameter, λ and for two ensembles, $X \approx Y$ means $|\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]| < \mathsf{negl}(\lambda)$ for either all QPT adversaries $\mathcal{A}$ or all unbounded adversaries $\mathcal{A}$ (will be clear from context). Sizes and bounds, such as leakage bounds, will implicitly be functions of the security parameter, $\ell = \ell(\lambda)$. In the context of security definitions, *all adversaries* will mean all adversaries that have the appropriate input/output size and interactive structure as required by the security game.

Definition 1 (Computational model)**.** *We fix a universal set of unitary gates, such as {Hadamard, phase, CNOT, $\frac{\pi}{8}$}. We define a quantum polynomial time (QPT) algorithm to be a uniform family of generalized quantum circuits $\{\Phi_\lambda\}_\lambda$ with some fixed polynomials $p(\lambda), q(\lambda)$ where each Φ_λ is constructed by introducing an ancilla register of size at most $q(\lambda)$, applying $p(\lambda)$ many gates from the fixed set of gates to input and ancilla, and finally tracing out some of the registers. Finally, if the output of the algorithm is classical, the remaining registers are measured in the computational basis. Writing Φ will implicitly mean Φ_λ. Note that by Stinespring Dilation Theorem and deferred measurement principle, this model captures all efficient quantum adversaries.*

We will also mainly use the quantum registers model. We consider registers as objects storing *quantum states, which can be correlated or entangled with other registers, and whose states evolve as a result of applying channels to them.*

3.2 Concepts from Quantum Information Theory

We assume familiarity with basic concepts from quantum computation, such as registers, pure and mixed states, density matrices, entanglement, measurements, quantum channels and (vanilla) quantum teleportation. We refer the reader to [NC10] and [Wat18] for an introduction.

3.3 Pseudorandom Functions

Below we introduce the definitions of various pseudorandom function family models.

Definition 2 (Weak pseudorandom functions)**.** *Let $\mathcal{K}$ be an efficient ensemble, denoting the key space, and $\mathcal{X}, \mathcal{Y}$ be families of sets denoting the input and output space respectively. A family of functions $\mathcal{F} = \{f_k\}_k$ is said to be* weak pseudorandom *if any QPT $\mathcal{A}$ has negligible advantage in the following game.*

1. *Challenger samples a key $k \leftarrow \mathcal{K}_\lambda$.*
2. *Challenger samples inputs $x_1, \ldots, x_{p(\lambda)} \leftarrow \mathcal{X}_\lambda$.*
3. *Challenger samples a challenge input x^*.*
4. *Challenger samples a challenge bit $b \leftarrow \{0,1\}$. If $b = 0$, it sets $y^* = x^*$. Otherwise, it samples $y^* \leftarrow \mathcal{Y}_\lambda$.*
5. *Adversary gets $(x_1, f_k(x_1)), \ldots, (x_{p(\lambda)}, f_k(x_p(\lambda))), (x^*, y^*)$, and outputs a guess b'.*
6. *Challenger outputs 1 if and only if $b = b'$.*

We define the advantage of $\mathcal{A}$ to be $\left|\Pr[b' = b] - \frac{1}{2}\right|$.

Definition 3 (Puncturable Pseudorandom Functions)**.** *A puncturable pseudorandom function family is a PRF family $\{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ with an additional algorithm* Puncture *that takes as input the PRF key K and a set $S \subseteq \{0,1\}^{m(\lambda)}$ and outputs a punctured key. We require that it satisfies the following.*

Correctness. *For $S \subseteq \{0,1\}^{m(\lambda)}$ and for all $x \notin S$, we require*

$$\Pr\Big[F(K_S, x) = F(K, x) : K \leftarrow \mathsf{KeyGen}(1^\lambda), K_S \leftarrow \mathsf{Puncture}(K, S)\Big] = 1.$$

Puncturing Security *We require that any stateful QPT adversary wins the following game with probability at most* $1/2 + \mathsf{negl}(\lambda)$.

1. $\mathcal{A}$ *outputs a set* S *and a point* $x \notin S$.
2. *The challenger samples* $K \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $K_S \leftarrow \mathsf{Puncture}(K, S)$ *and* $y \leftarrow \{0,1\}^{n(\lambda)}$.
3. *The challenger samples* $b \leftarrow \{0,1\}$. *If* $b = 0$, *the challenger submits* $K_S, F(K,x)$ *to the adversary. Otherwise, it submits* K_S, y *to the adversary.*
4. *The adversary outputs a guess* b'.
5. *We say that the adversary has won if* $b' = b$.

Definition 4 (Statistically Injective PRF)**.** *A statistically injective PRF with failure probability* ε *is a PRF family where with probability* $1 - \varepsilon$ *over sampling of the key* K, *the function* $F(K, \cdot)$ *is an injective function.*

Theorem 1. *An extracting puncturable PRF with error* $\varepsilon(\lambda)$ *for min-entropy* $k(\lambda)$ *is a puncturable PRF* $\{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ *such that if* X *is a distribution over* $\{0,1\}^{m(\lambda)}$ *with min-entropy* $k(\lambda)$, *then the statistical distance between* $K, F(K,X)$ *and* F, Y *is* $< \varepsilon$ *where* $K \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $Y \leftarrow \{0,1\}^{n(\lambda)}$.

Theorem 2 ([SW13])**.** *Assuming (subexponentially secure) post-quantum one-way functions, there exists a (subexponentially secure) post-quantum puncturable PRF family* $\{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ *for any efficiently computable* m, n.

Theorem 3 ([SW13])**.** *Assuming (subexponentially secure) post-quantum one-way functions, there exists a (subexponentially secure) post-quantum puncturable statistically injective PRF family* $\{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ *with error* $2^{-e(\lambda)}$ *for any efficiently computable* m, n, e *such that* $n(\lambda) \geq 2m(\lambda) + e(\lambda)$.

Theorem 4 ([SW13])**.** *Assuming (subexponentially secure) post-quantum one-way functions, there exists a (subexponentially secure) post-quantum puncturable extracting PRF family* $\{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ *with error* $2^{-e(\lambda)}$ *for min-entropy* $k(\lambda)$ *for any efficiently computable* m, n, e, k *such that* $m(\lambda) \geq k(\lambda) \geq n(\lambda) + 2e(\lambda) + 2$.

3.4 Indistinguishability Obfuscation

In this section, we introduce indistinguishability obfuscation.

Definition 5. *An indistinguishability obfuscation scheme* $i\mathcal{O}$ *for a circuit class* $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ *satisfies the following.*

Correctness. *For all* $\lambda, C \in \mathcal{C}_\lambda$ *and inputs* x, $\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(C)] = 1$.

Security. Let $\mathcal{B}$ be any QPT algorithm such that $\Pr\left[\forall x\ C_0(x) = C_1(x) : (C_0, C_1, R_{\mathsf{aux}}) \leftarrow \mathcal{B}(1^\lambda)\right] \geq 1 - \mathsf{negl}(\lambda)$. Then, for any QPT adversary $\mathcal{A}$,

$$\left| \Pr\left[\mathcal{A}(i\mathcal{O}(C_0), R_{\mathsf{aux}}) = 1 : (C_0, C_1, R_{\mathsf{aux}}) \leftarrow \mathcal{B}(1^\lambda)\right] - \Pr\left[\mathcal{A}(i\mathcal{O}(C_1), R_{\mathsf{aux}}) = 1 : (C_0, C_1, R_{\mathsf{aux}}) \leftarrow \mathcal{B}(1^\lambda)\right] \right| \leq \mathsf{negl}(\lambda).$$

3.5 Digital Signatures

In this section we introduce the basic definitions of signatures schemes.

Definition 6. *A digital signature scheme with message space $\mathcal{M}$ consists of the following algorithms that satisfy the correctness and security guarantees below.*

- $\mathsf{Setup}(1^\lambda)$: *Outputs a signing key sk and a verification key vk.*
- $\mathsf{Sign}(sk, m)$: *Takes the signing key sk, returns a signature for m.*
- $\mathsf{Verify}(vk, m, s)$: *Takes the public verification key vk, a message m and supposed signature s for m, outputs 1 if s is a valid signature for m.*

Correctness *We require the following for all messages $m \in \mathcal{M}$.*

$$\Pr\left[\mathsf{Verify}(vk, m, s) = 1 : \begin{array}{c} sk, vk \leftarrow \mathsf{Setup}(1^\lambda) \\ s \leftarrow \mathsf{Sign}(sk, m) \end{array}\right] = 1.$$

Adaptive existential-unforgability security under chosen message attack (EUF-CMA) *Any QPT adversary $\mathcal{A}$ with* classical *access to the signing oracle has negligible advantage in the following game.*

1. *Challenger samples the keys $sk, vk \leftarrow \mathsf{Setup}(1)$.*
2. *$\mathcal{A}$ receives vk, interacts with the signing oracle by sending classical messages and receiving the corresponding signatures.*
3. *$\mathcal{A}$ outputs a message m that it has not queried the oracle with and a forged signature s for m.*
4. *The challenger outputs 1 if and only if $\mathsf{Ver}(vk, m, s) = 1$.*

If $\mathcal{A}$ outputs the message m before the challenger samples the keys, we call it selective EUF-CMA *security.*

3.6 Functional Encryption

In this section we introduce the basic definitions of functional encryption schemes.

Definition 7 (Functional encryption)**.** *A functional encryption scheme for a family of functions $\mathcal{F}$ consists of the following algorithms that satisfy the correctness and security guarantees below.*

- $\mathsf{Setup}(1)$*: Outputs a master secret key msk and a public key pk.*
- $\mathsf{KeyGen}(msk, f)$*: Takes in the master secret key and a function f, outputs a functional key sk_f for f.*
- $\mathsf{Enc}(pk, m)$*: Takes in the public key and a message m, outputs an encryption of m.*
- $\mathsf{Dec}(sk_f, ct)$*: Takes in a functional key sk_f and a ciphertext, outputs evaluation of the encrypted message under f.*

Correctness *For all functions $f \in \mathcal{F}$ and all messages m, we require the following.*

$$\Pr\left[\mathsf{Dec}(sk_f, ct) = f(m) : \begin{array}{c} msk, pk \leftarrow \mathsf{Setup}(1) \\ sk_f \leftarrow \mathsf{KeyGen}(msk, f) \\ ct \leftarrow \mathsf{Enc}(pk, m) \end{array}\right] = 1.$$

Adaptive indistinguishability security *Any QPT adversary $\mathcal{A}$ has negligible advantage in the following game.*

1. *Challenger samples the keys $msk, pk \leftarrow \mathsf{Setup}(1)$.*
2. *The adversary receives pk. It makes polynomially many queries by sending functions $f \in \mathcal{F}$ and receiving the corresponding functional key $sk_f \leftarrow \mathsf{KeyGen}(msk, f)$.*
3. *The adversary outputs challenge messages m_0, m_1.*
4. *The challenger samples a challenge bit $b \leftarrow \{0,1\}$ and prepares $ct \leftarrow \mathsf{Enc}(pk, m_b)$.*
5. *The adversary receives ct, and it makes polynomially many functional key queries.*
6. *The adversary outputs a guess b'.*
7. *The challenger checks if $f(m_0) = f(m_1)$ for all f queried by the adversary. If not, it outputs 0 and terminates.*
8. *The challenger outputs 1 if $b' = b$.*

We define the advantage of the adversary to be $\left|\Pr[b' = b] - \frac{1}{2}\right|$. If the adversary outputs the challenge messages before the keys are sampled, we call it selective indistinguishability security*.*

3.7 Subspace Hiding Obfuscation

In this section, we introduce subspace hiding obfuscation.

Definition 8 ([CLLZ21])**.** *A subspace hiding obfuscator for a field $\mathbb{F}$ and dimensions d_0, d_1 is an efficient algorithm shO as follows.*

- *Input: Takes as input the description of a subspace $A \subseteq \mathbb{F}^\lambda$ of dimension $d \in \{d_0, d_1\}$.*
- *Output: Outputs a circuit $\hat{A}$.*

Correctness *$\hat{A}$ computes membership in A. That is, denoting by $A(x)$ the function that decides membership in A, the following holds:*

$$\Pr\left[\forall x \quad \hat{A}(x) = A(x)\right] \geq 1 - \mathsf{negl}(\lambda)$$

where $\hat{A} \leftarrow \mathsf{shO}(A)$.

Security *Any QPT adversary $\mathcal{A}$ wins the following game with probability $1/2 + \mathsf{negl}(\lambda)$.*

1. *The adversary submits to the challenger a subspace S_0 of dimension d_0.*
2. *The challenger samples a uniformly random subspace $S_1 \subseteq \mathbb{F}^\lambda$ of dimension d_1 satisfying $S_0 \subseteq S_1$. Then, the challenger runs $S' \leftarrow \mathsf{shO}(S_b)$ and gives S' to the adversary.*
3. *The adversary outputs a guess b' for b.*
4. *We say that the adversary won if $b' = b$.*

Theorem 5 ([Zha19, CLLZ21])**.** *If injective one-way functions exist, then any indistinguishability obfuscator is also a subspace hiding obfuscator for field $\mathbb{F}$ and appropriate dimensions d_0, d_1 such that $\mathbb{F}^{\lambda - d_1}$ is exponential.*

3.8 Compute-and-Compare Obfuscation

In this section, we introduce compute-and-compare obfuscation.

Definition 9 (Compute-and-compare program)**.** *Let $f : \{0,1\}^{a(\lambda)} \to \{0,1\}^{b(\lambda)}$ be a function, $y \in \{0,1\}^{b(\lambda)}$ be a target value and z a hidden message. The following program P, described by (f, y, z), is called a* compute-and-compare program.

$P(x)$: *Compute $f(x)$ and compare it to y. If they are equal, output z. Otherwise, output $\perp$.*

Definition 10. *A compute-and-compare obfuscation scheme for a class of distributions consists of efficient algorithms $\mathsf{CCObf.Obf}$ and $\mathsf{CCObf.Sim}$ that satisfy the following. Consider any distribution $\mathcal{D}$ over compute-and-compare programs along with quantum auxiliary input, in this class.*

Correctness. *For any function (f, y, z) in the support of $\mathcal{D}$ we have that*

$$\Pr\left[\forall x \ D'(x) = D(x) : D' \leftarrow \mathsf{CCObf.Obf}(f, y, z)\right] = 1.$$

Security $(\mathsf{CCObf.Obf}(f, y, z), R_{\mathsf{aux}}) \approx (\mathsf{CCObf.Sim}(1^\lambda, |f|, |y|, |z|), R_{\mathsf{aux}})$ *where* $(f, y, z), R_{\mathsf{aux}} \leftarrow \mathcal{D}(1^\lambda)$.

We say that a distribution $\mathcal{D}$ of such programs is sub-exponentially unpredictable if for any QPT adversary, given the auxilary information and the description of f, the adversary can predict the target value y with at most subexponential probability.

Theorem 6 ([WZ17, CLLZ21])**.** *Assuming the existence of post-quantum $i\mathcal{O}$ and the quantum hardness of LWE, then there exist compute-and-compare obfuscation for any class of sub-exponentially unpredictable distributions.*

3.9 Quantum Goldreich-Levin with Quantum Auxiliary Input

We introduce the Goldreich-Levin theorem with quantum auxiliary input, which will be needed in some of our constructions.

Theorem 7 ([AC02, CLLZ21])**.** *Let $\{|x\rangle\langle x| \otimes \rho_x\}_x$ be a classical-quantum ensemble. Sample r, with $|r| = |x|$, uniformly at random. If there exists an (efficient) quantum algorithm, that given r and ρ_x, outputs $\langle x, r\rangle$ with probability at least $1/2 + \varepsilon$, then there exists an (efficient) quantum algorithm that takes ρ_x and outputs x with probability at least $4 \cdot \varepsilon^2$.*

3.10 Almost As Good As New Lemma

For schemes with quantum keys satisfying correctness with overwhelming probability, we will use the following lemma and the related gentle measurement lemma [Aar05] to argue that the algorithm can be rewound so that the key can be used polynomially many times.

Lemma 1 (Almost As Good As New Lemma [Aar16], verbatim)**.** *Let ρ be a mixed state acting on $\mathbb{C}^d$. Let U be a unitary and $(\Pi_0, \Pi_1 = I - \Pi_0)$ be projectors all acting on $\mathbb{C}^d \otimes \mathbb{C}^{d'}$. We interpret (U, Π_0, Π_1) as a measurement performed by appending an ancillary system of dimension d' in the state $|0\rangle\langle 0|$, applying U and then performing the projective measurement Π_0, Π_1 on the larger system. Assuming that the outcome corresponding to Π_0 has probability $1 - \varepsilon$, we have*

$$\left\|\rho - \rho'\right\|_1 \leq \sqrt{\varepsilon}$$

where ρ' is the state after performing the measurement, undoing the unitary U and tracing out the ancillary system.

3.11 Quantum Lightning

In this section we introduce quantum lightning.

Definition 11 (Quantum lightning [Zha19, Zha23])**.** *A quantum lightning scheme consists of the following algorithms.*

- $\mathsf{Bolt}(1^\lambda)$ *Samples a lightning bolt with a serial number.*
- $\mathsf{Ver}(sn, R)$ *Takes in a supposed bolt register R and a serial number sn. Outputs 1 if R is a valid bolt with serial number sn. Otherwise, outputs 0.*

Correctness. *We require that any honestly generated bolt passes the verification with overwhelming probability.*

Security. *We require that for any QPT adversary $\mathcal{B}$; when we run $(sn, R_1, R_2) \leftarrow \mathcal{B}$, $b \leftarrow \mathsf{Ver}(sn, R_1)$ and $b' \leftarrow \mathsf{Ver}(sn, R_2)$, we have $\Pr[b = b' = 1] \leq \mathsf{negl}(\lambda)$.*

We can also define quantum lightning in a trusted setup model where there is a public-key, sampled during setup, that is used in both bolt generation and verification. Note that this is required if we want security against adversaries with non-uniform quantum advice.

We also define *weak* quantum lightning to be a quantum lightning scheme that satisfies correctness with probability $\alpha(\lambda)$ (instead of overwhelming) and security with probability $\beta(\lambda)$ (instead of negliglible) such that $\alpha - \beta(\lambda) \geq 1/p(\lambda)$ for some polynomial $p(\lambda)$ and all sufficiently large λ.

3.12 Infinitely Often Security

In this section, we define infinitely often security, which is usually needed in win-win results for quantum lightning (e.g., Section 5.2 and [Zha19]). Note that it is a weaker version of usual negligible security definition, however, it can still be considered a meaningful security guarantee.

Definition 12 ([Zha19])**.** *We say a scheme is* not *infinitely often secure if there exists an adversary that breaks it with probability $1/p(\lambda)$ for* all *sufficiently large λ where $p(\cdot)$ is a polynomial.*

4 Coset States

In this section, we start by giving the definition of *coset states* [CLLZ21, VZ21] that we utilize in our constructions and state the monogamy-of-entanglement property they satisfy. Then, we prove an LOCC leakage-resilience theorem for coset states.

Definition 13 ([CLLZ21])**.** *For a subspace $A \subseteq \mathbb{F}_2^n$ and vectors $s, s' \in \mathbb{F}_2^n$, we define $\left|A_{s,s'}\right\rangle$, the* coset state *associated with A, s, s', to be*

$$\left|A_{s,s'}\right\rangle = \sum_{a \in A} \frac{1}{\sqrt{|A|}} (-1)^{\langle s', a \rangle} |a + s\rangle.$$

We usually write $A + s$ to denote both the coset $A + s$ and the program that takes as input a vector $v \in \mathbb{F}_2^n$ and outputs 1 if and only if $v \in A + s$. The distinction will be clear from the context.

Fact 1 ([CLLZ21])**.** *Consider a subspace $A \subseteq \mathbb{F}_2^n$ and vectors $s, s' \in \mathbb{F}_2^n$.*

1. *Given s, s' and the description of A, we can efficiently construct $\left|A_{s,s'}\right\rangle$.*
2. *$H^{\otimes n}\left|A_{s,s'}\right\rangle = \left|A^{\perp}_{s',s}\right\rangle$.*
3. *Define the canonical representative $\mathsf{Can}_A(v)$ to be the lexicographically smallest element in the coset $A + v$. There exists an efficient algorithm to compute $\mathsf{Can}_A(v)$ on input v and the description of A.*

Now we state the monogamy-of-entanglement (*MoE*) properties coset states satisfy. In an MoE game, the adversary is presented with a coset state, and is required to split into two (possibly entangled) adversaries that will need to simultaneously output some vectors in the cosets $A + s$ and $A^{\perp} + s'$.

Theorem 8 (Strong Monogamy-of-Entanglement Property for Coset States [CLLZ21])**.** *Consider the following game between an adversary tuple $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and the challenger.*

$\underline{\mathsf{MoE}(\lambda, \mathcal{A})}$

1. *Sample uniformly at random a subspace A of $\mathbb{F}_2^\lambda$ of dimension $\frac{\lambda}{2}$ and two elements $s, s' \leftarrow \mathbb{F}_2^\lambda$.*
2. *Submit $\left|A_{s,s'}\right\rangle$ to $\mathcal{A}_0$.*
3. *$\mathcal{A}$ outputs two (possibly entangled) registers R_1, R_2.*
4. *For $\ell \in \{1, 2\}$, run $v_\ell \leftarrow \mathcal{A}_\ell(R_\ell, A)$.*
5. *Output 1 if and only if $v_1 \in A + s$ and $v_2 \in A^{\perp} + s'$.*

Then, there exists a constant $C_{\mathsf{MoE}} > 0$ such that for any adversary tuple $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$,

$$\Pr[\mathsf{MoE}(\lambda, \mathcal{A}) = 1] \leq 2^{-\lambda^{C_{\mathsf{MoE}}}}$$

for all sufficiently large λ.

In the previous constructions of unclonable primitives [CLLZ21, LLQZ22], and also in our constructions, we require the freeloader adversaries or the LOCC adversary to output a vector from either $A + s$ or $A^{\perp} + s'$, depending on a random bit presented to them. However, since the *leakage* adversary can always guess this bit with probability $1/2$ and measure the coset state accordingly, we amplify the security by using multiple coset states to achieve negligible security. To simplify our proofs, we formally define this variant of the game, which is implicitly used in some form in [CLLZ21, LLQZ22].

Definition 14. *Define* $\mathsf{CosetGen}(1^\lambda)$ *to be the following algorithm, where we set* $c(\lambda) = 3 \cdot \lambda^{\lceil 1/C_{\mathsf{MoE}} \rceil}$.

1. *For* $i \in [c(\lambda)]$, *sample uniformly at random a subspace* A_i *of* $\mathbb{F}_2^\lambda$ *of dimension* $\lambda/2$ *and two elements* $s_i, s_i' \leftarrow \mathbb{F}_2^\lambda$.
2. *Output* $(A_i, s_i, s_i')_{i \in [c(\lambda)]}$.

We call the output of $\mathsf{CosetGen}$ *a coset tuple.*

Theorem 9 (Strong Monogamy-of-Entanglement Property for Coset States - Multiple Challange Version)**.** *Consider the following game between an adversary tuple* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ *and the challenger.*

$\underline{\mathsf{MoE} - \mathsf{MultiChal}(\lambda, \mathcal{A})}$

1. *Sample* $(A_i, s_i, s_i')_{i \in [c(\lambda)]} \leftarrow \mathsf{CosetGen}(1^\lambda)$.
2. *Submit* $\left\{ \left| A_{i, s_i, s_i'} \right\rangle \right\}_{i \in [c(\lambda)]}$ *to* $\mathcal{A}_0$.
3. $\mathcal{A}$ *outputs two (possibly entangled) registers* R_1, R_2.
4. *Sample* $r_1 \leftarrow \{0,1\}^{c(\lambda)}$ *and* $r_2 \leftarrow \{0,1\}^{c(\lambda)}$.
5. *For* $\ell \in \{1,2\}$, *run* $(v_{\ell,i})_{i \in [c(\lambda)]} \leftarrow \mathcal{A}_\ell(R_\ell, r_\ell, (A_i)_{i \in [c(\lambda)]})$.
6. *For* $\ell \in \{1,2\}$ *and all* $i \in [c(\lambda)]$, *check if* $v_{\ell,i} \in A_i + s_i$ *if* $(r_\ell)_i = 0$ *and if* $v_{\ell,i} \in A_i^{\perp} + s_i'$ *if* $(r_\ell)_i = 1$. *Output* 1 *if and only if all the checks pass. Otherwise, output* 0.

Then, there exists a constant $C_{\mathsf{MoE-MultChal}} > 0$ *such that for any adversary tuple* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$,

$$\Pr[\mathsf{MoE} - \mathsf{MultiChal}(\lambda, \mathcal{A}) = 1] \leq 2^{-\lambda^{C_{\mathsf{MoE-MultChal}}}}$$

for all sufficiently large λ.

Proof. Suppose there exists an adversary tuple $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ that wins $\mathsf{MoE} - \mathsf{MultiChal}$ with probability $\varepsilon(\lambda)$. Define Hyb_0 to be the original game $\mathsf{MoE} - \mathsf{MultiChal}(\lambda, \mathcal{A})$ and define Hyb_1 by modifying it as follows. The challenger samples r_1, r_2 so that there is an index $i \in [c(\lambda)]$ such that $(r_1)_i = 0$ and $(r_2)_i = 1$. Since sampling r_1, r_2 uniformly and independently satisfies this with probability $1 - (3/4)^{c(\lambda)}$, the distance between the two distributions is $(3/4)^{c(\lambda)}$, hence we get that $\mathcal{A}$ wins in Hyb_1 with probability at least $\varepsilon(\lambda) - (3/4)^{c(\lambda)}$.

For any $j^* \in [c(\lambda)]$, we define the adversary $\mathcal{A}^{j^*} = (\mathcal{A}_0^{j^*}, \mathcal{A}_1', \mathcal{A}_2')$ for MoE as follows.

$\underline{\mathcal{A}_0^{j^*}}$

On input a state ρ, sample r_1, r_2 so that there is an index $j^* \in [c(\lambda)]$ such that $(r_1)_{j^*} = 0$ and $(r_2)_{j^*} = 1$. Set $\rho_{j^*} = \rho$. For all $j \in [c(\lambda)] \setminus \{j^*\}$, sample a subspace A_j, elements $s_j, s'_j \leftarrow \mathbb{F}_2^\lambda$, then set $\rho_j = \left|A_{j,s_j,s'_j}\right\rangle$. Then, run $\mathcal{A}_0((\rho_j)_{j\in[c(\lambda)]})$ to obtain a bipartite state σ. Finally, output

$$((\sigma[1], (A_j)_{j\in[c(\lambda)]\setminus\{j^*\}}, j^*, r_1), (\sigma[2], (A_j)_{j\in[c(\lambda)]\setminus\{j^*\}}, j^*, r_2)).$$

$\underline{\mathcal{A}'_\ell \textbf{ for } \ell \in \{1,2\}}$

$\mathcal{A}'_\ell$ runs $\mathcal{A}_\ell$ on its own input and the subspace description A it obtains from the challenger. Output the j^*-th vector in the output of $\mathcal{A}'_\ell$.

Note that $\mathcal{A}'_\ell$ can correctly rearrange the input order when passing it to $\mathcal{A}_\ell$ since it knows j^*. Observe that for any fixed value of j^*, when $\mathcal{A}^{j^*}$ is run on a uniformly random coset state, the input to $\mathcal{A}$ is distributed the same as a coset state tuple obtained using $\mathsf{CosetGen}$. Therefore, $\mathcal{A}_\ell$ above output the correct vectors (i.e., in $v \in A_j + s_j$ or $A_j^\perp + s'_j$ depending on $(r_b)_j$ for all $j \in [c(\lambda)]$ and $b \in \{0,1\}$) simultaneously with probability at least $\varepsilon(\lambda) - (3/4)^{c(\lambda)}$, since above is a perfect simulation of $\mathcal{A}$ playing Hyb_1.

Finally, we construct an adversary $\mathcal{A}' = (\mathcal{A}'_0, \mathcal{A}''_1, \mathcal{A}''_2)$ for MoE as follows. $\mathcal{A}'_0$ samples j^* uniformly at random, and then simulates $\mathcal{A}_0^{j^*}$, but without sampling r_1, r_2. We similarly define $\mathcal{A}''_1$ and $\mathcal{A}''_2$, which uses the challenge strings r_1, r_2 given to them by the challenger. By above, $\mathcal{A}'$ obtains the correct vectors with probability $\varepsilon(\lambda) - (3/4)^{c(\lambda)}$ as argued above. Moreover, with probability $1/c(\lambda)$, independent of $\mathcal{A}'$ obtaining the correct vectors, j^* will be such that $(r_1)_{j^*} = 0$ and $(r_2)_{j^*} = 1$. Note that when j^* satisfies this and the vectors obtained by $\mathcal{A}'$ are correct, $\mathcal{A}'$ wins MoE. Hence, $\mathcal{A}'$ wins with probability at least $\frac{\varepsilon(\lambda)-(3/4)^{c(\lambda)}}{c(\lambda)}$. This completes the proof, by our choice of $c(\lambda) = 3 \cdot \lambda^{\lceil 1/C_{\mathsf{MoE}} \rceil}$ and by Theorem 8. □

Now, we show an LOCC leakage-resilience property for coset states. In the LOCC leakage game, a leakage adversary in possesion of the coset state tuple and a *main* adversary will execute an LOCC protocol. After the protocol is completed, the main adversary is presented with a random challenge string r, and is required to produce vectors in correct cosets depending on r.

Theorem 10 (LOCC Leakage Property for Coset States)**.** *Consider the following game between an LOCC adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ and the challenger.*

$\underline{\mathsf{Coset} - \mathsf{LOCC}(\lambda, \mathcal{A})}$

1. *Sample* $(A_i, s_i, s'_i)_{i\in[c(\lambda)]} \leftarrow \mathsf{CosetGen}(1^\lambda)$.
2. *Submit* $\left\{\left|A_{i,s_i,s'_i}\right\rangle\right\}_{i\in[c(\lambda)]}$ *to* $\mathcal{A}_{\mathsf{Leak}}$.
3. $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *execute their LOCC protocol.*
4. *After the LOCC protocol is over, challenger samples* $r \leftarrow \{0,1\}^{c(\lambda)}$ *and submits* $(A_i)_{i\in[c(\lambda)]}$ *and* r *to* $\mathcal{A}_{\mathsf{Main}}$.
5. $\mathcal{A}_{\mathsf{Main}}$ *outputs* $(v_i)_{i\in[c(\lambda)]}$.
6. *For all* $i \in [c(\lambda)]$*, check if* $v_i \in A_i + s_i$ *if* $(r)_i = 0$ *and if* $v_i \in A_i^\perp + s'_i$ *if* $(r)_i = 1$*. Output* 1 *if and only if all the checks pass. Otherwise, output* 0.

Then, there exists a constant $C_{\mathsf{LOCC}} > 0$ such that for any LOCC adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$,

$$\Pr[\mathsf{Coset} - \mathsf{LOCC}(\lambda, \mathcal{A}) = 1] \leq 2^{-\lambda^{C_{\mathsf{LOCC}}}}$$

for all sufficiently large λ.

We will prove the above result through a reduction to $\mathsf{MoE} - \mathsf{MultiChal}$, which we briefly sketch. In the reduction, since the adversary for $\mathsf{MoE} - \mathsf{MultiChal}$ will be in possession of the coset state tuple, it can simulate both $\mathcal{A}_{\mathsf{Leak}}$ and $\mathcal{A}_{\mathsf{Main}}$ to produce the final state of the latter. However, the MoE adversary needs to produce two registers that are capable of answering the challenges correctly simultaneously. Hence, during the simulation of the LOCC protcol, we run $\mathcal{A}_{\mathsf{Main}}$ many times each round to produce multiple copies of its state, culminating in two copies of its final state, which then we output. While the probability of obtaining another copy of the adversary's state during a round might be arbitrarily small, we show that on average this is not the case.

Proof. Suppose for a contradiction that there exists a $g(n)$-round LOCC adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins $\mathsf{Coset} - \mathsf{LOCC}$ with probability $2^{-0.1\lambda^{C_{\mathsf{MoE}-\mathsf{MultChal}}}}$. Without loss of generality, assume that the messages sent by both adversaries each round are of the same length and denote it as $k(n)$. We will construct an adversary $\mathcal{A}' = (\mathcal{A}'_0, \mathcal{A}'_1, \mathcal{A}'_2)$ that wins $\mathsf{MoE} - \mathsf{MultiChal}$ with probability $2^{-0.3\lambda^{C_{\mathsf{MoE}-\mathsf{MultChal}}}}$.

Let $\mathcal{P}$ denote the random variable that contains the transcript of the LOCC protocol $\mathcal{A}$ during $\mathsf{Coset} - \mathsf{LOCC}$ and let $\mathcal{P}^{-1}$ denote the same but with the last leakage message sent by $\mathcal{A}_{\mathsf{Leak}}$ removed. For some fixed value (ℓ^{-1}, m) of $\mathcal{P}^{-1}$, let $\rho_{\ell^{-1},m}$ denote the final state of $\mathcal{A}_{\mathsf{Main}}$, i.e., its state right before it receives the last leakage message $\ell_{g(n)}$ and the challenge string r, conditioned on $\mathcal{P}^{-1} = (\ell^{-}1, m)$. Note that the final state of $\mathcal{A}_{\mathsf{Main}}$ does not depend on $\ell_{g(n)}$. Define deterministic $f(w)$ as $f(w) = (A_i)_{i\in[c(\lambda)]}$ where $(A_i, s_i, s'_i)_{i\in[c(\lambda)]} = \mathsf{CosetGen}(1^\lambda; w)$ and similarly deterministic predicate P as the function that outputs 1 if and only if the given vectors are in the correct cosets according to r. Then, the winning probability of $\mathcal{A}$ can be written as

$$\mathbb{E}_{(w,(\ell,m))\leftarrow(W,\mathcal{P})}\left[\Pr_{r\leftarrow\{0,1\}^{c(\lambda)}}[P(\mathcal{A}_{\mathsf{Main}}(\rho_{\ell^{-1},m}, \ell_{g(n)}, f(w), r), w, r) = 1]\right]. \tag{1}$$

If we had two copies of $\rho_{\ell^{-1},m}$ for the same coset state tuple and transcript ℓ^{-1}, m, and ran the last step of $\mathcal{A}_{\mathsf{Main}}$ twice independently on independent challenge strings r_1, r_2, we would have that the probability of both copies *winning* simultaneously is

$$\begin{aligned}
&\mathbb{E}_{(w,(\ell,m))\leftarrow(W,\mathcal{P})}\left[\left(\Pr_{r_1\leftarrow\{0,1\}^{c(\lambda)}}[P(\mathcal{A}_{\mathsf{Main}}(\rho_{l^{-1},m}, \ell_{g(n)}, f(w), r_1), w, r_1) = 1]\right)\cdot\right.\\
&\qquad\left.\left(\Pr_{r_2\leftarrow\{0,1\}^{c(\lambda)}}[P(\mathcal{A}_{\mathsf{Main}}(\rho_{l^{-1},m}, \ell_{g(n)}, f(w), r_2), w, r_2) = 1]\right)\right]\\
&= \mathbb{E}_{(w,(\ell,m))\leftarrow(W,\mathcal{P})}\left[\left(\Pr_{r\leftarrow\{0,1\}^{c(\lambda)}}[P(\mathcal{A}_{\mathsf{Main}}(\rho_{l^{-1},m}, \ell_{g(n)}, f(w), r), w, r) = 1]\right)^2\right]
\end{aligned} \tag{2}$$

Then, since we have $(1) > 2^{-0.1\lambda^{C_{\mathsf{MoE}-\mathsf{MultChal}}}}$, we get $(2) > 2^{-0.2\lambda^{C_{\mathsf{MoE}-\mathsf{MultChal}}}}$ by Jensen's inequality.

We will construct an adversary $\mathcal{A}'_0$ for $\mathsf{MoE} - \mathsf{MultiChal}$ such that given only a single copy of the coset state tuple, it produces two copies of $\rho_{\ell^{-1},m}$ with probability at least $1/2$ (independently of any fixed value of the transcript). Note that we produce two copies of $\rho_{\ell^{-1},m}$ for the same ℓ^{-1}, m, since we only have one copy of the coset state tuple and running $\mathcal{A}_{\mathsf{Leak}}$ modifies this state.

Therefore, we can run $\mathcal{A}_{\mathsf{Leak}}$ once each round, hence we run it on a single value of a message m_i from $\mathcal{A}_{\mathsf{Main}}$ and receive only a single leakage ℓ_i each round. $\mathcal{A}'_0$ outputs $((\rho_{\ell^{-1},m}, \ell_{g(n)}), (\rho_{\ell^{-1},m}, \ell_{g(n)}))$ as its bipartite state output. Finally, $\mathcal{A}'_1$ and $\mathcal{A}'_2$ both simulate $\mathcal{A}_{\mathsf{Main}}$ on the state they receive, along with the subspace descriptions and the random challenge string they receive from the challenger. Then, winning corresponds exactly to Equation (2), therefore we get that $\mathcal{A}'$ wins $\mathsf{MoE}-\mathsf{MultiChal}$ with probability $2^{-0.2\lambda^{C_{\mathsf{MoE}-\mathsf{MultChal}}}} \cdot \frac{1}{2} > 2^{-\lambda^{C_{\mathsf{MoE}-\mathsf{MultChal}}}}$, which is a contradiction by Theorem 9.

Now we will show how to construct such an adversary. First, note that the final state (before receiving the final leakage message) ρ of the main adversary is output together with a message $m_{g(n)}$, by running $\mathcal{A}_{\mathsf{Main}}$ on its previous state $\rho_{g(n)-1}$ and the previous leakage message $\ell_{g(n)-1}$. In turn, $\rho_{g(n)-1}$, $m_{g(n)-1}$ and so on, all the way down to the initial state of $\mathcal{A}_{\mathsf{Main}}$ are sampled similarly. Consider any input σ to $\mathcal{A}_{\mathsf{Main}}$, i.e. a previous state and a leakage message[4]. Let $\sum_{x\in\{0,1\}^{k(n)}} p_x|x\rangle\langle x| \otimes \tau_x$ denote the output of $\mathcal{A}_{\mathsf{Main}}(\sigma)$. Then, we claim that given $a(n)$ copies of σ and a fixed value x, we can produce $d(n)$ extra copies of τ_x for any $d(n)$ with probability $(1/2)^{1/g(n)}$ averaged over x, where

$$a(n) = \frac{2^{k(n)+1} \cdot d^2(n)}{1-(1/2)^{(1/g(n))-1}}.$$

Starting with $d(n) = 1$ for the last round, we can calculate the number of copies needed all the way down to the first level. While grows large with every round, the total number of copies of the initial state needed is bounded since we have $g(n)$ rounds. Therefore, we can construct a valid $\mathcal{A}'_0$ as follows. For each round, it first simulates $\mathcal{A}_{\mathsf{Main}}$ to obtain a message x and a state. Then, it keeps running $\mathcal{A}_{\mathsf{Main}}$ repeatedly until obtains the same message x again, in which case it has also obtained the required copy of the state. It repeats this procedure many times to obtain sufficiently many copies for the next round. Finally, it runs $\mathcal{A}_{\mathsf{Leak}}$ on the coset state tuple along with the message x to produce the leakage. Repeating this simulation until the last round shows that we can obtain two copies of $\rho_{\ell^{-1},m}$ in a bounded amount of time. Note that by above, the many copy preparation procedure succeeds with probability $(1/2)^{1/g(n)}$ for each round, independently of succeeding in the previous rounds since we made the claim above for any input σ. Hence, we will obtain two copies of $\rho_{\ell^{-1},m}$ with probability $1/2$ as desired.

Lastly, we prove our claim that $a(n)$ copies of the input to $\mathcal{A}_{\mathsf{Main}}$ is sufficient to produce $d(n)$ extra copies of its output. While the desired output x that we want for it to reoccur might have arbitrarily small probability, in which case it would take arbitrarily long to obtain the same state again, this happens *rarely*. More formally, define the set GOOD to be all $x \in \{0,1\}^{k(n)}$ such that

$$p_x > 2^{-k(n)}(1-(1/2)^{(1/g(n))-1}).$$

Then, a simple calculation shows that $\sum_{x\in\mathsf{GOOD}} p_x > (1/2)^{(1/g(n))-1}$. We have that the probability of getting the first outcome x again $d(n)$ many times in $a(n)$ trials, averaged over all x, is at least

$$\sum_{x\in\mathsf{GOOD}} p_x(1-(1-p_x)^{a(n)/d(n)})^{(d(n))}.$$

A simple calculation shows that this value is at least $(1/2)^{1/g(n)}$ as desired. □

Finally, similar to the computational MoE theorem shown by [CLLZ21, Theorem 4.19], we prove a computational version of the above theorem, where the adversary is also presented with a (obfuscated) program that checks for membership in the cosets.

Theorem 11 (Computational LOCC Leakage Property for Coset States)**.** *Consider the following game between an LOCC adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ and the challenger.*

[4]We bundle the leakage message in the state σ

$\underline{\mathsf{Coset}-\mathsf{CompLOCC}(\lambda,\mathcal{A})}$

1. *Sample* $(A_i, s_i, s_i')_{i\in[c(\lambda)]} \leftarrow \mathsf{CosetGen}(1^\lambda)$.
2. *For* $i \in [c(\lambda)]$,
   2.1. *Sample* $\mathsf{OP}_i^0 \leftarrow i\mathcal{O}(A_i + s_i)$.
   2.2. *Sample* $\mathsf{OP}_i^1 \leftarrow i\mathcal{O}(A_i^\perp + s_i')$.
3. *Submit* $\left\{\left|A_{i,s_i,s_i'}\right\rangle\right\}_{i\in[c(\lambda)]}, (\mathsf{OP}_i^0, \mathsf{OP}_i^1)_{i\in[c(\lambda)]}$ *to* $\mathcal{A}_{\mathsf{Leak}}$.
4. $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *execute their LOCC protocol.*
5. *After the LOCC protocol is over, challenger samples* $r \leftarrow \{0,1\}^{c(\lambda)}$ *and submits* $(A_i)_{i\in[c(\lambda)]}$ *and* r *to* $\mathcal{A}_{\mathsf{Main}}$.
6. $\mathcal{A}_{\mathsf{Main}}$ *outputs* $(v_i)_{i\in[c(\lambda)]}$.
7. *For all* $i \in [c(\lambda)]$, *check if* $v_i \in A_i + s_i$ *if* $(r)_i = 0$ *and if* $v_i \in A_i^\perp + s_i'$ *if* $(r)_i = 1$. *Output* 1 *if and only if all the checks pass. Otherwise, output* 0.

Then, assuming the existence of $i\mathcal{O}$ *and one-way functions, for any QPT LOCC adversary pair* $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *we have that*

$$\Pr[\mathsf{Coset}-\mathsf{CompLOCC}(\lambda,\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda).$$

If we assume the existence of subexponentially-secure $i\mathcal{O}$ *and one-way functions, then there exists a constant* $C_{\mathsf{CompLOCC}} > 0$ *such that for any QPT LOCC adversary* $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *we have that*

$$\Pr[\mathsf{Coset}-\mathsf{CompLOCC}(\lambda,\mathcal{A}) = 1] \leq 2^{-\lambda^{C_{\mathsf{CompLOCC}}}}$$

for all sufficiently large λ.

Our proof closely follows the reduction from the computational monogamy-of-entanglement theorem to the information-theoretic version given in [CLLZ21], generalized to the multiple coset states in a straightforward manner. The main idea of the proof is to replace all obfuscated membership checking programs with functionally equivalent programs that instead use subspace hiding obfuscation for the subspaces $A_i, A_i^\perp$. Then, we can further replace these with subspace hiding obfuscation for random superspaces, which eventually allows us to remove the membership checking programs and reduce to $\mathsf{Coset}-\mathsf{LOCC}$. We give our proof in full detail for completeness.

Proof. We will only prove the subexponential case, and the other case follows similarly. Assume that $i\mathcal{O}$ and shO are $2^{-(\lambda/2)^{C_{\mathsf{LOCC}}}}$-secure. Suppose for a contradiction that there exists an adversary $\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}}$ that wins $\mathsf{Coset}-\mathsf{CompLOCC}$ with probability $2^{-0.5\cdot(\lambda/2)^{C_{\mathsf{LOCC}}}}$.

We prove security through a series of hybrids, each of which is constructed by modifying the previous one. Note that while we change the way the membership checking programs are computed, the challenger still tests the vectors output by the adversary according to A_i, s_i, s_i'. Also note that obfuscated programs are assumed to be padded to appropriate size.

$\underline{\mathsf{Hyb}_0}$**:** The original game $\mathsf{Coset}-\mathsf{CompLOCC}(\lambda,\mathcal{A})$.

$\underline{\mathsf{Hyb}_1}$**:** We now compute OP_i^0 as $\mathsf{OP}_i^0 \leftarrow i\mathcal{O}(\mathsf{shO}(A_i)(\cdot - s_i))$[5] and compute OP_i^1 as $\mathsf{OP}_i^1 \leftarrow i\mathcal{O}(\mathsf{shO}(A_i^\perp)(\cdot - s_i'))$ for all $i \in [c(\lambda)]$.

$\underline{\mathsf{Hyb}_{2,j}}$ **for** $j \in [c(\lambda)]$**:** For all $i \leq j$, sample a uniformly random subspace $B_i \subseteq \mathbb{F}_2^\lambda$ such that $B_i \supseteq A_i$ and $\dim(B_i) = 3\lambda/4$. Now we compute OP_i^0 as $\mathsf{OP}_i^0 \leftarrow i\mathcal{O}(\mathsf{shO}(B_i)(\cdot - s_i))$

$\underline{\mathsf{Hyb}_{3,j}}$ **for** $j \in [c(\lambda)]$**:** For all $i \leq j$, sample a uniformly random subspace $C_i \subseteq \mathbb{F}_2^\lambda$ such that $C_i \subseteq A_i$ and $\dim(C_i) = \lambda/4$. Compute OP_i^1 as $\mathsf{OP}_i^1 \leftarrow i\mathcal{O}(\mathsf{shO}(C_i^\perp)(\cdot - s_i'))$.

$\underline{\mathsf{Hyb}_{4,j}}$ **for** $j \in [c(\lambda)]$**:** For all $i \leq j$, sample $u_i \leftarrow B_i$. Compute OP_i^0 as $\mathsf{OP}_i^0 \leftarrow i\mathcal{O}(\mathsf{shO}(B_i)(\cdot - u_i - s_i))$.

$\underline{\mathsf{Hyb}_{5,j}}$ **for** $j \in [c(\lambda)]$**:** For all $i \leq j$, sample $u_i' \leftarrow C_i^\perp$. Compute OP_i^1 as $\mathsf{OP}_i^1 \leftarrow i\mathcal{O}(\mathsf{shO}(C_i^\perp)(\cdot - u_i' - s_i'))$.

$\underline{\mathsf{Hyb}_6}$**:** Instead of submitting $\left\{\left|A_{i,s_i,s_i'}\right\rangle\right\}_{i\in[c(\lambda)]}, (\mathsf{OP}_i^0, \mathsf{OP}_i^1)_{i\in[c(\lambda)]}$ to $\mathcal{A}_{\mathsf{Leak}}$, the challenger now submits $\left\{\left|A_{i,s_i,s_i'}\right\rangle\right\}_{i\in[c(\lambda)]}, (B_i, C_i, u_i + s_i, u_i' + s_i')_{i\in[c(\lambda)]}$ to $\mathcal{A}_{\mathsf{Leak}}$.

Claim 1. $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$.

By correctness of shO, all the obfuscated programs in these hybrids have the same functionality. The result follows by the security of $i\mathcal{O}$.

Claim 2. $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_{2,1}$ *and* $\mathsf{Hyb}_{2,j} \approx \mathsf{Hyb}_{2,j+1}$ *for all* $j \in [c(\lambda) - 1]$.

By the security of shO, we have $\mathsf{shO}(A_i) \approx \mathsf{shO}(B_i)$. Since the wrapper $(\cdot - s_i)$ and the outer obfuscation are constructed efficiently, the result follows. Note that while the adversary also has access to the coset state, we can still invoke subspace hiding obfuscation security since shO is secure even when the subspace A_i is selected by the adversary (Definition 8).

Claim 3. $\mathsf{Hyb}_{2,c(\lambda)} \approx \mathsf{Hyb}_{3,1}$ *and* $\mathsf{Hyb}_{3,j} \approx \mathsf{Hyb}_{3,j+1}$ *for all* $j \in [c(\lambda) - 1]$.

Note that $C_i \subseteq A_i$ and $\dim(C_i) = \lambda/4$ implies $C_i^\perp \supseteq A_i^\perp$ and $\dim(C_i) = 3\lambda/4$. The result follows from the same argument as Claim 2.

Claim 4. $\mathsf{Hyb}_{3,c(\lambda)} \approx \mathsf{Hyb}_{4,1}$ *and* $\mathsf{Hyb}_{4,j} \approx \mathsf{Hyb}_{4,j+1}$ *for all* $j \in [c(\lambda) - 1]$.

Observe that $v - s_i \in B_i$ if and only if $v - u_i - s_i \in B_i$ since $u_i \in B_i$. Since the obfuscated programs have the same functionality, the result follows from the security of $i\mathcal{O}$.

Claim 5. $\mathsf{Hyb}_{4,c(\lambda)} \approx \mathsf{Hyb}_{5,1}$ *and* $\mathsf{Hyb}_{5,j} \approx \mathsf{Hyb}_{5,j+1}$ *for all* $j \in [c(\lambda) - 1]$.

Same argument as above.

Claim 6. *The winning probability of the adversary in* Hyb_6 *is higher than that in* $\mathsf{Hyb}_{5,c(\lambda)}$.

[5] $\mathsf{shO}(A_i)(-s_i)$ means the program that takes an input v, subtracts s_i and passes it to $\mathsf{shO}(A_i)$.

Since the adversary can compute the obfuscated programs itself using $(B_i, C_i, u_i + s_i, u'_i + s'_i)_{i\in[c(\lambda)]}$, the result follows.

Finally, by above and by our choice of parameters for $i\mathcal{O}$ and shO, we get that $\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}}$ wins in Hyb_6 with probability at least $\varepsilon(\lambda) = 2^{-0.6\cdot(\lambda/2)^{C_{\mathsf{LOCC}}}}$. That is, $\Pr[\mathsf{Hyb}_6 = 1] > \varepsilon(\lambda) = 2^{-0.6\cdot(\lambda/2)^{C_{\mathsf{LOCC}}}}$.

We cannot reduce directly to $\mathsf{Coset} - \mathsf{LOCC}$ yet because we cannot sample B_i, C_i since we do not have A_i. To remedy this, we first fix B_i and C_i as follows. Let B^* and C^* denote the unique subspaces of dimensions $3\lambda/4$ and $\lambda/4$ where the last $\lambda/4$ and $3\lambda/4$ are all zeroes, respectively. Consider the modified version $\mathcal{G}$ of $\mathsf{Coset} - \mathsf{CompLOCC}$ where we sample uniformly at random a subspace A_i^* of dimension $\lambda/2$ such that $C^* \subseteq A_i^* \subseteq B^*$, and use A_i^* instead of A_i.

Lemma 2 ([CLLZ21])**.** *Sample $A_i, B_i, C_i, s_i, s'_i, u_i, u'_i$ as above. Sample uniformly at random an isomorphism $\mathcal{T}_i$ such that $\mathcal{T}_i(C^*) = C_i$ and $\mathcal{T}_i(B^*) = B_i$. Sample $u_i^* \leftarrow B^*$ and $u_i'^* \leftarrow (C^*)^\perp$. Then,*

$$\left(\bigotimes_{i\in[c(\lambda)]]} U_{\mathcal{T}_i}\left|A^*_{i,s_i,s'_i}\right\rangle, (\mathcal{T}_i(B^*), \mathcal{T}_i(C^*), \mathcal{T}_i(s_i + u_i^*), (\mathcal{T}_i^{-1})^T(s'_i + u_i'^*), \mathcal{T}_i(A_i))_{i\in[c(\lambda)]]}\right) \equiv$$
$$\left(\bigotimes_{i\in[c(\lambda)]]} \left|A_{i,s_i,s'_i}\right\rangle, (B_i, C_i, s_i + u_i, s'_i + u'_i, A_i)_{i\in[c(\lambda)]]}\right).$$

By the above lemma, we see that there exist $\mathcal{A}'_{\mathsf{Main}}, \mathcal{A}'_{\mathsf{Leak}}$[6] that wins the modified game with probability $\varepsilon(\lambda)$. Finally, we reduce the modified game $\mathcal{G}$ to $\mathsf{Coset} - \mathsf{LOCC}$ with security parameter $\lambda/2$. Observe that $C^* \subseteq A_i^* \subseteq B^*$ implies that the last $\lambda/4$ components of A_i^* are zeroes, while the first $\lambda/4$ are completely unrestricted. Therefore, sampling A_i^* is equivalent to sampling uniformly at random a subspace A_i^{**} of $\mathbb{F}_2^{\lambda/2}$ of dimension $\lambda/4$, then adding all vectors $v \in \mathbb{F}_2^{\lambda/4}$ at the beginning and $0^{\lambda/4}$ at the end of each vector. Similarly, sampling $s_i \leftarrow \mathbb{F}_2^\lambda$ is equivalent to sampling $w_i, q_i \leftarrow \mathbb{F}_2^{\lambda/4}$ and $s_i^{**} \leftarrow \mathbb{F}_2^{\lambda/2}$ and outputting $w_i||s_i^{**}||q_i$. Therefore, given the state $\left|A^{**}_{i,s_i^{**},s_i^{**\prime}}\right\rangle$ where A_i^{**} is sampled as above, we can sample $s_i^{**}, s_i^{**\prime}, w'_i \leftarrow \mathbb{F}_2^{\lambda/4}$ and construct $\left(\sum_{v\in\mathbb{F}_2^{\lambda/4}}(-1)^{\langle v,w'_i\rangle}|v + w_i\rangle\right) \otimes \left(\left|A^{**}_{i,s_i^{**},s_i^{**\prime}}\right\rangle\right) \otimes |q_i\rangle$ which is distributed exactly as $\left|A^*_{i,s_i^*,s_i^{*\prime}}\right\rangle$. By similarly converting the output vectors, it is easy to see that we can construct $\mathcal{A}''_{\mathsf{Main}}, \mathcal{A}''_{\mathsf{Leak}}$ that wins $\mathsf{Coset} - \mathsf{LOCC}$ with probability $\varepsilon(\lambda) > 2^{-(\lambda/2)^{C_{\mathsf{LOCC}}}}$, which is a contradiction by Theorem 10. □

5 Public-Key Encryption with Key Protection

In this section, we introduce the concept of public-key encryption schemes with key protection and define various security models for it.

Definition 15 (Public-key encryption with key protection)**.** *A public-key encryption scheme* PKE *with key protection consists of the following efficient algorithms.*

- $\mathsf{Setup}(1^\lambda)$: *Outputs classical secret key sk and a classical public key pk.*
- $\mathsf{QKeyGen}(sk)$: *Takes the secret key sk, outputs a quantum key R_{dec}.*
- $\mathsf{Enc}(pk, m)$*: Takes the public key and a message m, returns an encryption of m.*

[6]The adversary simply samples $\mathcal{T}_i$ and simulates $\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}}$ with appropriate conversions using $\mathcal{T}_i$.

- $\mathsf{Dec}(R_{\mathsf{key}}, ct)$: *Takes a quantum secret key register and a ciphertext ct, outputs decryption of ct.*

We require that the scheme satisfies correctness. Let $\mathcal{M}$ denote the message space.

Correctness *For all messages $m \in \mathcal{M}$,*

$$\Pr\left[\mathsf{Dec}(R_{\mathsf{key}}, ct) = m : \begin{array}{c} sk, pk \leftarrow \mathsf{Setup}(1^\lambda) \\ R_{\mathsf{key}} \leftarrow \mathsf{QKeyGen}(sk) \\ ct \leftarrow \mathsf{Enc}(pk, m) \end{array}\right] = 1.$$

We can also define a relaxed correctness notion where we require correct decryption with probability $1 - \mathsf{negl}(\lambda)$. However, our constructions will satisfy perfect correctness.

As argued by [CLLZ21], the correctness property implies, via the *As Good As New Lemma* (Lemma 1), that we can efficiently implement the decryption in a way such that we *rewind* after decryption and the key register is not disturbed (or negligibly disturbed in the relaxed correctness regime). We will assume that it is implemented as such and therefore correctness implies that the protected key can be used to correctly decrypt any polynomial number of ciphertexts.

We first reproduce the two anti-piracy security[7] definitions of [CLLZ21]. In the first one, we require that an adversary cannot split a quantum key in a way that both registers can be used to simultaneously decrypt encryptions of random messages.

Definition 16 (Anti-piracy security for public-key encryption - random challenge message [CLLZ21])**.** *Consider the following game between an adversary $\mathcal{A}$ and the challenger.*

$\underline{\mathsf{PKE} - \mathsf{AntiPiracy} - \mathsf{Guess}(\lambda, \mathcal{A})}$

1. *Sample $pk, sk \leftarrow \mathsf{PKE.Setup}(1^\lambda)$.*
2. *Sample $R_{\mathsf{key}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$ and submit pk, R_{key} to $\mathcal{A}$.*
3. *$\mathcal{A}$ gets access to R_{dec} and pk and it produces a pair of registers (R_1, R_2).*
4. *The challenger samples two challenge messages $m_1^*, m_2^* \leftarrow \mathcal{M}$.*
5. *The challenger computes $ct_1 \leftarrow \mathsf{PKE.Enc}(pk, m_1^*)$ and $ct_2 \leftarrow \mathsf{PKE.Enc}(pk, m_2^*)$.*
6. *The challenger runs $m_1' \leftarrow \mathsf{U}_{\mathsf{univ}}(R_1, ct_1)$ and $m_2' \leftarrow \mathsf{U}_{\mathsf{univ}}(R_2, ct_2)$.*
7. *Output 1 if and only if both ${m_1}' = m_1^*$ and ${m_2}' = m_2^*$.*

A public-key encryption scheme PKE *with key protection is said to satisfy* random challenge message anti-piracy security *if for all QPT adversaries $\mathcal{A}$,*

$$\Pr[\mathsf{PKE} - \mathsf{AntiPiracy} - \mathsf{Guess}(\lambda, \mathcal{A}) = 1] \leq \frac{1}{|\mathcal{M}|} + \mathsf{negl}(\lambda).$$

We call each register R_1, R_2 a freeloader or a decryptor. They are each interpreted as the description of some quantum circuit Φ (with some *hardwired* state ρ) that we run using the universal quantum circuit $\mathsf{U}_{\mathsf{univ}}$, to obtain $\Phi(\rho, ct)$.

We can define a variation of the above security game, called *identical challenge mode*, where we present both freeloaders with the same challenge ciphertext. [AKL23] show that security in the independent challenge mode implies security in the identical challenge mode. However, we show that the other direction is not true.

[7] Public key-encryption with anti-piracy security is also called single-decryptor encryption in the literature.

Theorem 12. *Let* PKE *be a public-key encryption scheme with key protection that satisfies random challenge message anti-piracy security in the identical challenge mode. Then,* PKE' *constructed below satisfies random challenge message anti-piracy security in the identical challenge mode but not in the independent challenge mode.*

$\underline{\mathsf{PKE}'.\mathsf{Setup}(1^\lambda)}$

1. $pk_0, sk_0 \leftarrow \mathsf{PKE.Setup}(1^\lambda)$.
2. $pk_1, sk_1 \leftarrow \mathsf{PKE.Setup}(1^\lambda)$.
3. Output $(pk_0, pk_1), (sk_0, sk_1)$.

$\underline{\mathsf{PKE}'.\mathsf{QKeyGen}((sk_0, sk_1))}$

1. Output $\mathsf{PKE.QKeyGen}(sk_0), \mathsf{PKE.QKeyGen}(sk_1)$.

$\underline{\mathsf{PKE}'.\mathsf{Enc}(pk, m)}$

1. Sample $c \leftarrow \{0,1\}$.
2. Output $(\mathsf{PKE.Enc}(pk_c, m), c)$.

$\underline{\mathsf{PKE}'.\mathsf{Dec}((R_{\mathsf{key},0}, R_{\mathsf{key},1}), (ct, c))}$

1. Output $\mathsf{PKE.Dec}(R_{\mathsf{key},c}, ct)$.

Proof. In the identical challenge mode, both freeloaders receive encryptions under the same public key pk_c. Therefore, by anti-piracy security of PKE, we get that PKE' is also anti-piracy secure in the identical challenge mode.

However, the following adversary $\mathcal{A}$ breaks independent challenge anti-piracy security for PKE'. $\mathcal{A}$ outputs $R_{\mathsf{key},0}$ to the first register and $R_{\mathsf{key},1}$ to the second register. Observe that with probability $1/4$, the first freeloader will receive an encryption under pk_0 and the second one will receive one under pk_1. In this case, both freeloaders will be able to decrypt their challenge ciphertexts by the correctness of PKE. Therefore, $\mathcal{A}$ wins anti-piracy security game in the independent challenge mode with probability $1/4$. □

In the second definition, we require that an adversary cannot split a quantum key in a way that both registers can be used to simultaneously distinguish encryptions of challenge messages. Note the baseline success probability is $1/2$ since the adversary can output the key to one of the registers, and for the other register it outputs an adversary that randomly guesses the challenge bit. By the correctness of the key, this attack succeeds with probability $1/2$.

Definition 17 (Anti-piracy security for public-key encryption - CPA-style [CLLZ21])**.** *Consider the following game between an adversary* $\mathcal{A}$ *and the challenger.*

$\mathsf{PKE} - \mathsf{AntiPiracy} - \mathsf{CPA}(\lambda, \mathcal{A})$

1. *Sample* $pk, sk \leftarrow \mathsf{PKE.Setup}(1^\lambda)$.
2. *Sample* $R_{\mathsf{key}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$ *and submit* pk, R_{key} *to* $\mathcal{A}$.
3. $\mathcal{A}$ *gets access to* R_{dec} *and* pk*, and it produces a pair of registers* (R_1, R_2) *and two messages* m_0, m_1.
4. *The challenger samples two challenge bits* $b_1, b_2 \leftarrow \{0,1\}$.
5. *The challenger computes* $ct_1 \leftarrow \mathsf{PKE.Enc}(pk, m_{b_1})$ *and* $ct_2 \leftarrow \mathsf{PKE.Enc}(pk, m_{b_2})$.
6. *The challenger runs* $b_1' \leftarrow \mathsf{U}_{\mathsf{univ}}(R_1, ct_1)$ *and* $b_2' \leftarrow \mathsf{U}_{\mathsf{univ}}(R_2, ct_2)$.
7. *Output* 1 *if and only if both* $b_1{}' = b_1$ *and* $b_2' = b_2$.

A public-key encryption scheme PKE *with key protection is said to satisfy* CPA-style anti-piracy security *if for all QPT adversaries* $\mathcal{A}$*,*

$$\Pr[\mathsf{PKE} - \mathsf{AntiPiracy} - \mathsf{CPA}(\lambda, \mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Lemma 3 ([CLLZ21])**.** *Suppose a public-key encryption scheme satisfies CPA-style anti-piracy security. Then, it also satisfies regular CPA-security.*

For anti-piracy security, unlike the classical case, CPA-style security is not known to imply random challenge message security since there are two adversaries that need to decrypt simultaneously. See [CLLZ21, Appendix D.4] for a discussion.

Similar to the previous case, we can define a variation of the above security game, called *identical challenge mode*, where use the same randomness to compute both of the challenge ciphertexts.

Theorem 13. *Let* PKE *be a public-key encryption scheme with key protection that satisfies CPA-style anti-piracy security in the identical challenge mode. Then,* PKE′ *constructed below Theorem 12 satisfies CPA-style anti-piracy security in the identical challenge mode but not in the independent challenge mode.*

Proof. In the identical challenge mode, both freeloaders receive encryptions under the same public key pk_c. Therefore, by anti-piracy security of PKE, we get that PKE′ is also anti-piracy secure in the identical challenge mode.

However, the following adversary $\mathcal{A}$ breaks independent challenge anti-piracy security for PKE′. $\mathcal{A}$ outputs $R_{\mathsf{key},0}$ to the first register and $R_{\mathsf{key},1}$ to the second register. Let c_0, c_1 denote the random bits contained in the challenge ciphertexts.

- If $c_0 = 0, c_1 = 0$, the first freeloader uses its key to decrypt, the second freeloader outputs a random prediction.
- If $c_0 = 0, c_1 = 1$, then both freeloaders can use their keys to correctly decrypt their challenge ciphertexts.
- If $c_0 = 1, c_1 = 0$, then both freeloaders output random predictions.
- If $c_0 = c_1 = 1$,the second freeloader uses its key to decrypt, the first freeloader outputs a random prediction.

Observe that $\mathcal{A}$ wins the independent challenge anti-piracy security game with probability

$$\frac{1}{4}\cdot\frac{1}{2}+\frac{1}{4}\cdot 1+\frac{1}{4}\cdot\frac{1}{4}+\frac{1}{4}\cdot\frac{1}{2}=\frac{9}{16}.$$

□

We now introduce the notion of LOCC leakage-resilience. Similar to anti-piracy, we can define two variants: random challenge message and CPA style. In both definitions, we assume that the last message in the LOCC protocol is from the leakage adversary to the main adversary. Therefore, if the number of rounds is even, the first message will be from the main adversary, and otherwise it will be from the leakage adversary. We also define a relaxation of our model where the LOCC adversary is only allowed a bounded, $< g(\lambda)$, number of rounds[8].

Definition 18 (LOCC Leakage-resilience for public-key encryption - random challenge message)**.** *Consider the following game between the challenger and an LOCC adversary* $\mathcal{A}=(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$*.*

$\underline{\mathsf{PKE}-\mathsf{LOCC}-\mathsf{Guess}(\lambda,\mathcal{A})}$

1. *Sample* $pk, sk \leftarrow \mathsf{PKE.Setup}(1^\lambda)$*.*
2. *Sample* $R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$ *and submit* pk, R_{dec} *to* $\mathcal{A}_{\mathsf{Leak}}$*.*
3. $(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$ *execute their LOCC protocol.*
4. *After the LOCC protocol is over, challenger samples* $m \leftarrow \mathcal{M}$ *and* $ct \leftarrow \mathsf{PKE.Enc}(pk,m)$*. It submits* ct *to* $\mathcal{A}_{\mathsf{Main}}$*.*
5. $\mathcal{A}_{\mathsf{Main}}$ *outputs a guess* $m' \in \{0,1\}$*.*
6. *The challenger outputs* 1 *if and only if* $m'=m$*.*

A public-key encryption scheme PKE *with key protection is said to satisfy* random challenge message $g(\lambda)$-round LOCC leakage-resilience *if for any QPT* $g(\lambda)$*-LOCC adversary* $(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$*,*

$$\Pr[\mathsf{PKE}-\mathsf{LOCC}-\mathsf{Guess}(\lambda,\mathcal{A})=1]\leq\frac{1}{|\mathcal{M}|}+\mathsf{negl}(\lambda).$$

If PKE *satisfies the above for any polynomial* $g(\lambda)$*, we simply say that it satisfies* random challenge message LOCC leakage-resilience*.*

Definition 19 (LOCC Leakage-resilience for public-key encryption - CPA-style)**.** *Consider the following game between the challenger and an LOCC adversary* $\mathcal{A}=(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$*.*

$\underline{\mathsf{PKE}-\mathsf{LOCC}-\mathsf{CPA}(\lambda,\mathcal{A})}$

1. *Sample* $pk, sk \leftarrow \mathsf{PKE.Setup}(1^\lambda)$*.*
2. *Sample* $R_{\mathsf{key}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$ *and submit* pk, R_{dec} *to* $\mathcal{A}_{\mathsf{Leak}}$*.*
3. $(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$ *execute their LOCC protocol.*
4. *After the LOCC protocol is over,* $\mathcal{A}_{\mathsf{Main}}$ *outputs two messages* m_0, m_1*.*

[8]Note that $g(\lambda)$ is implicitly polynomial since we are considering QPT adversaries.

5. *Challenger samples* $b \leftarrow \{0,1\}$ *and* $ct \leftarrow \mathsf{PKE.Enc}(pk, m_b)$. *It submits* ct *to* $\mathcal{A}_{\mathsf{Main}}$.

6. $\mathcal{A}_{\mathsf{Main}}$ *outputs a guess* $b' \in \{0,1\}$.

7. *The challenger outputs* 1 *if and only if* $b' = b$.

A public-key encryption scheme PKE *with key protection is said to satisfy* CPA-style $g(\lambda)$-round LOCC leakage-resilience *if for any QPT* $g(\lambda)$*-round LOCC adversary* $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$,

$$\Pr[\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}(\lambda, \mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

If PKE *satisfies the above for any polynomial* $g(\lambda)$*, we simply say that it satisfies* CPA-style LOCC leakage-resilience.

We make the following observations regarding these definitions.

Remark 1. *$g(\lambda)$-round LOCC leakage-resilience implies $h(\lambda)$-round LOCC leakage-resilience for any $h(\lambda) < g(\lambda)$.*

Remark 2. *$g(\lambda)$-round LOCC leakage-resilience for $g(\lambda) > 1$ can be considered an adaptive leakage model: The main adversary can send the leakage adversary description of an adaptive leakage circuit, which the leakage adversary can execute on the public key pk and the quantum key using a universal quantum circuit.*

Lemma 4. *Suppose a public-key encryption scheme satisfies CPA-style 1-round LOCC leakage-resilience. Then, it also satisfies regular CPA-security.*

Proof. Obvious: consider the leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ that sends pk as its only message. □

5.1 Relationship Between CPA-style and Random Challenge Message Leakage-Resilience

In this section, we show that similar to the classical case, CPA-style security implies security against random challenge messages in the LOCC leakage model. Then, we show that any scheme satisfying the latter can be used to construct a scheme satisfying the former, by using randomness extractors (more specifically, Goldreich-Levin bits).

Theorem 14. *Let* PKE *be a public-key encryption scheme with key protection that satisfies CPA-style $g(\lambda)$-round LOCC leakage-resilience. Then, it also satisfies random challenge message $g(\lambda)$-round LOCC leakage-resilience.*[9]

The results follows from the standard reduction.

Proof. Suppose for a contradiction that there exists a $g(\lambda)$-round QPT LOCC adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the random challenge message LOCC leakage-resilience game with probability $1/q(\lambda)$ for infinitely many values of $\lambda > 0$ where $q(\cdot)$ is a polynomial. We construct an adversary $\mathcal{A}' = (\mathcal{A}'_{\mathsf{Main}}, \mathcal{A}'_{\mathsf{Leak}})$ for the CPA-style LOCC leakage-resilience game as follows.

We define $\mathcal{A}'_{\mathsf{Leak}}$ to be the same as $\mathcal{A}_{\mathsf{Leak}}$. $\mathcal{A}'_{\mathsf{Main}}$ is defined to be the same as $\mathcal{A}_{\mathsf{Main}}$ until the end of the LOCC protocol, but we only change the challenge-response part of $\mathcal{A}_{\mathsf{Main}}$. After the protocol is over, $\mathcal{A}'_{\mathsf{Main}}$ samples two random messages $m_0, m_1 \leftarrow \mathcal{M}$. It outputs (m_0, m_1) as its choice of challenge messages and also saves them in its state. Then, when the challenge ciphertext

[9] We make the standard assumption that the message space $\mathcal{M}$ is of superpolynomial size.

ct is presented, it runs $\mathcal{A}_{\mathsf{Main}}$ on ct (and its state) to obtain a guess m'. If $m' = m_0$ it outputs 0, if $m' = m_1$ it outputs 1, otherwise it outputs a random bit $b'' \leftarrow \{0,1\}$.

Assume $m_0 \neq m_1$, which is without loss of generality since it happens with overwhelming probability.

Conditioned on the challenge bit value $b = 0$, $\mathcal{A}'$ perfectly simulates $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{Guess}(\lambda, \mathcal{A})$. Therefore, $\mathcal{A}_{\mathsf{Main}}$ outputs m_0 with probability $1/q(\lambda)$. Note that $\mathcal{A}_{\mathsf{Main}}$ outputs m_1 with negligible probability since m_1 is independent of its view. Hence, $\mathcal{A}'$ wins with probability at least $1/q(\lambda) + (1 - 1/q(\lambda)) \cdot 1/2 - \mathsf{negl}(\lambda)$.

Conditioned on the challenge bit value $b = 1$, the adversary $\mathcal{A}'$ again perfectly simulates $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{Guess}(\lambda, \mathcal{A})$. Therefore, by the same argument as above, $\mathcal{A}'$ wins with probability at least $1/q(\lambda) + (1 - 1/q(\lambda)) \cdot 1/2 - \mathsf{negl}(\lambda)$.

Finally, by above, we see that $\mathcal{A}'$ wins $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}$ with probability $1/2 + 1/2 \cdot q(\lambda)$, which is a contradiction. □

Now we show that we can construct a CPA-style leakage-resilient scheme in a black-box way from a scheme that is only leakage-resilient in the random challenge message setting.

Theorem 15. *Suppose there exists a public-key encryption scheme encrypting plaintexts of size $m(\lambda) > \lambda$ into ciphertexts of size $n(\lambda)$, that satisfies $g(\lambda)$-round LOCC leakage-resilience against random challenge messages. Then, for any polynomial $p(\cdot)$, then there exists a public-key encryption scheme, encrypting messages of size $p(\lambda)$ into ciphertexts of size $p(\lambda) \cdot (m(\lambda) + n(\lambda) + 1)$, that satisfies CPA-style $g(\lambda)$-round LOCC leakage-resilience. Further, it uses the previous scheme in a black-box way and it has the same key size.*

Proof. Suppose PKE' is a public-key encryption scheme that satisfies $g(\lambda)$-round LOCC leakage-resilience against random challenge messages, as in the theorem statement. We construct PKE as follows.

$\underline{\mathsf{PKE.Setup}(1^\lambda)}$
Same as $\mathsf{PKE}'.\mathsf{Setup}$.

$\underline{\mathsf{PKE.QKeyGen}(sk)}$
Same as $\mathsf{PKE}'.\mathsf{QKeyGen}$.

$\underline{\mathsf{PKE.Enc}(pk, m)}$

1. For each $i \in [p(\lambda)]$, sample $k_i, r_i \leftarrow \{0,1\}^{m(\lambda)}$.
2. Output $(\mathsf{PKE}'.\mathsf{Enc}(pk, k_i), r_i, \langle k_i, r_i \rangle \oplus (m)_i)_{i \in [p(\lambda)]}$.

$\underline{\mathsf{PKE.Dec}(R_{\mathsf{dec}}, ct)}$

1. Parse $(ct_i, r_i, b_i)_{i \in [p(\lambda)]} = ct$.
2. For each $i \in [p(\lambda)]$, compute $x_i \leftarrow b_i \oplus \langle \mathsf{PKE}'.\mathsf{Dec}(R_{\mathsf{dec}}, ct_i), r_i \rangle$.
3. Output $(x_i)_{i \in [p(\lambda)]}$.

It is easy to see that PKE satisfies correctness. Only subtlety is the fact that to decrypt a ciphertext, we are using $\mathsf{PKE}'.\mathsf{Dec}$ multiple times. However, as discussed before, by correctness of $\mathsf{PKE}'.\mathsf{Dec}$ and by rewinding, we can correctly decrypt polynomially many messages.

We claim that PKE satisfies CPA-style $g(\lambda)$-round LOCC leakage-resilience. For each $j \in \{0, 1, \ldots, p(\lambda)\}$, define the hybrid game Hyb_j as follows by modifying $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}$. Change the challenge ciphertext from

$$(\mathsf{PKE}'.\mathsf{Enc}(pk, k_i), r_i, \langle k_i, r_i \rangle \oplus (m_b)_i)_{i \in [p(\lambda)]}$$

to

$$(\mathsf{PKE}'.\mathsf{Enc}(pk, k_i), r_i, U_1^{(i)} \oplus (m_b)_i)_{i \in [j]}, (\mathsf{PKE}'.\mathsf{Enc}(pk, k_i), r_i, \langle k_i, r_i \rangle \oplus (m_b)_i)_{i \in \{j+1, \ldots, p(\lambda)\}}.$$

where each $U_1^{(i)}$ for $i \in [p(\lambda)]$ are independent uniformly random samples from $\{0,1\}$. Observe that Hyb_0 is the original leakage-resilience game $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}$. Further, it is easy to see that $\Pr\left[\mathsf{Hyb}_{p(\lambda)} = 1\right] \leq 1/2$ since each bit of the message is encrypted with the one-time pad key $U_1^{(i)}$, an independent random bit. Now, we will show that $\mathsf{Hyb}_j \approx \mathsf{Hyb}_{j+1}$ for each $j \in \{0, 1, \ldots, p(\lambda)\}$, and then an applying the hybrid lemma will complete the proof.

Suppose for a contradiction that there is an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$, an index j and a polynomial $w(\lambda)$ such that $|\mathsf{Hyb}_j - \mathsf{Hyb}_{j+1}| \geq \frac{1}{w(\lambda)}$ for infinitely many values of λ. Let ρ denote the final state of $\mathcal{A}_{\mathsf{Main}}$, before it receives the challenge ciphertext. Note that this state has the same distribution in both hybrids since they only differ in their computation of the challenge ciphertext.

Then, it is easy to see that we can distinguish $(\mathsf{PKE}'.\mathsf{Enc}(pk, k_{j+1}), r_{j+1}, U_1 \oplus (m_b)_{j+1})$ versus $(\mathsf{PKE}'.\mathsf{Enc}(pk, k_{j+1}), r_{j+1}, \langle r_{j+1} k_{j+1} \rangle \oplus (m_b)_{j+1})$ with advantage $1/w(\lambda)$ using ρ. In turn, by a standard argument, this implies that we can predict $\langle r_{j+1} k_{j+1} \rangle$ with probability at least $1/2 + 1/(2 \cdot w(\lambda))$ using ρ, $\mathsf{PKE}'.\mathsf{Enc}(pk, k_{j+1})$ and r_{j+1}. Finally, by Theorem 7, this means that there exists an efficient algorithm that predicts k_{j+1} with probability $1/w^2(\lambda)$ on input ρ and $\mathsf{PKE}'.\mathsf{Enc}(pk, k_{j+1})$. However, this is a contradiction to random challenge message $g(\lambda)$-round LOCC leakage-resilience of PKE. □

5.2 Relationship Between Anti-Piracy Security and Leakage-Resilience

In this section, we show various results regarding the relationship between anti-piracy security and leakage-resilience.

Theorem 16. *Let* PKE *be a public-key encryption scheme with key protection that satisfies anti-piracy security with random challenge messages. Then, it also satisfies* 1*-round random challenge message LOCC leakage-resilience.*

The simple proof relies on the fact that if there is an adversary that can win the leakage-resilience game, then in the anti-piracy game we can *leak* on the key first and then *clone* the leakage which is classical.

Proof. Suppose for a contradiction that there exists a 1-round QPT LOCC adversary pair $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the random challenge message LOCC leakage-resilience game with non-negligible probability. We construct an adversary $\mathcal{A}'$ for $\mathsf{PKE} - \mathsf{AntiPiracy} - \mathsf{Guess}$ as follows.

$\underline{\mathcal{A}'(R_{\mathsf{key}}, pk)}$
Simulate $\mathcal{A}_{\mathsf{Leak}}$ on R_{key}, pk to obtain a classical leakage message ℓ. Output $((\mathcal{A}'', \ell), (\mathcal{A}'', \ell))$.

$\mathcal{A}''(R, ct)$

Same as $\mathcal{A}_{\mathsf{Main}}$.

Now, observe that conditioned on some fixed values of the leakage and the public key, the success of the two freeloaders are independent. Hence, can write the probability of $\mathcal{A}'$ winning as

$$\begin{aligned}&\Pr\left[\mathsf{PKE}-\mathsf{AntiPiracy}-\mathsf{Guess}(\lambda,\mathcal{A}')=1\right]\\&=\mathbb{E}_{pk,\ell}[\left(\Pr\left[\mathcal{A}''(\ell',ct)=m^*|pk'=pk,\ell'=\ell\right]\right)^2]=\mathbb{E}_{pk,\ell}[(\Pr[\mathcal{A}_{\mathsf{Main}}(\ell,ct)=m^*])^2]\end{aligned}$$

where $ct \leftarrow \mathsf{PKE.Enc}(pk', m^*)$ and $m^* \leftarrow \mathcal{M}$. Note that we also have

$$\frac{1}{p(\lambda)} < \Pr[\mathsf{PKE}-\mathsf{LOCC}-\mathsf{Guess}(\lambda,\mathcal{A})=1]=\mathbb{E}_{pk,\ell}[(\Pr[\mathcal{A}_{\mathsf{Main}}(\ell,ct)=m^*])]$$

for infinitely many values of $\lambda > 0$ by assumption. Then, by Jensen's inequality,

$$\mathbb{E}_{pk,\ell}\left[\Pr[\mathcal{A}_{\mathsf{Main}}(\ell,ct)=m^*]^2\right] > \frac{1}{p^2(\lambda)}$$

for infinitely many values of $\lambda > 0$, which is a contradiction to anti-piracy security of PKE.

Our proof generalizes to the non-uniform quantum advice adversaries in a straightforwards manner: If $\mathcal{A}_{\mathsf{Main}}$ has advice $\{\sigma_\lambda\}_\lambda$ and $\mathcal{A}_{\mathsf{Leak}}$ has advice $\{\tau_\lambda\}_\lambda$, we define $\mathcal{A}'$ so that it has advice $\{\sigma_\lambda^{\otimes 2} \otimes \tau_\lambda\}_\lambda$. It uses τ_λ to simulate $\mathcal{A}_{\mathsf{Leak}}$, and it outputs $((\mathcal{A}'', \ell, \sigma_\lambda), (\mathcal{A}'', \ell, \sigma_\lambda))$ at the end. $\mathcal{A}''$ uses σ_λ that it receives in its input to simulate $\mathcal{A}_{\mathsf{Main}}$. □

Combining with our previous results, we obtain the following corollary.

Corollary 1. *Suppose there exists a public-key encryption scheme with key protection that satisfies anti-piracy security against random challenge messages. Then, there exists a public-key encryption scheme with key protection that uses the former in a black-box way and satisfies CPA-style* 1*-round LOCC leakage-resilience.*

Theorem 17. *Let* PKE *be a public-key encryption scheme with key protection that satisfies anti-piracy security with random challenge messages against adversaries with non-uniform quantum advice. Then, it also satisfies* 2*-round random challenge message LOCC leakage-resilience against adversaries with non-uniform quantum advice.*

Proof is similar to above, but it utilizes the non-uniform quantum advice to obtain two copies of the state of $\mathcal{A}_{\mathsf{Main}}$.

Proof. Suppose for a contradiction that there exists a 2-round QPT LOCC adversary pair $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the random challenge message LOCC leakage-resilience game with non-negligible probability. Suppose $\mathcal{A}_{\mathsf{Main}}$ has advice $\{\sigma_\lambda\}_\lambda$ and $\mathcal{A}_{\mathsf{Leak}}$ has advice $\{\tau_\lambda\}_\lambda$. Since $\mathcal{A}_{\mathsf{Main}}$ outputs a classical message to $\mathcal{A}_{\mathsf{Leak}}$ and a state at the beginning, we can write $\sum_{x\in\{0,1\}^{k(\lambda)}} q_{x,\lambda}|x\rangle\langle x| \otimes \xi_{x,\lambda} = \mathcal{A}_{\mathsf{Main}}(1^\lambda, \sigma_\lambda)$ where $k(\lambda)$ is the length of the first message. Define

$$\xi'_\lambda = \sum_{x\in\{0,1\}^{k(\lambda)}} q_{x,\lambda}|x\rangle\langle x| \otimes \xi_{x,\lambda} \otimes \xi_{x,\lambda}.$$

We construct an adversary $\mathcal{A}'$ for $\mathsf{PKE}-\mathsf{AntiPiracy}-\mathsf{Guess}$ as follows. We define the non-uniform quantum advice of $\mathcal{A}'$ to be $\{\tau_\lambda \otimes \xi'_\lambda\}$.

$\underline{\mathcal{A}'(R_{\mathsf{key}}, pk, \xi'_\lambda)}$
Sample from ξ'_λ to obtain x and $\xi_{x,\lambda} \otimes \xi_{x,\lambda}$ for some $x \in \{0,1\}^{k(\lambda)}$. Simulate $\mathcal{A}_{\mathsf{Leak}}$ on $R_{\mathsf{key}}, pk, x, \tau_\lambda$ to obtain a classical leakage message ℓ. Output $((\mathcal{A}'', \ell, \xi_{x,\lambda}), (\mathcal{A}'', \ell, \xi_{x,\lambda}))$.

$\underline{\mathcal{A}''(R, ct)}$
Same as $\mathcal{A}_{\mathsf{Main}}$.

Similar to the proof of the previous theorem, observe that conditioned on some fixed values of the leakage, the public key and the first message x, the success of the two freeloaders are independent. Hence, can write the probability of $\mathcal{A}'$ winning as

$$\begin{aligned}
&\Pr\big[\mathsf{PKE} - \mathsf{AntiPiracy} - \mathsf{Guess}(\lambda, \mathcal{A}') = 1\big] \\
&= \mathbb{E}_{pk,x,\ell}[\big(\Pr\big[\mathcal{A}''(\ell', ct, \xi_{x',\lambda}) = m^* | pk' = pk, x' = x, \ell' = \ell\big]\big)^2] \\
&= \mathbb{E}_{pk,x,\ell}[(\Pr[\mathcal{A}_{\mathsf{Main}}(\ell, ct, \xi_{x,\lambda}) = m^*])^2],
\end{aligned}$$

where $ct \leftarrow \mathsf{PKE.Enc}(pk', m^*)$, $m^* \leftarrow \mathcal{M}$ and x is sampled with probability $q_{x,\lambda}$ independently. Note that we also have

$$\frac{1}{p(\lambda)} < \Pr[\mathsf{PKE} - \mathsf{LOCC} - \mathsf{Guess}(\lambda, \mathcal{A}) = 1] = \mathbb{E}_{pk,x,\ell}[(\Pr[\mathcal{A}_{\mathsf{Main}}(\ell, ct, \xi_{x,\lambda}) = m^*])]$$

for infinitely many values of $\lambda > 0$ by assumption. Then, by Jensen's inequality,

$$\mathbb{E}_{pk,x,\ell}\Big[\Pr[\mathcal{A}_{\mathsf{Main}}(\ell, ct, \xi_{x,\lambda}) = m^*]^2\Big] > \frac{1}{p^2(\lambda)}$$

for infinitely many values of $\lambda > 0$, which is a contradiction to anti-piracy security of PKE. □

Again, combining with our extractor based transformation from the previous section, we obtain the following result.

Corollary 2. *Suppose there exists a public-key encryption scheme with key protection that satisfies random challenge message anti-piracy security against adversaries with non-uniform quantum advice. Then, there exists a public-key encryption scheme with key protection that uses the former in a black-box way and satisfies CPA-style 2-round LOCC leakage-resilience against adversaries with non-uniform quantum advice.*

While we are not able to directly show that CPA-style anti-piracy security implies CPA-style LOCC leakage-resilience, we have the following win-win result that show that either such an implication does exist, or we can construct (weak) quantum lightning. Later, we also show that based on existence of quantum lightning (and an additional *strong* obfuscation assumption), we can show that anti-piracy security does not imply stronger LOCC leakage-resilience notions. Therefore, it is likely that the current state of affairs we establish is not a limitation of our proof techniques but rather it is the due to an inherent separation between anti-piracy and LOCC leakage-resilience.

Theorem 18. *Let* PKE *be a public-key encryption scheme with key protection that satisfies CPA-style anti-piracy security. Then both of the following are true.*

- *Either* PKE *is infinitely often CPA-style* 1*-round LOCC leakage-resilient, or* PKE *can be used to build weak quantum lightning (in the setup model).*
- *Either* PKE *CPA-style* 1*-round LOCC leakage-resilient, or* PKE *can be used to build infinitely often secure weak quantum lightning (in the setup model).*

Proof. We will assume that PKE is not LOCC leakage-resilient, and we will construct a quantum lightning scheme. If PKE is anti-piracy secure against adversaries with quantum advice, the quantum lightning scheme we construct is also secure against adversaries with quantum advice. The only difference between the two bullet points is the advantage of the LOCC adversary being an inverse polynomial or non-negligible. Therefore, we will focus on the first bullet, and the second bullet follows by the same argument.

Let PKE be a CPA-style anti-piracy secure public-key encryption scheme. Let $\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}}$ be a 1-round QPT LOCC adversary that wins the CPA-style leakage-resilience game with probability $1/2 + 1/p(\lambda)$ for sufficiently large λ. We construct a weak quantum lightning scheme QL as follows.

$\underline{\mathsf{QL.Setup}(1^\lambda)}$

1. $pk, sk \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.
2. $R_{\mathsf{key}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$.
3. $\ell, \sigma \leftarrow \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}}, pk)$.
4. Output pk, ℓ.

$\underline{\mathsf{QL.Bolt}((pk, \ell))}$

1. Sample $m_0, m_1, \rho \leftarrow \mathcal{A}_{\mathsf{Main}}(\ell)$
2. Output $(m_0, m_1), \rho$.

$\underline{\mathsf{QL.Verify}((\ell, pk), (m_0, m_1), \rho)}$

1. $b \leftarrow \{0, 1\}$.
2. $ct \leftarrow \mathsf{PKE.Enc}(pk, m_b)$.
3. $b' \leftarrow \mathcal{A}_{\mathsf{Main}}(\rho, ct)$.
4. Output 1 if $b' = b$. Otherwise, output 0.

First, we argue correctness. Observe that when the verification procedure is run on an honestly generated bolt, it perfectly simulates $\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}}$ playing $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}$. Since we assumed that the adversary wins with probability $1/2 + 1/p(\lambda)$, correctness of QL with same probability follows.

We now argue security. Suppose for a contradiction that there exists an adversary $\mathcal{B}(\ell, pk)$ that produces a pair of registers R_1, R_2 and a serial number (m_0, m_1) such that when we run $\mathsf{QL.Verify}((\ell, pk), (m_0, m_1), R_i)$ for $i \in \{1, 2\}$, with probability $1/2 + 1/q(\lambda)$ both verification passes simultaneously, where $q(\lambda)$ is a polynomial. Then, we construct the following adversary for the anti-piracy game for PKE. Given the keys pk, R_{key}, we first run $\mathcal{A}_{\mathsf{Leak}}$ to obtain ℓ, and then we run $\mathcal{B}$ on these to obtain R_1, R_2 and (m_0, m_1) as above. Finally, we output $((\mathcal{A}_{\mathsf{Main}}(R_1, \cdot)), (\mathcal{A}_{\mathsf{Main}}(R_2, \cdot))$ as the freeloader adversaries and m_0, m_1 as the challenge messages. Then, observe that running $\mathsf{QL.Verify}((\ell, pk), (m_0, m_1), R_i)$ for $i \in \{1, 2\}$ exactly corresponds to the second part of the challenger of the anti-piracy game. Hence, this adversary we constructed wins the anti-piracy game with probability $1/2 + 1/q(\lambda)$, which is a contradiction. This establishes the $1/2 + \mathsf{negl}(\lambda)$ security of our mini-scheme.

□

Quantum Lightning Implies the Other Direction

Above, we have established that anti-piracy security implies 2-round random challenge message LOCC leakage-resilience (in non-uniform quantum advice model), and it also implies 1-round CPA-style LOCC leakage-resilience or quantum lightning exists with no additional assumptions.

Therefore, one might wonder if LOCC leakage-resilience in general is a weaker notion than anti-piracy and if it is directly implied by anti-piracy. In this section, we give a heuristical justification that this is not the case: If quantum lightning exists, then based on additional (strong) obfuscation assumptions, there exists a public-key encryption scheme with key protection that satisfies anti-piracy security but not 2-round LOCC leakage-resilience (in no quantum advice model). Since our construction is based on coset states, we suggest the reader first read Section 5.3.

We first give the following property of coset states.

Theorem 19 (Computational Direct Product Hardness for Coset States [CLLZ21, Theorem 4.6])**.** *Assume the existence of $i\mathcal{O}$ and one-way functions. Sample uniformly at random a subspace A of $\mathbb{F}_2^\lambda$ of dimension $\frac{\lambda}{2}$ and two elements $s, s' \leftarrow \mathbb{F}_2^\lambda$. Sample $\mathsf{OP}^0 \leftarrow i\mathcal{O}(A+s)$. and $\mathsf{OP}^1 \leftarrow i\mathcal{O}(A^\perp + s')$.*

Then, any QPT adversary, on input $\left|A_{s,s'}\right\rangle, \mathsf{OP}^0, \mathsf{OP}^1$, outputs (v, w) such that $v \in A+s$ and $w \in A^\perp + s'$ with negligible probability.

Our theorem below is in the *no* quantum advice model for adversaries.

Theorem 20. *Suppose $q\mathcal{O}$ is a virtual black-box obfuscation scheme for quantum circuits and QL is a quantum lightning scheme. Then, there exists a public-key encryption scheme that satisfies anti-piracy security but does not satisfy 2-round LOCC leakage-resilience.*

We construct such a scheme as follows. Let $d(\lambda)$ denote the length of the serial numbers of QL. Proof utilizes the fact that in the anti-piracy game, we need to come up with two decrypting adversaries as opposed to a single adversary in the LOCC leakage-resilience game. In our construction, we require that, to be able to decrypt a ciphertext, the adversary needs to have a valid quantum lightning bolt and it needs to output some vectors in either $A_i + s_i$ or $A_i^\perp + s_i'$ depending on the serial number of the bolt. Since the 2-round LOCC adversary can sample a bolt, send the serial number to the leakage adversary and measure the coset state accordingly, the construction is not LOCC leakage-resilient. However, the anti-piracy adversary will need to produce two valid bolts, which will have different serial numbers by the security of quantum lightning. Then, by direct product hardness, it will not be able to produce two freeloaders that can both decrypt.

We now give the full construction.

$\underline{\mathsf{PKE.Setup}(1^\lambda)}$

1. For $i \in [d(\lambda)]$,
   1.1. Sample A_i a subspace of dimension $\lambda/2$ of $\mathbb{F}_2^\lambda$.
   1.2. Sample two vectors $s_i, s_i' \leftarrow \mathbb{F}_2^\lambda$.
   1.3. Sample $\mathsf{OP}_i^0 \leftarrow q\mathcal{O}(A_i + s_i)$.
   1.4. Sample $\mathsf{OP}_i^1 \leftarrow q\mathcal{O}(A_i^\perp + s_i')$.
2. Set $sk = (A_i, s_i, s_i')_{i\in[d(\lambda)]}$.
3. Set $pk = (\mathsf{OP}_i^0, \mathsf{OP}_i^1)_{i\in[d(\lambda)]}$.
4. Output (pk, sk).

$\underline{\mathsf{PKE.QKeyGen}(sk)}$

1. Parse $(A_i, s_i, s'_i)_{i\in[d(\lambda)]} = sk$.
2. Output $\left(\left|A_{i,s_i,s'_i}\right\rangle\right)_{i\in[d(\lambda)]}$.

$\underline{\mathsf{PKE.Enc}(pk, m)}$

1. Parse $(\mathsf{OP}^0_i, \mathsf{OP}^1_i)_{i\in[d(\lambda)]} = pk$.
2. Sample $ct \leftarrow q\mathcal{O}(\mathsf{PCt})$.

   $\underline{\mathsf{PCt}(R_{\mathsf{bolt}}, x, u_1, \ldots, u_{d(\lambda)})}$

   Hardcoded: $(\mathsf{OP}^0_i, \mathsf{OP}^1_i)_{i\in[d(\lambda)]}, m$

   1. Check if $\mathsf{QL.Ver}(R_{\mathsf{bolt}}, x) = 1$. If check fails, output $\bot$ and terminate.
   2. For $i \in [d(\lambda)]$, check if $\mathsf{OP}^0_i(u_i) = 1$ if $(x)_i = 0$ and if $\mathsf{OP}^1_i(u_i) = 1$ if $(x)_i = 1$.
   3. Output m if all the checks pass. Otherwise, output $\bot$.

3. Output OPCt.

$\underline{\mathsf{PKE.Dec}(R_{\mathsf{key}}, ct)}$

1. Parse $((R_i)_{i\in[d(\lambda)]}) = R_{\mathsf{key}}$ and $\mathsf{OPCt} = ct$.
2. Sample $x, R_{\mathsf{bolt}} \leftarrow \mathsf{QL.Bolt}(1^\lambda)$.
3. For indices $i \in [d(\lambda)]$ such that $(x)_i = 1$, apply $H^{\otimes\lambda}$ to R_i.
4. Measure $((R_i)_{i\in[d(\lambda)]})$ in the computational basis to obtain vectors $(v_i)_{i\in[d(\lambda)]}$.
5. Run the program OPCt on $R_{\mathsf{bolt}}, v_1, \ldots, v_{d(\lambda)}$.
6. Output the result of the program.

Proof. Correctness follows in a straightforward manner from the correctness of the underlying primitives.[10]

We first show that this scheme is not 2-round LOCC leakage-resilient. $\mathcal{A}_{\mathsf{Main}}$ samples a bolt $x, R_{\mathsf{bolt}} \leftarrow \mathsf{QL.Bolt}(1^\lambda)$. It keeps R_{bolt}, x as its state and also sends x to $\mathcal{A}_{\mathsf{Leak}}$. Then, $\mathcal{A}_{\mathsf{Leak}}$, which has $((R_i)_{i\in[d(\lambda)]}) = R_{\mathsf{key}}$, applies $H^{\otimes\lambda}$ to R_i for indices $i \in [d(\lambda)]$ such that $(x)_i = 1$. Finally, it measures the resulting registers in the computational basis to obtain vectors $(v_i)_{i\in[d(\lambda)]}$. It submits the vectors to $\mathcal{A}_{\mathsf{Main}}$. When $\mathcal{A}_{\mathsf{Main}}$ is presented with a challenge ciphertext, it runs it on R_{bolt}, x and $(v_i)_{i\in[d(\lambda)]}$. It is easy to see that this is a perfect simulation of an honest decryption, therefore, $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ wins the 2-round leakage-resilience game with probability $1 - \mathsf{negl}(\lambda)$.

Now we show that PKE satisfies anti-piracy security. Suppose for a contradiction that there exists an adversary that wins the anti-piracy security game with non-negligible probability. We first replace the freeloader adversaries with oracle-aided simulated ones that we obtain using the security of $q\mathsf{O}$. Observe that for both to successfully decrypt, the simulated adversaries need to

[10] We only have overwhelming correctness here since $\mathsf{QL.Ver}$ is only guaranteed to have overwhelming correctness.

query their PCt oracles with inputs $R_{\mathsf{bolt},1}, x, v_1, \ldots, v_{d(\lambda)}$ and $R_{\mathsf{bolt},2}, y, w_1, \ldots, w_{d(\lambda)}$ such that $x = \mathsf{QL.Ver}(R_{\mathsf{bolt},1}, x) = 1, y = \mathsf{QL.Ver}(R_{\mathsf{bolt},2}, y) = 1$ and v_i, w_i for $i \in [d(\lambda)]$ that are in correct cosets with respect to $(x)_i, (y)_i$, respectively[11]. Now, observe that, by security of QL, we will have $x \neq y$ except with negligible probability. Hence there exists an index $i^* \in [d(\lambda)]$ such that $(x)_{i^*} \neq (y)_{i^*}$. Hence, once we have $R_{\mathsf{bolt},1}, v_1, \ldots, v_{d(\lambda)}$ and $R_{\mathsf{bolt},2}, w_1, \ldots, w_{d(\lambda)}$, we can run $\mathsf{QL.Ver}$ on both bolts to find i^*, after which we obtain either $v_{i^*} \in A_{i^*} + s_{i^*}, w_{i^*} \in A_{i^*}^{\perp} + s'_{i^*}$ or $w_{i^*} \in A_{i^*} + s_{i^*}, v_{i^*} \in A_{i^*}^{\perp} + s'_{i^*}$. Overall, we obtain such vectors with probability $1/q(\lambda)$ for some polynomial $q(\cdot)$. Finally, we can break the direct product hardness (Theorem 19) as follows. We place the coset state (and the obfuscated programs) obtained from the challenger at a random index $j \leftarrow [d(\lambda)]$ and run the reduction above by filling in the other indices with coset states we sample ourselves. Since for any value of j the input distribution to the reduction above is the same, we obtain vectors that break the direct product hardness for the coset state obtained from the challenger with probability $1/(q(\lambda) \cdot d(\lambda))$, which is a contradiction by Theorem 19. Therefore, PKE satisfies anti-piracy security. □

It is easy to see that the result above can be generalized to the case of adversaries with non-uniform quantum advice in a straightforward manner. In this case, we assume that the quantum lightning scheme has a setup phase and a public key. The LOCC leakage attack proceeds the same as before, with the only difference being that we now need 3 rounds and in the first round, $\mathcal{A}_{\mathsf{Leak}}$ sends the public key of the lightning scheme to $\mathcal{A}_{\mathsf{Main}}$.

Theorem 21. *Suppose* QL *is a quantum lightning scheme and* $q\mathcal{O}$ *is a virtual black-box obfuscation scheme for quantum circuits. Then, there exists a public-key encryption scheme that satisfies anti-piracy security against adversaries with non-uniform quantum advice but does not satisfy* 3*-round LOCC leakage-resilience.*

5.3 Coset State-Based Construction

In this section, we show that the anti-piracy secure public-key encryption scheme of [CLLZ21] based on coset states also satisfies CPA-style LOCC leakage-resilience. For completeness, we first recall the construction of [CLLZ21], slightly modified to match our notation and the parameters we require for LOCC leakage-resilience.

Assume the existence of following schemes.

- $i\mathcal{O}$, subexponentially secure indistinguishability obfuscation scheme,
- CCObf, compute-and-compare obfuscation for $2^{-\lambda^{0.5 \cdot C_{\mathsf{CompLOCC}}}}$-unpredictable distributions.
- Subexponentially-secure one-way functions.

$\underline{\mathsf{PKE.Setup}(1^\lambda)}$

1. Sample $(A_i, s_i, s'_i)_{i \in [c(\lambda)]} \leftarrow \mathsf{CosetGen}(1^\lambda)$.
2. Set $sk = (A_i, s_i, s'_i)_{i \in [c(\lambda)]}$.

[11] While the simulated adversaries will have many queries to their oracles, there will be polynomially many, and we can make a random guess and we will correctly predict which query is of the above form with only a polynomial loss. After out prediction, we do keep the register that is in the query but we do not run $\mathsf{QL.Ver}$ on it, since later we will output it and the quantum lightning will run $\mathsf{QL.Ver}$ on it. We simply output the hidden message m for this query. Observe that conditioned on having hit the correct index for this special query, we do not affect the rest of the adversary.

3. For $i \in [c(\lambda)]$,
   3.1. Sample $\mathsf{OP}_i^0 \leftarrow i\mathcal{O}(A_i + s_i)$.
   3.2. Sample $\mathsf{OP}_i^1 \leftarrow i\mathcal{O}(A_i^{\perp} + s_i')$.
4. Set $pk = (\mathsf{OP}_i^0, \mathsf{OP}_i^1)_{i\in[c(\lambda)]}$.
5. Output (pk, sk).

$\underline{\mathsf{PKE.QKeyGen}(sk)}$

1. Parse $(A_i, s_i, s_i')_{i\in[c(\lambda)]} = sk$.
2. Output $\left(\left|A_{i,s_i,s_i'}\right\rangle\right)_{i\in[c(\lambda)]}$.

$\underline{\mathsf{PKE.Enc}(pk, m)}$

1. Parse $(\mathsf{OP}_i^0, \mathsf{OP}_i^1)_{i\in[c(\lambda)]} = pk$.
2. Sample $r \leftarrow \{0,1\}^{c(\lambda)}$.
3. Sample $\mathsf{OPCt} \leftarrow i\mathcal{O}(\mathsf{PCt})$, where PCt is the following program.

   $\underline{\mathsf{PCt}(u_1, \ldots, u_{c(\lambda)})}$
   Hardcoded: $(\mathsf{OP}_i^0, \mathsf{OP}_i^1)_{i\in[c(\lambda)]}, m$
   1. For $i \in [c(\lambda)]$, check if $\mathsf{OP}_i^0(u_i) = 1$ if $(r)_i = 0$ and if $\mathsf{OP}_i^1(u_i) = 1$ if $(r)_i = 1$.
   2. Output m if all the checks pass. Otherwise, output $\perp$.

4. Output (OPCt, r).

$\underline{\mathsf{PKE.Dec}(R_{\mathsf{key}}, ct)}$

1. Parse $((R_i)_{i\in[c(\lambda)]}) = R_{\mathsf{key}}$ and $(\mathsf{OPCt}, r) = ct$.
2. For indices $i \in [c(\lambda)]$ such that $(r)_i = 1$, apply $H^{\otimes\lambda}$ to R_i.
3. Run the program OPCt coherently on $(R_i)_{i\in[c(\lambda)]}$.
4. Measure the output register and output the outcome.

Theorem 22 ([CLLZ21])**.** PKE *satisfies correctness and both CPA-style and random challenge message anti-piracy security.*

We claim that the construction is also LOCC-leakage-resilient.

Theorem 23. PKE *satisfies both CPA-style LOCC leakage-resilience.*

When we instantiate the assumed building blocks with known constructions, we get the following corollary.

Corollary 3. *Assuming subexponentially secure $i\mathcal{O}$, one-way functions and polynomially hard qLWE, there exists a public-key encryption scheme that satisfies CPA-style LOCC leakage-resilience.*

Proof of Security

In this section, we prove Theorem 23, that is, we show that the construction PKE satisfies LOCC leakage-resilience. Our proof will be similar to the anti-piracy proof in [CLLZ21] in the sense that we also utilize compute-and-compare obfuscation. However, our proof for LOCC leakage-resilience is much more straightforward since we will not need to simultaneously extract vectors from entangled adversaries.

We prove security through a series of hybrids.

$\underline{\mathsf{Hyb}_0}$**:** The original game $\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}(\lambda, \mathcal{A})$.

$\underline{\mathsf{Hyb}_1}$**:** We change the way we compute the challenge ciphertext.

1. Sample $r \leftarrow \{0,1\}^{c(\lambda)}$.
2. Parse $(A_i, s_i, s_i') = sk$.
3. Sample $r \leftarrow \{0,1\}^{c(\lambda)}$.
4. For $i \in [c(\lambda)]$, set $g_i = \mathsf{Can}_{A_i}$ if $(r)_i = 0$ and set $g_i = \mathsf{Can}_{(A_i)^\perp}$ if $(r)_i = 1$.
5. For $i \in [c(\lambda)]$, compute $y_i = g_i(s_i)$ if $(r)_i = 0$ and $y_i = g_i(s_i')$ if $(r)_i = 1$.
6. Set g to be the function $g(v_1, \ldots, v_{c(\lambda)}) = (g_1(v_1)||\ldots||g_{c(\lambda)}(v_{c(\lambda)}))$.
7. Set $y = y_1||\ldots||y_{c(\lambda)}$.
8. Compute $\mathsf{OCC} \leftarrow \mathsf{CCObf.Obf}(g, y, m_b)$.
9. $\mathsf{OPCt} \leftarrow i\mathcal{O}(\mathsf{PCt}')$.

   $\underline{\mathsf{PCt}'(u_1, \ldots, u_{c(\lambda)})}$

   Hardcoded: OCC

   1. Output $\mathsf{OCC}(u_1, \ldots, u_{c(\lambda)})$.

10. Output (OPCt, r).

$\underline{\mathsf{Hyb}_2}$**:** We again change the computation of the challenge ciphertext.

1. Sample $r \leftarrow \{0,1\}^{c(\lambda)}$.
2. Parse $(A_i, s_i, s_i') = sk$.
3. Sample $r \leftarrow \{0,1\}^{c(\lambda)}$.
4. For $i \in [c(\lambda)]$, set $g_i = \mathsf{Can}_{A_i}$ if $(r)_i = 0$ and set $g_i = \mathsf{Can}_{(A_i)^\perp}$ if $(r)_i = 1$.
5. For $i \in [c(\lambda)]$, compute $y_i = g_i(s_i)$ if $(r)_i = 0$ and $y_i = g_i(s_i')$ if $(r)_i = 1$.
6. Set g to be the function $g(v_1, \ldots, v_{c(\lambda)}) = (g_1(v_1)||\ldots||g_{c(\lambda)}(v_{c(\lambda)}))$.
7. Set $y = y_1||\ldots||y_{c(\lambda)}$.

8. $\mathsf{PSim} \leftarrow \mathsf{CCObf.Sim}(1^\lambda, |g|, |y|, |m|)$

9. $\mathsf{OPCt} \leftarrow i\mathcal{O}(\mathsf{PCt}'')$.

   $\underline{\mathsf{PCt}''(u_1, \ldots, u_{c(\lambda)})}$

   Hardcoded: PSim

   1. Output $\mathsf{PSim}(u_1, \ldots, u_{c(\lambda)})$.

10. Output (OPCt, r).

Claim 7. $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$.

Proof. Note that $v \in A_i + s_i$ if and only if $\mathsf{Can}_{A_i}(v) = \mathsf{Can}_{A_i}(s_i)$. Similarly for $A_i^\perp + s_i'$. Then, by correctness of the inner obfuscations (i.e., $\mathsf{OP}_i^0, \mathsf{OP}_i^1$) in PCt and the correctness of CCObf, we have that PCt and PCt' have the same functionality. The result follows from the security of the outer obfuscation. □

Claim 8. $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$.

Proof. Suppose for a contradiction that there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ such that $|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]|$ is non-negligible. Then, we can construct a distribution $\mathcal{D}$ over compute-and-compare functions and an adversary $\mathcal{A}'$ for CCObf as follows.

$\underline{\mathcal{D}(1^\lambda)}$

1. Sample $pk, sk \leftarrow \mathsf{PKE.Setup}(1^\lambda)$.
2. Sample $R_{\mathsf{key}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$ and submit pk, R_{key} to $\mathcal{A}_{\mathsf{Leak}}$.
3. $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ execute their LOCC protocol.
4. After the LOCC protocol is over, $\mathcal{A}_{\mathsf{Main}}$ outputs two messages m_0, m_1. Let ρ be the final state of $\mathcal{A}_{\mathsf{Main}}$.
5. Sample $b \leftarrow \{0,1\}$.
6. Sample $r \leftarrow \{0,1\}^{c(\lambda)}$.
7. Parse $(A_i, s_i, s_i') = sk$.
8. For $i \in [c(\lambda)]$, set $g_i = \mathsf{Can}_{A_i}$ if $(r)_i = 0$ and set $g_i = \mathsf{Can}_{(A_i)^\perp}$ if $(r)_i = 1$.
9. For $i \in [c(\lambda)]$, compute $y_i = g_i(s_i)$ if $(r)_i = 0$ and $y_i = g_i(s_i')$ if $(r)_i = 1$.
10. Set g to be the function $g(v_1, \ldots, v_{c(\lambda)}) = (g_1(v_1)||\ldots||g_{c(\lambda)}(v_{c(\lambda)}))$.
11. Set $y = y_1||\ldots||y_{c(\lambda)}$.
12. Output $(g, y, m_b), (\rho, b, r)$.

$\underline{\mathcal{A}'(P, R_{\mathsf{aux}})}$

1. Parse $(\rho, b, r) = R_{\mathsf{aux}}$.
2. Sampe $\mathsf{OPCt} \leftarrow i\mathcal{O}(\mathsf{PCt}'')$.

   $\underline{\mathsf{PCt}''(u_1, \ldots, u_{c(\lambda)})}$

   Hardcoded: P

   1. Output $P(u_1, \ldots, u_{c(\lambda)})$.

3. Run $\mathcal{A}_{\mathsf{Main}}$ on ρ and (OPCt, r). Let b' be the output.
4. Output 1 if $b' = b$. Otherwise, output 0.

It is easy to see that $\mathcal{A}'(\mathsf{OCC}, R_{\mathsf{aux}})$ corresponds to Hyb_1 and $\mathcal{A}'(\mathsf{PSim}, R_{\mathsf{aux}})$ corresponds to Hyb_2. Since we assumed $\mathsf{Hyb}_1 \not\approx \mathsf{Hyb}_2$, by Definition 10 we get that there exists an adversary $\mathcal{A}''$ such that given R_{aux} and g, it outputs y with probability at least $2^{-\lambda^{0.5 \cdot C_{\mathsf{CompLOCC}}}}$.

Then, we construct an adversary $\mathcal{A}''' = (\mathcal{A}'''_{\mathsf{Main}}, \mathcal{A}'''_{\mathsf{Leak}})$ for $\mathsf{Coset} - \mathsf{CompLOCC}$ as follows.

$$\underline{\mathcal{A}'''_{\mathsf{Leak}}\left(\left\{\left|A_{i,s_i,s'_i}\right\rangle\right\}_{i\in[c(\lambda)]}, (\mathsf{OP}^0_i, \mathsf{OP}^1_i)_{i\in[c(\lambda)]}\right)}$$

Set $R_{\mathsf{key}} = \left|A_{i,s_i,s'_i}\right\rangle_{i\in[c(\lambda)]}$ and $pk = (\mathsf{OP}^0_i, \mathsf{OP}^1_i)_{i\in[c(\lambda)]}$. Simulate $\mathcal{A}_{\mathsf{Leak}}$ on R_{key}, pk.

$\underline{\mathcal{A}_{\mathsf{Main}}'''}$

Simulate $\mathcal{A}_{\mathsf{Main}}$ until the end of the LOCC protocol to obtain the final state ρ and the challenge messages m_0, m_1. Sample $b \leftarrow \{0,1\}$. Then, compute the description of g as done by $\mathcal{D}$ using the subspace descriptions (A_i, s_i, s'_i) and r obtained from the challenger. Finally, run $\mathcal{A}''$ on (ρ, b, r) and g, and output the vectors output by it.

By above, we can see that $\mathcal{A}'''$ outputs vectors in the correct cosets with probability $2^{-\lambda^{0.5 \cdot C_{\mathsf{CompLOCC}}}}$ in the game $\mathsf{Coset} - \mathsf{CompLOCC}$, which is a contradiction by Theorem 11. □

Observe that in Hyb_2, the challenge ciphertext is independent of b. Hence, $\Pr[\mathsf{Hyb}_2 = 1] \leq \frac{1}{2}$ and therefore $\Pr[\mathsf{PKE} - \mathsf{LOCC} - \mathsf{CPA}(\lambda, \mathcal{A})] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$ by above.

6 Digital Signatures Schemes with Key Protection

In this section, we introduce the concept of digital signatures with key protection and formally define LOCC leakage-resilience for digital signature schemes with key protection. Then, we show that the construction of [LLQZ22] of a digital signature scheme that satisfies anti-piracy security also satisfies LOCC leakage-resilience.

Definition 20 (Digital signature scheme with key protection)**.** *A digital signature scheme* DS *with key protection consists of the following algorithms.*

- $\mathsf{Setup}(1^\lambda)$: *Outputs a classical signing key* sk *and a public classical verification key* vk*.*
- $\mathsf{QKeyGen}(sk)$: *Takes the signing key* sk*, outputs a quantum key register* R_{sign}*.*

- $\mathsf{Sign}(R_{\mathsf{sign}}, m)$: *Takes the quantum signing key and a message m, returns a signature for m.*
- $\mathsf{Ver}(vk, m, sg)$: *Takes the public verification key, a message m and a (supposed) signature sg for m, returns 1 if sg is a valid signature for m.*

We require that the scheme satisfies correctness and psuedodeterministic signatures properties. Let $\mathcal{M}$ denote the message space.

Correctness *For all messages $m \in \mathcal{M}$,*

$$\Pr\left[\mathsf{Ver}(vk, m, sg) = 1 : \begin{array}{c} sk, vk \leftarrow \mathsf{Setup}(1^\lambda) \\ R_{\mathsf{sign}} \leftarrow \mathsf{QKeyGen}(sk) \\ sg \leftarrow \mathsf{Sign}(R_{\mathsf{sign}}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Psuedodeterministic signatures *For all messages $m \in \mathcal{M}$, there exists sg^*_m such that*

$$\Pr\left[sg = sg^*_m : \begin{array}{c} sk, vk \leftarrow \mathsf{Setup}(1^\lambda) \\ R_{\mathsf{sign}} \leftarrow \mathsf{QKeyGen}(sk) \\ sg \leftarrow \mathsf{Sign}(R_{\mathsf{sign}}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Similar to the public-key encryption case, psuedodeterministic signatures property implies, via Lemma 1, that we can assume that the signing procedure only negligibly disturbs the key. Hence, we can sign any polynomial number of messages.

We now reproduce the anti-piracy security[12] definition of [LLQZ22]. We will require that an adversary, given the protected key, cannot produce two (possibly entangled) adversaries that can simultaneously sign random challenge messages. Note that in both the anti-piracy security and later in the LOCC leakage-resilience security definition, we will require the adversary to sign a random challenge message that is presented to it at the end of the game[13]. This is an inherent requirement since the signatures are classical: If we allow the adversary to choose or even see the challenge message before it splits (in anti-piracy game) or its LOCC protocol is done (in the leakage-resilience game), the adversary can simple sign the challenge message(s) and output it.

Definition 21 (Anti-piracy security for digital signature schemes [LLQZ22])**.** *Consider the following game between the challenger and an adversary $\mathcal{A}$.*

$\underline{\mathsf{Sign} - \mathsf{AntiPiracy}(\lambda, \mathcal{A})}$

1. *Sample $vk, sk \leftarrow \mathsf{DS.Setup}(1^\lambda)$.*
2. *Sample $R_{\mathsf{sign}} \leftarrow \mathsf{DS.QKeyGen}(sk)$ and submit vk, R_{sign} to $\mathcal{A}$.*
3. *$\mathcal{A}$ gets access to vk, R_{sign} and produces a pair of registers R_1, R_2.*
4. *The challenger samples $x^*_1, x^*_2 \leftarrow \mathcal{M}$.*
5. *The challenger runs $sg_1 \leftarrow \mathsf{U}_{\mathsf{univ}}(R_1, x^*_1)$ and $sg_2 \leftarrow \mathsf{U}_{\mathsf{univ}}(R_2, x^*_2)$.*
6. *The challenger outputs 1 if and only if $\mathsf{DS.Ver}(vk, x^*_1, sg_1) = 1$ and $\mathsf{DS.Ver}(vk, x^*_2, sg_2) = 1$.*

[12] Also called copy protection

[13] We also make the standard assumption that the message space $\mathcal{M}$ is of superpolynomial size.

We say that the digital signature scheme DS *with key protection satisfies* anti-piracy security *if for any QPT adversary* $\mathcal{A}$*,*

$$\Pr[\mathsf{Sign} - \mathsf{AntiPiracy}(\lambda, \mathcal{A}) = 1] \leq \mathsf{negl}(\lambda).$$

We now introduce LOCC leakage-resilience for digital signature schemes.

Definition 22 (LOCC leakage-resilience for digital signature schemes)**.** *Consider the following game between the challenger and an LOCC adversary* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$*.*

$\underline{\mathsf{Sign} - \mathsf{LOCC}(\lambda, \mathcal{A})}$

1. *Sample* $vk, sk \leftarrow \mathsf{DS.Setup}(1^\lambda)$*.*
2. *Sample* $R_{\mathsf{sign}} \leftarrow \mathsf{DS.QKeyGen}(sk)$ *and submit* vk, R_{sign} *to* $\mathcal{A}_{\mathsf{Leak}}$*.*
3. $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *execute their LOCC protocol.*
4. *After the LOCC protocol is over, challenger samples* $x^* \leftarrow \mathcal{M}$ *and submits* x^* *to* $\mathcal{A}_{\mathsf{Main}}$*.*
5. $\mathcal{A}_{\mathsf{Main}}$ *outputs a forged signature* sg*.*
6. *The challenger outputs* 1 *if and only if* $\mathsf{DS.Ver}(vk, x^*, sg) = 1$*.*

We say that the digital signature scheme DS *with key protection satisfies* $g(\lambda)$-round LOCC leakage-resilience *if for any QPT* $g(\lambda)$*-round LOCC adversary* $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$*,*

$$\Pr[\mathsf{Sign} - \mathsf{LOCC}(\lambda, \mathcal{A}) = 1] \leq \mathsf{negl}(\lambda).$$

If DS *satisfies the above for any polynomial* $g(\lambda)$*, then we simply say that it satisfies* LOCC leakage-resilience*.*

Remark 3. *Since both definitions above are with respect to random challenge messages, they do not imply existential unforgeability. Therefore, we will separately require that a digital signature scheme satisfies regular EUF-CMA security (Definition 6).*

We also make the following observations regarding these definitions.

Remark 4. *$g(\lambda)$-round LOCC leakage-resilience implies $h(\lambda)$-round LOCC leakage-resilience for any $h(\lambda) < g(\lambda)$.*

Remark 5. *$g(\lambda)$-round LOCC leakage-resilience for $g(\lambda) > 1$ can be considered an adaptive leakage model since the main adversary can send the description of an adaptive leakage circuit to leakage adversary, who then executes it on the public verification key vk and the quantum key.*

6.1 Relationship Between Anti-Piracy Security and Leakage-Resilience

In this section, we show various results regarding the relationship between anti-piracy security and leakage-resilience.

Theorem 24. *Let* DS *be a digital signature scheme with key protection that satisfies anti-piracy security. Then, it also satisfies* 1*-round LOCC leakage-resilience.*

Proof. Follows from the same argument as Theorem 16: When reducing to the anti-piracy game, we simply output the classical leakage obtained using $\mathcal{A}_{\mathsf{Leak}}$ twice. Like Theorem 16, our result is true in both classical advice and non-uniform quantum advice models for adversaries. □

Theorem 25. *Let* DS *be a digital signature scheme with key protection that satisfies anti-piracy security against adversaries with non-uniform quantum advice. Then, it also satisfies* 2*-round LOCC leakage-resilience against adversaries with non-uniform quantum advice.*

Proof. Follows from the same argument as Theorem 17: When reducing to the anti-piracy game, we simply define the non-uniform quantum advice of the adversary so that it automatically obtains two copies of its state. □

6.2 Coset State-Based Construction

In this section, we show that anti-piracy secure digital signature scheme construction of [LLQZ22] also satisfies LOCC leakage-resilience. For completeness, we first recall their construction, slightly modified to match our notation and the parameters we require for LOCC leakage-resilience.

Set $d_0(\lambda) = c(\lambda)$. Pick some $d_2(\lambda)$ such that $d_2(\lambda) - d_0(\lambda)$ is large enough to describe some circuits that will be defined in the proof. Pick $d_1(\lambda) \geq 2 \cdot d_2(\lambda) + \lambda$. Set $n(\lambda) = d_0(\lambda) + d_1(\lambda) + d_2(\lambda)$, and it will also be the message length.

Assume the existence of following schemes.

- $i\mathcal{O}$, $2^{-\lambda^{c_{\mathsf{iO}}}}$-secure indistinguishability obfuscation scheme,
- CCObf, compute-and-compare obfuscation for $2^{-\lambda^{0.5 \cdot C_{\mathsf{CompLOCC}}}}$-unpredictable and $2^{-m(\lambda)}$-unpredictable distributions.
- F_1, an extracting PRF family with input length $n(\lambda) = d_0(\lambda) + d_1(\lambda) + d_2(\lambda)$, output length $m(\lambda)$ and extraction error probability $2^{-\lambda-1}$ for min-entropy n.
- F_2, a puncturable statistically injective PRF family with failure probability $2^{-\lambda}$, input length d_2 and output length d_1.
- F_3, a puncturable PRF family with input length d_1 and output length d_2.
- f, a one-way function $\{0,1\}^{m(\lambda)} \to \{0,1\}^{m(\lambda)}$.

$\underline{\mathsf{DS.Setup}(1^\lambda)}$

1. Sample PRF keys $K_i \leftarrow F_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in [3]$.
2. Sample $\mathsf{OPVer} \leftarrow i\mathcal{O}(\mathsf{PVer})$ where PVer is the following program.

   $\underline{\mathsf{PVer}(x, sg)}$

   Hardcoded: K_1, K_2, K_3

   1. Parse $x_0||x_1||x_2 = x$ with $|x_i| = d_i$.
   2. Parse $x_0'||Q' = F_3(K_3, x_1) \oplus x_2$ with $|x_0'| = d_0$.
   3. If $x_0 = x_0'$ and $x_1 = F_2(K_2, x_0'||Q')$, then interpret Q' as a classical circuit and output $Q'(mode = check, sg||0^{d_0(\lambda) \cdot \lambda - m(\lambda)})$ and terminate.
   4. Output 1 if and only if $f(F_1(K_1, x)) = f(sg)$. Otherwise, output 0.
3. Set $vk = \mathsf{OPVer}$ and $sk = (K_1, K_2, K_3)$.
4. Output (vk, sk).

$\underline{\mathsf{DS.QKeyGen}(sk)}$

1. Parse $(K_1, K_2, K_3) = sk$.
2. Sample $(A_i, s_i, s'_i)_{i\in[c(\lambda)]} \leftarrow \mathsf{CosetGen}(1^\lambda)$.
3. For $i \in [c(\lambda)]$,
   3.1. Sample $\mathsf{OP}^0_i \leftarrow i\mathcal{O}(A_i + s_i)$.
   3.2. Sample $\mathsf{OP}^1_i \leftarrow i\mathcal{O}(A_i^\perp + s'_i)$.
4. Sample $\mathsf{OPSign} \leftarrow i\mathcal{O}(\mathsf{PSign})$ where PSign is the following program.

   $\underline{\mathsf{PSign}(x, u_1, \ldots, u_{c(\lambda)})}$

   Hardcoded: $K_1, K_2, K_3, (\mathsf{OP}^0_i, \mathsf{OP}^1_i)_{i\in[c(\lambda)]}$

   1. Parse $x_0||x_1||x_2 = x$ with $|x_i| = d_i$.
   2. Parse $x'_0||Q' = F_3(K_3, x_1) \oplus x_2$ with $|x'_0| = d_0$.
   3. If $x_0 = x'_0$ and $x_1 = F_2(K_2, x'_0||Q')$, then interpret Q' as a classical circuit and output $Q'(mode = eval, u_1|| \ldots ||u_{c(\lambda)})$ and terminate.
   4. For $i \in [c(\lambda)]$, check if $\mathsf{OP}^0_i(u_i) = 1$ if $(x_0)_i = 0$ and if $\mathsf{OP}^1_i(u_i) = 1$ if $(x_0)_i = 1$.
   5. Output $F_1(K_1, x)$ if all the checks pass. Otherwise, output $\perp$.

5. Output $\left(\left|A_{i,s_i,s'_i}\right\rangle\right)_{i\in[c(\lambda)]}, \mathsf{OPSign}$.

$\underline{\mathsf{DS.Sign}(R_{\mathsf{key}}, x)}$

1. Parse $((R_i)_{i\in[c(\lambda)]}, \mathsf{OPSign}) = R_{\mathsf{key}}$.
2. Parse $x_0||x_1||x_2 = x$ with $|x_i| = d_i$.
3. For indices $i \in [c(\lambda)]$ such that $(x_0)_i = 1$, apply $H^{\otimes\lambda}$ to R_i.
4. Run the program OPSign coherently on $(R_i)_{i\in[c(\lambda)]}$ and x.
5. Measure the output register and output the measurement outcome.

$\underline{\mathsf{DS.Ver}(vk, x, sg)}$

1. Parse $vk = \mathsf{OPVer}$.
2. Output $\mathsf{OPVer}(x, sg)$.

Theorem 26 ([LLQZ22])**.** DS *satisfies correctness, psuedodeterministic signatures property, EUF-CMA security and anti-piracy security.*

Theorem 27. DS *satisfies LOCC leakage-resilience.*

When we instantiate the assumed building blocks with known constructions, we get the following corollary.

Corollary 4. *Assuming subexponentially secure $i\mathcal{O}$, one-way functions and polynomially hard qLWE, there exists a digital signature scheme that satisfies LOCC leakage-resilience.*

Proof of Security

In this section, we prove Theorem 27. Our proof uses the puncturing argument employed in the proof of copy-protection security of DS given by [LLQZ22]. The argument relates the security of DS to the security of PKE (Section 5.3).

We start by reproducing the following result related to trigger inputs.

Lemma 5 ([LLQZ22])**.** *Call any input x that passes the tests in Step 3 of PVer a hidden trigger input. Then, over the sampling of K_1, K_2, K_3 and $x \leftarrow \{0,1\}^n$, the probability that x is a hidden trigger input is negligible.*

Lemma 6 ([LLQZ22])**.** *We define the algorithm $\mathsf{GenTrigger}$ as follows on input x_0, y, K_2, K_3, and the tuple $(A_i, s_i, s'_i)_{i\in[c(\lambda)]}$:*

1. *Sample $\mathsf{OQ} \leftarrow i\mathcal{O}(\mathsf{Q})$ where Q is the following program.*

   $\underline{\mathsf{Q}(mode, u_1||\ldots||u_{c(\lambda)})}$

   Hardcoded: $y, x_0, (A_i, s_i, s'_i)_{i\in[c(\lambda)]}$

   1. *If $mode = check$,*
      - 1.1. *Set sg to be the first $m(\lambda)$ bits of $u_1||\ldots||u_{c(\lambda)}$.*
      - 1.2. *Output 1 if $sg = y$. Otherwise, output 0.*
      - 1.3. *Terminate.*
   2. *If $mode = eval$,*
      - 2.1. *For $i \in [c(\lambda)]$, check if $u_i \in A_i + s_i$ if $(x_0)_i = 0$ and if $u_i \in A_i^{\perp} + s'_i$ if $(x_0)_i = 1$. Output y if all the checks pass. Otherwise, output $\perp$.*

2. *$q_1 = F_2(K_2, x_0||\mathsf{OQ})$.*
3. *$q_2 \leftarrow F_2(K_3, q_1) \oplus (x_0||\mathsf{OQ})$.*
4. *Output $x_0||q_1||q_2$.*

Then, even given the keys R_{key}, vk, we have that $\mathsf{GenTrigger}(x_0, F_1(K_1, x), K_2, K_3, (A_i, s_i, s'_i)_{i\in[c(\lambda)]})$ is indistinguishable from a uniformly random sample from $\{0,1\}^n$ when $x \leftarrow \{0,1\}^n$.

We prove security through a series of hybrids, each of which is constructed by modifying the previous one.

$\underline{\mathsf{Hyb}_0}$**:** The original security game $\mathsf{Sign} - \mathsf{LOCC}(\mathcal{A}, \lambda)$.

$\underline{\mathsf{Hyb}_1}$**:** After sampling the challenge input x^*, the challenger computes $y^* = F_1(K_1, x^*)$. We change the way the challenger verifies the forged signature sg at the end of the game: the challenger outputs 1 if and only if $sg = y^*$.

$\underline{\mathsf{Hyb}_2}$**:** After sampling the challenge input x^*, the challenger parses it as $x_0^*||x_1^*||x_2^*$ with $|x_i^*| = d_i$. Then, it samples $x_0^{**}||x_1^{**}||x_2^{**} \leftarrow \mathsf{GenTrigger}(x_0^*, y^*, K_2, K_3, (A_i, s_i, s'_i)_{i\in[c(\lambda)]})$. Finally, it now submits x^{**} to the adversary as the challenge input instead of x^*. However, at the end it still checks the forged signature sg by comparing to y^*.

$\underline{\mathsf{Hyb}_3}$**:** The challenger now samples y^* uniformly at random from the output space of F_1.

$\underline{\mathsf{Hyb}_4}$**:** We open up the sampling of x^{**}.

1. Sample $\mathsf{OQ} \leftarrow i\mathcal{O}(\mathsf{Q})$ where Q is the following program.

> $\underline{\mathsf{Q}(mode, u_1||\ldots||u_{c(\lambda)})}$
>
> **Hardcoded:** $y^*, x_0^*, (A_i, s_i, s_i')_{i\in[c(\lambda)]}$
>
> 1. If $mode = check$,
>
> 1.1. Set sg to be the first $m(\lambda)$ bits of $u_1||\ldots||u_{c(\lambda)}$.
>
> 1.2. Output 1 if $sg = y^*$. Otherwise, output 0.
>
> 1.3. Terminate.
>
> 2. If $mode = eval$,
>
> 2.1. For $i \in [c(\lambda)]$, check if $u_i \in A_i + s_i$ if $(x_0^*)_i = 0$ and if $u_i \in A_i^{\perp} + s_i'$ if $(x_0^*)_i = 1$. Output y^* if all the checks pass. Otherwise, output $\perp$.

2. $q_1 = F_2(K_2, x_0^*||\mathsf{OQ})$.
3. $q_2 \leftarrow F_2(K_3, q_1) \oplus (x_0^*||\mathsf{OQ})$.
4. Set $x_0^{**} = x_0^*$, $x_1^{**} = q_1$ and $x_2^{**} = q_2$.

$\underline{\mathsf{Hyb}_5}$**:** We change the way we sample OQ above. First compute

1. For $i \in [c(\lambda)]$, set $g_i = \mathsf{Can}_{A_i}$ if $(x_0^{**})_i = 0$ and set $g_i = \mathsf{Can}_{(A_i)^{\perp}}$ if $(x_0^{**})_i = 1$.
2. For $i \in [c(\lambda)]$, compute $w_i = g_i(s_i)$ if $(x_0^{**})_i = 0$ and $w_i = g_i(s_i')$ if $(x_0^{**})_i = 1$.
3. Set g to be the function $g(v_1, \ldots, v_{c(\lambda)}) = (g_1(v_1)||\ldots||g_{c(\lambda)}(v_{c(\lambda)}))$.
4. Set $w = w_1||\ldots||w_{c(\lambda)}$.
5. Compute $\mathsf{OCC} \leftarrow \mathsf{CCObf.Obf}(g, w, y^*)$.

Then, sample $\mathsf{OQ} \leftarrow i\mathcal{O}(\mathsf{Q}')$ where Q' is the following program.

> $\underline{\mathsf{Q}'(mode, u_1, \ldots, u_{c(\lambda)})}$
>
> **Hardcoded:** y^*, OCC
>
> 1. If $mode = check$,
>
> 1.1. Set sg to be the first $m(\lambda)$ bits of $u_1||\ldots||u_{c(\lambda)}$.
>
> 1.2. Output 1 if $sg = y^*$. Otherwise, output 0.
>
> 1.3. Terminate.
>
> 2. If $mode = eval$,
>
> 2.1. Output $\mathsf{OCC}(u_1, \ldots, u_{c(\lambda)})$.

$\underline{\mathsf{Hyb}_6}$**:** We first sample $\mathsf{PSim} \leftarrow \mathsf{CCObf.Sim}(1^\lambda, |g|, |y|, |m|)$. Then, we now sample $\mathsf{OQ} \leftarrow i\mathcal{O}(\mathsf{Q}'')$ where Q'' is the following program.

$\underline{\mathsf{Q}''(mode, u_1, \ldots, u_{c(\lambda)})}$

Hardcoded: y^*, PSim

1. If $mode = check$,
   - 1.1. Set sg to be the first $m(\lambda)$ bits of $u_1 || \ldots || u_{c(\lambda)}$.
   - 1.2. Output 1 if $sg = y^*$. Otherwise, output 0.
   - 1.3. Terminate.
2. If $mode = eval$,
   - 2.1. Output $\mathsf{PSim}(u_1, \ldots, u_{c(\lambda)})$.

$\underline{\mathsf{Hyb}_7}$**:** Let CC denote the following compute-and-compare program. On an input sg of length $c(\lambda) \cdot \lambda$, it parses out the first $m(\lambda)$ bits and compares it to y^*. On equality, it outputs 1 as its hidden value. Then, sample $\mathsf{OCC}' \leftarrow \mathsf{CCObf.Obf}(\mathsf{CC}, y^*, 1)$.

We now sample OQ as $\mathsf{OQ} \leftarrow i\mathcal{O}(\mathsf{Q}''')$ where Q''' is the following program.

$\underline{\mathsf{Q}'''(mode, u_1, \ldots, u_{c(\lambda)})}$

Hardcoded: $\mathsf{PSim}, \mathsf{OCC}'$

1. If $mode = check$,
   - 1.1. Compute $\mathsf{OCC}'(u_1, \ldots, u_{c(\lambda)})$. If it outputs 1, output 1. Otherwise, output 0.
   - 1.2. Terminate.
2. If $mode = eval$,
   - 2.1. Output $\mathsf{PSim}(u_1, \ldots, u_{c(\lambda)})$.

$\underline{\mathsf{Hyb}_8}$**:** First sample $\mathsf{PSim}' \leftarrow \mathsf{CCObf.Sim}(1^\lambda, |\mathsf{CC}|, |y^*|, 1)$. We now sample $\mathsf{OQ} \leftarrow i\mathcal{O}(\mathsf{Q}'''')$ where Q'''' is the following program.

$\underline{\mathsf{Q}''''(mode, u_1, \ldots, u_{c(\lambda)})}$

Hardcoded: $\mathsf{PSim}, \mathsf{PSim}'$

1. If $mode = check$,
   - 1.1. Compute $\mathsf{PSim}'(u_1, \ldots, u_{c(\lambda)})$. If it outputs 1, output 1. Otherwise, output 0.
   - 1.2. Terminate.
2. If $mode = eval$,
   - 2.1. Output $\mathsf{PSim}(u_1, \ldots, u_{c(\lambda)})$.

Lemma 7. $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$.

Proof. Since the hidden trigger set is sparse (i.e., negligibly small) by Lemma 5, with all but negligible probability x^* will be outside this set. Hence, the check $sg = y^*$ is equivalent to comparing to $\mathsf{DS.Ver}(vk, x^*, sg)$. □

Lemma 8. $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$.

Proof. Follows from Lemma 6. □

Lemma 9. $\mathsf{Hyb}_2 \approx \mathsf{Hyb}_3$.

Proof. $\mathsf{GenTrigger}$ only uses x_0^* rather than all of x^*. More generally, x_1^* and x_2^* are independent from the view of the adversary. Hence, x^* has min-entropy at least $d_1 + d_2$. The result follows from the extracting property of F_1.

□

Lemma 10. $\mathsf{Hyb}_3 \approx \mathsf{Hyb}_4$.

Proof. These hybrids are exactly the same. □

Lemma 11. $\mathsf{Hyb}_4 \approx \mathsf{Hyb}_5$.

Proof. Similar to Claim 7, the programs Q and Q' have the exact same functionality. The result follows by security of $i\mathcal{O}$. □

Lemma 12. $\mathsf{Hyb}_5 \approx \mathsf{Hyb}_6$.

Proof. Similar to Claim 8, the target vector values are unpredictable by the LOCC security of the coset states (Theorem 10). Note that the *check* part of the program and the verification key vk are sampled independently of the coset states, therefore, they do not affect this property. Finally, the signing program OPSign is correlated with the coset states, it is only through the membership checking programs, of which it is an efficient function. Therefore, the unpredictability still remains, and the proof follows as in Claim 8. □

Lemma 13. $\mathsf{Hyb}_6 \approx \mathsf{Hyb}_7$.

Proof. Q'' and Q''' have the exact same functionalty by correctness of CCObf. The result follows by security of $i\mathcal{O}$. □

Lemma 14. $\mathsf{Hyb}_7 \approx \mathsf{Hyb}_8$.

Proof. Observe that y^* is independent of the view of the adversary (except for the program OCC'). Hence, given the rest of the view of the adversary and the program description CC, it is $2^{-m(\lambda)}$-unpredictable. The result follows by the security of CCObf. □

In Hyb_8, the adversary is required to output y^*, which is sampled uniformly at random from $\{0,1\}^{m(\lambda)}$ and is independent of view of the adversary. Therefore, we have that $\Pr[\mathsf{Hyb}_8 = 1] \leq 2^{-m(\lambda)}$. By above, this implies $\Pr[\mathsf{Sign} - \mathsf{LOCC}(\lambda, \mathcal{A})] \leq \mathsf{negl}(\lambda)$, completing the proof.

7 Pseudorandom Function Families with Key Protection

In this section, we introduce the concept of psuedorandom function families with key protection and formally define LOCC leakage-resilience for such schemes. Then, we show that the construction of [CLLZ21] of a psuedorandom function family that satisfies anti-piracy security also satisfies LOCC leakage-resilience.

Definition 23 (Pseudorandom function family with key protection)**.** *A pseudorandom function family with key protection consists of the following QPT algorithms.*

- $\mathsf{Setup}(1^\lambda)$: *Outputs a classical PRF key k for a PRF family $\mathcal{F} = \{f_k\}_k$.*
- $\mathsf{QKeyGen}(k)$: *Takes the classical key k and outputs a quantum key register R_{key}.*
- $\mathsf{Eval}(R_{\mathsf{key}}, x)$*: Takes a quantum key and an input x and returns the evaluation $f_k(x)$.*

We require that the scheme satisfies correctness. Let $\mathcal{X}$ denote the input space.

Correctness *For all inputs $x \in \mathcal{X}$,*

$$\Pr\left[\mathsf{Eval}(R_{\mathsf{key}}, x) = f_k(x) : \begin{array}{c} k \leftarrow \mathsf{Setup}(1^\lambda) \\ R_{\mathsf{key}} \leftarrow \mathsf{QKeyGen}(k) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Similar to public-key encryption, correctness and Lemma 1 implies that we can correctly evaluate the PRF on any polynomial number of points.

We first reproduce the anti-piracy security definitions of [CLLZ21]. Note that in both the anti-piracy security and later in the LOCC leakage-resilience security definition, similar to the case of digital signatures, we will require the adversary to either predict or distinguish the evaluation of the PRF on a *random* challenge input[14]. This is an inherent requirement since if we allow the adversary choose the challenge input, it can simply evaluate the PRF at that point and leak or clone the classical evaluation result.

Similar to public-key encryption, we can define two different variations: indistinguishability and unpredictability security.

Definition 24 (Unpredictability anti-piracy security for PRFs [CLLZ21])**.** *Consider the following game between the challenger and an adversary $\mathcal{A}$.*

$\mathsf{PRF} - \mathsf{AntiPiracy} - \mathsf{Guess}(\lambda, \mathcal{A})$

1. *Sample $k \leftarrow \mathsf{PRF.Setup}(1^\lambda)$.*
2. *Sample $R_{\mathsf{key}} \leftarrow \mathsf{PRF.QKeyGen}(k)$ and submit R_{key} to $\mathcal{A}$.*
3. *$\mathcal{A}$ gets access to R_{key} and produces a pair of registers R_1, R_2.*
4. *The challenger samples $x_1^*, x_2^* \leftarrow \mathcal{X}$.*
5. *The challenger runs $y_1' \leftarrow \mathsf{U}_{\mathsf{univ}}(R_1, x_1^*)$ and $y_2' \leftarrow \mathsf{U}_{\mathsf{univ}}(R_2, x_2^*)$.*
6. *The challenger outputs 1 if and only if $y_1' = f_k(x_1^*)$ and $y_2' = f_k(x_2^*)$.*

[14] We also make the standard assumption that the input space $\mathcal{X}$ is of superpolynomial size.

We say that the PRF family PRF *with key protection satisfies* unpredictability anti-piracy security *if for any QPT adversary* $\mathcal{A}$*,*

$$\Pr[\mathsf{PRF}-\mathsf{AntiPiracy}-\mathsf{Guess}(\lambda,\mathcal{A})=1]\leq \mathsf{negl}(\lambda).$$

Definition 25 (Indistinguishability anti-piracy security for PRFs [CLLZ21])**.** *Consider the following game between the challenger and an adversary* $\mathcal{A}$*. Let* $\mathcal{Y}$ *denote the output space of the PRF family.*

$\underline{\mathsf{PRF}-\mathsf{AntiPiracy}-\mathsf{IND}(\lambda,\mathcal{A})}$

1. *Sample* $k\leftarrow \mathsf{PRF.Setup}(1^\lambda)$*.*
2. *Sample* $R_{\mathsf{key}}\leftarrow \mathsf{PRF.QKeyGen}(k)$ *and submit* R_{key} *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *gets access to* R_{key} *and produces a pair of registers* R_1,R_2*.*
4. *The challenger samples* $x_1^*,x_2^*\leftarrow\mathcal{X}$ *and* $b_1,b_2\leftarrow\{0,1\}$*. It sets* $y_1^*=f_k(x_1^*)$ *and* $y_2^*=f_k(x_2^*)$*.*
5. *The challenger samples* $z_1^*\leftarrow\mathcal{Y}$ *and* $z_2^*\leftarrow\mathcal{Y}$*.*
6. *The challenger runs* $b_1'\leftarrow \mathsf{U}_{\mathsf{univ}}(R_1,x_1^*,y_1^*)$ *if* $b_1=0$ *and* $b_1'\leftarrow \mathsf{U}_{\mathsf{univ}}(R_1,x_1^*,z_1^*)$ *if* $b_1=1$*.*
7. *The challenger runs* $b_2'\leftarrow \mathsf{U}_{\mathsf{univ}}(R_2,x_2^*,y_2^*)$ *if* $b_2=0$ *and* $b_2'\leftarrow \mathsf{U}_{\mathsf{univ}}(R_2,x_2^*,z_2^*)$ *if* $b_2=1$*.*
8. *The challenger outputs* 1 *if and only if* $b_1'=b_1$ *and* $b_2'=b_2$*.*

We say that the PRF family PRF *with key protection satisfies* indistinguishability anti-piracy security *if for any QPT adversary* $\mathcal{A}$*,*

$$\Pr[\mathsf{PRF}-\mathsf{AntiPiracy}-\mathsf{IND}(\lambda,\mathcal{A})=1]\leq \frac{1}{2}+\mathsf{negl}(\lambda).$$

For anti-piracy security, unlike the classical case, indistinguishability security is not known to imply unpredictability security since there are two adversaries that need to simultaneously distinguish the PRF output from a random value. See [CLLZ21] for a discussion.

We now introduce LOCC leakage-resilience for PRF families.

Definition 26 (Unpredictability LOCC leakage-resilience for PRFs)**.** *Consider the following game between the challenger and an LOCC adversary* $\mathcal{A}=(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$*.*

$\underline{\mathsf{PRF}-\mathsf{LOCC}-\mathsf{Guess}(\lambda,\mathcal{A})}$

1. *Sample* $k\leftarrow \mathsf{PRF.Setup}(1^\lambda)$*.*
2. *Sample* $R_{\mathsf{key}}\leftarrow \mathsf{PRF.QKeyGen}(k)$ *and submit* R_{key} *to* $\mathcal{A}_{\mathsf{Leak}}$*.*
3. $(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$ *execute their LOCC protocol.*
4. *After the LOCC protocol is over, the challenger samples* $x^*\leftarrow\mathcal{X}$ *and submits it to* $\mathcal{A}_{\mathsf{Main}}$*.*
5. $\mathcal{A}_{\mathsf{Main}}$ *outputs a guess* y'*.*
6. *The challenger outputs* 1 *if and only if* $y'=f_k(x^*)$*.*

We say that the PRF family PRF *with key protection satisfies* unpredictability $g(\lambda)$-round LOCC leakage-resilience *if for any QPT $g(\lambda)$-round LOCC adversary $\mathcal{A}$,*

$$\Pr[\mathsf{PRF}-\mathsf{LOCC}-\mathsf{Guess}(\lambda,\mathcal{A})=1]\leq \mathsf{negl}(\lambda).$$

If PRF *satisfies the above for any polynomial $g(\lambda)$, we simply say that it satisfies* unpredictability LOCC leakage-resilience*.*

Definition 27 (Indistinguishability LOCC leakage-resilience for PRFs)**.** *Consider the following game between the challenger and an LOCC adversary $\mathcal{A}=(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$. Let $\mathcal{Y}$ denote the output space of the PRF family.*

$\underline{\mathsf{PRF}-\mathsf{LOCC}-\mathsf{IND}(\lambda,\mathcal{A})}$

1. *Sample $k\leftarrow \mathsf{PRF.Setup}(1^\lambda)$.*
2. *Sample $R_{\mathsf{key}}\leftarrow \mathsf{PRF.QKeyGen}(k)$ and submit R_{key} to $\mathcal{A}_{\mathsf{Leak}}$.*
3. *$(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$ execute their LOCC protocol.*
4. *After the LOCC protocol is over, the challenger samples $x^*\leftarrow\mathcal{X}$, $b\leftarrow\{0,1\}$ and $z^*\leftarrow\mathcal{Y}$. It sets $y^*=f_k(x^*)$.*
5. *If $b=0$, the challenger submits x^*,y^* to $\mathcal{A}_{\mathsf{Main}}$. Otherwise, it submits x^*,z^*.*
6. *$\mathcal{A}_{\mathsf{Main}}$ outputs a guess b'.*
7. *The challenger outputs 1 if and only if $b'=b$.*

We say that the PRF family PRF *with key protection satisfies* indistinguishability $g(\lambda)$-round LOCC leakage-resilience *if for any QPT $g(\lambda)$-round LOCC adversary $\mathcal{A}$,*

$$\Pr[\mathsf{PRF}-\mathsf{LOCC}-\mathsf{Guess}(\lambda,\mathcal{A})=1]\leq \frac{1}{2}+\mathsf{negl}(\lambda).$$

If PRF *satisfies the above for any polynomial $g(\lambda)$, we simply say that it satisfies* indistinguishability LOCC leakage-resilience*.*

We make the following observations regarding these definitions.

Remark 6. *$g(\lambda)$-round LOCC leakage-resilience implies $h(\lambda)$-round LOCC leakage-resilience for any $h(\lambda)<g(\lambda)$.*

Remark 7. *LOCC leakage-resilience implies regular weak PRF security (Definition 2) or even a relaxed version of regular PRF security where the adversary can query the PRF at arbitrary points but at the end it is presented with a challenge input that is chosen uniformly at random. Both implications follow from the fact that the leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ can simply simulate the regular (weak) PRF adversary's queries and leak the results.*

Like the classical case, single-challenge indistinguishability security implies the multi-challenge version where the adversary is either presented with $p(\lambda)$ evaluations of the PRF ($b=0$) or $p(\lambda)$ random samples from the output space $\mathcal{Y}$ (the case $b=1$).

Lemma 15. *Suppose the PRF family* PRF *with key protection satisfies single-challenge indistinguishability $g(\lambda)$-round LOCC leakage-resilience. Then, it also satisfies multi-challenge indistinguishability $g(\lambda)$-round LOCC leakage-resilience.*

Proof. A simple analogue of the classical hybrid argument yields the result in a straightforward manner, since there is only a single adversary that needs to predict the challenge bit b. □

7.1 Relationship Between Indistinguishability and Unpredictability Leakage-Resilience

In this section, we show that similar to the classical case, indistinguishability security implies unpredictability security in the LOCC leakage model. Then, we show that any scheme satisfying the latter can be used to construct a scheme satisfying the former, by using randomness extractors (more specifically, Goldreich-Levin bits).

Theorem 28. *Let* PRF *be a PRF scheme with key protection that satisfies indistinguishability* $g(\lambda)$*-round LOCC leakage-resilience. Then, it also satisfies unpredictability* $g(\lambda)$*-round LOCC leakage-resilience.*

The results follows from the standard reduction.

Proof. Suppose for a contradiction that there exists a $g(\lambda)$-round QPT LOCC adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the unpredictability LOCC leakage-resilience game with probability $1/q(\lambda)$ for infinitely many values of $\lambda > 0$ where $q(\cdot)$ is a polynomial. We construct an adversary $\mathcal{A}' = (\mathcal{A}'_{\mathsf{Main}}, \mathcal{A}'_{\mathsf{Leak}})$ for the indistinguishability LOCC leakage-resilience game as follows.

We define $\mathcal{A}'_{\mathsf{Leak}}$ to be the same as $\mathcal{A}_{\mathsf{Leak}}$. $\mathcal{A}'_{\mathsf{Main}}$ is defined to be the same as $\mathcal{A}_{\mathsf{Main}}$ until the end of the LOCC protocol, but we only change the challenge-response part of $\mathcal{A}_{\mathsf{Main}}$. After the protocol is over, when $\mathcal{A}'_{\mathsf{Main}}$ receives the challenge input (x^*, a^*), it runs $\mathcal{A}_{\mathsf{Main}}$ on x^* (and its state) to obtain a guess y'. If $y' = a^*$ it outputs 0, otherwise it outputs 1.

A simple calculation akin to the proof of Theorem 14 shows that $\mathcal{A}'$ wins the indistinguishability LOCC leakage-resilience game with probability $1/2+1/(2 \cdot q(\lambda))-\mathsf{negl}(\lambda)$, which is a contradiction. □

Now we show that we can construct an indistinguishability leakage-resilient scheme in a black-box way from a scheme that is only unpredictable leakage-resilient.

Theorem 29. *Let* $\mathcal{F}' = \{f'_k\}_k$ *be a wPRF family with input space* $\{0,1\}^{p_1(\lambda)}$ *and output space* $\{0,1\}^{p_2(\lambda)}$*. Let* PRF' *be an unpredictable* $g(\lambda)$*-round LOCC leakage-resilient protection scheme for* $\mathcal{F}'$*. Then,* PRF *constructed below is an indistinguishability* $g(\lambda)$*-round LOCC leakage-resilient protection scheme for* $\mathcal{F}'$ *wPRF* $\mathcal{F} = \{f_k\}_k$ *defined below.*

$\underline{\mathcal{F} = \{f_k\}_k}$

- *Key distribution:* *Same as* $\mathcal{F}'$
- *Input space:* $\{0,1\}^{p_1(\lambda)} \times \{0,1\}^{p_1(\lambda)}$
- *Output space:* $\{0,1\}^1$
- *Evaluation:* $f_k(x_1||x_2) = \langle f'_k(x_1), x_2\rangle$

$\underline{\mathsf{PRF}}$

- $\underline{\mathsf{PRF.Setup}(1^\lambda)}$
  Same as $\mathsf{PRF}'.\mathsf{Setup}$.
- $\underline{\mathsf{PRF.QKeyGen}(k)}$
  Same as $\mathsf{PRF}'.\mathsf{QKeyGen}$.

- $\underline{\mathsf{PRF.Eval}(R_{\mathsf{key}}, x_1||x_2)}$

  1. *Output* $\langle \mathsf{PRF}'.\mathsf{Eval}(R_{\mathsf{key}}, x_1), x_2 \rangle$.

Proof. Follows from the same argument as Theorem 15. □

While the PRF family constructed above has 1-bit output, it can be generalized to have any polynomial output length: We simply sample $q(\lambda)$ independent instances of PRF and to evaluate it on some x, we evaluate x on indiviual schemes and concatenate the outputs. The resulting scheme satisfies indistinguishability LOCC leakage-resilience in a straightforward manner.

7.2 Relationship Between Anti-Piracy Security and Leakage-Resilience

In this section, we show various results regarding the relationship between anti-piracy security and leakage-resilience.

Theorem 30. *Let* PRF *be a PRF family with key protection that satisfies unpredictability anti-piracy security. Then, it also satisfies unpredictability* 1*-round LOCC leakage-resilience.*

Proof. Follows from the same argument as Theorem 16: When reducing to the anti-piracy game, we simply output the classical leakage obtained using $\mathcal{A}_{\mathsf{Leak}}$ twice. Like Theorem 16, our result is true in both classical advice and non-uniform quantum advice models for adversaries. □

Combining with our previous results, we obtain the following corollary.

Corollary 5. *Suppose there exists a PRF family with key protection that satisfies unpredictability anti-piracy security. Then, there exists a PRF family with key protection that uses the former in a black-box way and satisfies indistinguishability* 1*-round LOCC leakage-resilience.*

Theorem 31. *Let* PRF *be a PRF family with key protection that satisfies unpredictability anti-piracy security against adversaries with non-uniform quantum advice. Then, it also satisfies unpredictability* 2*-round LOCC leakage-resilience against adversaries with non-uniform quantum advice.*

Proof. Follows from the same argument as Theorem 17: When reducing to the anti-piracy game, we simply define the non-uniform quantum advice of the adversary so that it automatically obtains two copies of its state. □

Again, combining with our extractor based transformation from the previous section, we obtain the following result.

Corollary 6. *Suppose there exists a PRF family with key protection that satisfies unpredictability anti-piracy security against adversaries with non-uniform quantum advice. Then, there exists a PRF family with key protection that uses the former in a black-box way and satisfies indistinguishability* 2*-round LOCC leakage-resilience against adversaries with non-uniform quantum advice.*

7.3 Coset State-Based Construction

In this section, we show that anti-piracy secure PRF scheme construction of [CLLZ21] also satisfies LOCC leakage-resilience. Note that the construction is the same as the digital signatures construction in Section 6.2, with verification key and the verification algorithm removed. For completeness, we recall the construction.

Set $d_0(\lambda) = c(\lambda)$. Pick some $d_2(\lambda)$ such that $d_2(\lambda) - d_0(\lambda)$ is large enough to describe some circuits that will be defined in the proof. Pick $d_1(\lambda) \geq 2 \cdot d_2(\lambda) + \lambda$. Set $n(\lambda) = d_0(\lambda) + d_1(\lambda) + d_2(\lambda)$, and it will also be the message length.

Assume the existence of following schemes.

- $i\mathcal{O}$, $2^{-\lambda^{c_{\mathsf{iO}}}}$-secure indistinguishability obfuscation scheme,
- CCObf, compute-and-compare obfuscation for $2^{-\lambda^{0.5 \cdot C_{\mathsf{CompLOCC}}}}$-unpredictable and $2^{-m(\lambda)}$-unpredictable distributions.
- F_1, an extracting PRF family with input length $n(\lambda) = d_0(\lambda) + d_1(\lambda) + d_2(\lambda)$, output length $m(\lambda)$ and extraction error probability $2^{-\lambda-1}$ for min-entropy n.
- F_2, a puncturable statistically injective PRF family with failure probability $2^{-\lambda}$, input length d_2 and output length d_1.
- F_3, a puncturable PRF family with input length d_1 and output length d_2.
- f, a one-way function $\{0,1\}^{m(\lambda)} \to \{0,1\}^{m(\lambda)}$.

$\underline{\mathsf{PRF.Setup}(1^\lambda)}$

1. Sample PRF keys $K_i \leftarrow F_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in [3]$.
2. Set $sk = (K_1, K_2, K_3)$.
3. Output sk.

$\underline{\mathsf{PRF.QKeyGen}(sk)}$

1. Parse $(K_1, K_2, K_3) = sk$.
2. Sample $(A_i, s_i, s'_i)_{i \in [c(\lambda)]} \leftarrow \mathsf{CosetGen}(1^\lambda)$.
3. For $i \in [c(\lambda)]$,
   3.1. Sample $\mathsf{OP}^0_i \leftarrow i\mathcal{O}(A_i + s_i)$.
   3.2. Sample $\mathsf{OP}^1_i \leftarrow i\mathcal{O}(A_i^\perp + s'_i)$.
4. Sample $\mathsf{OPEval} \leftarrow i\mathcal{O}(\mathsf{PEval})$ where PEval is the following program.

$\underline{\mathsf{PEval}(x, u_1, \ldots, u_{c(\lambda)})}$

Hardcoded: $K_1, K_2, K_3, (\mathsf{OP}^0_i, \mathsf{OP}^1_i)_{i \in [c(\lambda)]}$

1. Parse $x_0 || x_1 || x_2 = x$ with $|x_i| = d_i$.
2. Parse $x'_0 || Q' = F_3(K_3, x_1) \oplus x_2$ with $|x'_0| = d_0$.
3. If $x_0 = x'_0$ and $x_1 = F_2(K_2, x'_0 || Q')$, then interpret Q' as a classical circuit and output $Q'(mode = eval, u_1 || \ldots || u_{c(\lambda)})$ and terminate.
4. For $i \in [c(\lambda)]$, check if $\mathsf{OP}^0_i(u_i) = 1$ if $(x_0)_i = 0$ and if $\mathsf{OP}^1_i(u_i) = 1$ if $(x_0)_i = 1$.
5. Output $F_1(K_1, x)$ if all the checks pass. Otherwise, output $\perp$.

5. Output $\left(\left|A_{i,s_i,s_i'}\right\rangle\right)_{i\in[c(\lambda)]}, \mathsf{OPEval}$.

$\underline{\mathsf{PRF.Eval}(R_{\mathsf{key}}, x)}$

1. Parse $((R_i)_{i\in[c(\lambda)]}, \mathsf{OPEval}) = R_{\mathsf{key}}$.
2. Parse $x_0||x_1||x_2 = x$ with $|x_i| = d_i$.
3. For indices $i \in [c(\lambda)]$ such that $(x_0)_i = 1$, apply $H^{\otimes\lambda}$ to R_i.
4. Run the program OPEval coherently on $(R_i)_{i\in[c(\lambda)]}$ and x.
5. Measure the output register and output the measurement outcome.

Theorem 32 ([CLLZ21])**.** PRF *satisfies correctness and both unpredictability and indistinguishability anti-piracy security.*

Theorem 33. PRF *satisfies indistinguishability LOCC leakage-resilience.*

Proof. The scheme is the same as the digital signature scheme from Section 6.2, with only the verification key removed, which makes the security proof even easier. Therefore, the security follows from the same argument as the security proof of Section 6.2. The only difference is that we need to show an indistinguishability security, while in the digital signatures proof, we show an unpredictability security. However, by a close inspection of the security argument there, we can see that the proof shows that the target value (which is the target signature in the signature game) can be made uniformly random and independent of the view of the adversary, through a series of hybrids. Therefore, the initial hybrid there (once we remove the verification program) corresponds to the $b = 0$ case in PRF security game and the final hybrids corresponds to the $b = 1$ case. Hence the adversary has negligible advantage in the indistinguishability LOCC leakage-resilience game. □

When we instantiate the assumed building blocks with known constructions, we get the following corollary.

Corollary 7. *Assuming subexponentially secure $i\mathcal{O}$, one-way functions and polynomially hard qLWE, there exists a PRF scheme that satisfies indistinguishability LOCC leakage-resilience.*

8 Cryptographic Schemes with Intrusion-Detection

In this section, we initiate the study of cryptographic primitives with intrusion-detection, yet another set of schemes that are only possible through utilization of quantum phenomena. We show that such schemes are equivalent to schemes with *publicly verifiable* secure leasing[15].

In our intrusion-detection models, we will require that if our detection algorithm does not detect intrusion after an attack, then the adversary should have negligible advantage in breaking the security of the scheme. This should hold even if the adversary can arbitrarily tamper with the secret. Intuitively, this means that any *useful* (to the adversary) leakage will be detected. Furthermore, we will require that if there was no attack, then testing for intrusion only negligibly

[15] Also called *certified deletion* in the literature.

disturbs the key. This means that the honest party can test for intrusion any polynomial number of times while preserving the correctness and security guarantees, as long as there is no attack.

Now, we describe our transformation on a high level. Suppose there exists a scheme with secure leasing. We will construct a TestIntrusion algorithm that essentially tries to produce a deletion certificate for the secret, and outputs NO INTRUSION if it succeeds. Intuitively, we can argue intrusion-detection security as follows. If an adversary has obtained a leakage that allows it to break the underlying security guarantee, then we should fail to produce a valid deletion certificate using our leftover state. Otherwise, one can create an attacker against the certified deletion game that applies the previous adversary on their secret, produces a valid deletion certificate using the leftover state, and still succeeds in breaking the security guarantee using the *leak* they obtained. The major problem with this approach is that even when there was no attack, we destroy our key when we test for leakage, since we produce a deletion certificate. However, note that producing a valid deletion certificate using an undisturbed secret succeeds with overwhelming probability. Therefore, using Lemma 1, we can construct an algorithm for producing a deletion certificate in a way such that we can rewind our algorithm afterwards. While seemingly contradictory, this is not a violation of the certified deletion security. In the certified deletion game, the certificate generation circuit will end with a measurement, while our intrusion-detection procedure will skip this measurement, and will instead run the verification procedure *coherently*. Furthermore, the intrusion-detection procedure will not trace out the *garbage* registers that are produced while constructing a certificate or testing for certificate validity, which we then use to rewind the algorithm.

Remark 8. *Note that public verification property of the certified deletion scheme is essential to build a intrusion detection scheme, since the leakage adversary will have access to the complete state of the honest parties, including the key that is used to test for leakage.*

8.1 Public-key Encryption with Intrusion-Detection

First, we start with public-key encryption. We define PKE schemes that allow us to test if a *useful* leakage has been obtained on the secret decryption key. Then, we show that such schemes are equivalent to public-key encryption schemes with secure key leasing. Our results hold both for schemes with classical certificates and for schemes with quantum certificates.

Definition 28 (Public-key encryption with intrusion-detection)**.** *A public-key encryption scheme with intrusion-detection is a public-key encryption scheme (Definition 15) with the following additional algorithms that satisfy the reusability and security guarantees below.*

- $\mathsf{QLKeyGen}(sk)$: *Along with a quantum decryption key* R_{dec}[16]*, also outputs a classical intrusion-detection key* tk*.*
- $\mathsf{TestIntrusion}(tk, R_{dec})$: *Takes the intrusion-detection key and the decryption key, outputs* INTRUSION *if* leakage is detected*,* NO INTRUSION *otherwise.*

PKE correctness and security: *We require the usual correctness and security (Definition 15) satisfied for the decryption key generated by* QKeyGen *to now also hold for the decryption key generated by* QLKeyGen*.*

Detection correctness:

$$\Pr\left[\mathsf{TestIntrusion}(tk, R_{\mathsf{dec}}) = \text{NO INTRUSION} : \begin{array}{c} sk, pk \leftarrow \mathsf{Setup}(1) \\ tk, R_{\mathsf{dec}} \leftarrow \mathsf{QLKeyGen}(sk) \end{array}\right] = 1.$$

[16]Note that the public key pk is still classical.

Reusability after testing: *Initialize the decryption register as* $(R_{\mathsf{dec}}, tk) \leftarrow \mathsf{QLKeyGen}(sk)$ *and let* ρ *denote its state. Run the algorithm* TestIntrusion *on* tk *and* R_{dec}*, and let* ρ' *denote the state of the register* R_{dec} *immediately afterwards. Then,* $\|\rho - \rho'\|_1 \leq \mathsf{negl}(\lambda)$.

We note that reusability after testing will follow from detection correctness by utilizing Lemma 1.

Intrusion-detection security: *Consider the following game played by the challenger and an adversary.*

1. *The challenger runs* $sk, pk \leftarrow \mathsf{PKE.Setup}(1)$*, and then*

$$tk, R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QLKeyGen}(sk).$$

2. *The adversary* $\mathcal{A}_{\mathsf{intr}}$ *gets access to* R_{dec}, tk*, and* pk*, and produces a register* R_{adv} *and two challenge messages* m_0, m_1*, along with the* updated *register* R_{dec}.

3. *The challenger runs* $tb \leftarrow \mathsf{PKE.TestIntrusion}(tk, R_{\mathsf{dec}})$*. If* tb *is* INTRUSION*, the challenger outputs* 0 *and terminates.*

4. *The challenger samples a challenge bit* $b \leftarrow \{0,1\}$ *and prepares the ciphertext* $ct \leftarrow \mathsf{PKE.Enc}(pk, m_b)$.

5. $\mathcal{A}_{\mathsf{Main}}$ *gets* $R_{\mathsf{adv}}, m_0, m_1, tk, pk$ *and* ct*, and outputs a prediction* b'.

6. *The challenger outputs* 1 *if* $b' = b$.

We say that the adversary has won the game if the challenger outputs 1 *and we require that any QPT adversary* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ *wins the game with probability at most*

$$\frac{\Pr[tb = \text{NO INTRUSION}]}{2} + \mathsf{negl}(\lambda).$$

Similarly to the notions of secure software leasing [AL21] and functional encryption with secure key leasing [KN22], we define public-key encryption schemes with secure key leasing.

Definition 29 (Public-key encryption with secure key leasing)**.** *A public-key encryption scheme with secure key leasing is a public-key encryption scheme (Definition 15) with the following additional algorithms that satisfies the correctness and security guarantees below.*

- $\mathsf{QVKeyGen}(sk)$: *Along with a quantum decryption key* R_{dec}*, also outputs a classical verification key* cvk.
- $\mathsf{Cert}(R_{dec})$: *Takes the decryption key and outputs a certificate.*
- $\mathsf{Verify}(cvk, R_{\mathsf{cert}})$: *Takes the verification key and a deletion certificate, outputs* VALID *if it is a valid certificate.*

PKE correctness and security: *We require the usual correctness and security (Definition 15) satisfied for the decryption key generated by* QKeyGen *to now also hold for the decryption key generated by* QVKeyGen.

Verification correctness:

$$\Pr\left[\mathsf{Verify}(cvk, R_{\mathsf{cert}}) = \mathsf{VALID} : \begin{array}{c} sk, pk \leftarrow \mathsf{Setup}(1) \\ cvk, R_{\mathsf{dec}} \leftarrow \mathsf{QVKeyGen}(sk) \\ R_{\mathsf{cert}} \leftarrow \mathsf{Cert}(R_{\mathsf{dec}}) \end{array}\right] = 1.$$

Lessor security: *Consider the following game played by the challenger and an adversary.*

1. *The challenger runs* $sk, pk \leftarrow \mathsf{PKE.Setup}(1)$*, and then*

$$cvk, R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QVKeyGen}(sk).$$

2. *The adversary* $\mathcal{A}_1$ *gets access to* R_{dec}, cvk *and* pk*, it produces a certificate* R_{cert}*, a state register* R *and two challenge messages* m_0, m_1*.*

3. *The challenger runs* $vb \leftarrow \mathsf{PKE.Verify}(cvk, R_{\mathsf{cert}})$*. If* vb *is* INVALID*, challenger outputs* 0 *and terminates.*

4. *The challenger samples a challenge bit* $b \leftarrow \{0,1\}$ *and prepares the ciphertext* $ct \leftarrow \mathsf{PKE.Enc}(pk, m_b)$*.*

5. $\mathcal{A}_2$ *gets* R, m_0, m_1, cvk, pk *and* ct*, it outputs a prediction* b'*.*

6. *The challenger outputs* 1 *if* $b' = b$*.*

We say that the adversary has won the game if the challenger outputs 1 *and we require that any QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *wins the game with probability at most*

$$\frac{\Pr[vb = \mathsf{VALID}]}{2} + \mathsf{negl}(\lambda).$$

Now, we move onto our construction of PKE with intrusion-detection.

Theorem 34. *Suppose there exists a public-key encryption scheme with publicly verifiable secure key leasing. Then, there exists a public-key encryption scheme with intrusion-detection.*

Proof. Let PKE' be a public-key encryption scheme with secure key leasing. We construct PKE as follows. Let $\mathsf{PKE.Setup}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec}$ be the same as those of PKE', and $\mathsf{PKE.QLKeyGen}$ be the same as $\mathsf{PKE}'.\mathsf{QVKeyGen}$. Consider the binary measurement implemented by the channel $\mathsf{PKE}'.\mathsf{Verify}(cvk, \mathsf{PKE}'.\mathsf{Cert}(\cdot))$, and let $\mathsf{PKE.TestIntrusion}$ be its rewinding version as obtained by Lemma 1. We associate NO INTRUSION with VALID and INTRUSION with INVALID.

By Lemma 1, $\mathsf{TestIntrusion}(tk, R)$ has the same output distribution as the procedure $\mathsf{PKE}'.\mathsf{Verify}(vk, \mathsf{PKE}'.\mathsf{Cert}(R))$ for any register R in any state. Therefore, by the verification correctness of PKE', when we initialize $tk, R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QLKeyGen}(sk)$, we have

$$\Pr[\mathsf{PKE.TestIntrusion}(tk, R_{\mathsf{dec}}) = \mathsf{NO\ INTRUSION}] = 1. \tag{3}$$

which shows that PKE satisfies detection correctness.

Now, again initialize $tk, R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QLKeyGen}(sk)$ and let ρ, ρ' denote the state of R_{dec} before and immediately after $\mathsf{PKE.TestIntrusion}(tk, R_{\mathsf{dec}})$, respectively. Then, by Equation (3) and Lemma 1, we get

$$\left\|\rho - \rho'\right\|_1 \leq \mathsf{negl}(\lambda),$$

which shows that PKE satisfies reusability.

Finally, we argue intrusion-detection security. For a contradiction, suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ such that $\mathcal{A}$ wins the intrusion-detection game with probability

$$p_{detection} = \frac{\Pr[E_{\mathsf{INTRUSION}}]}{2} + \frac{1}{p(\lambda)}$$

for some polynomial $p(\cdot)$ and infinitely many values of λ where we let $E_{\mathsf{INTRUSION}}$ be the event that the output of $\mathsf{TestIntrusion}$ is $\mathsf{INTRUSION}$. Then, we construct the following adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ against the secure key leasing game for PKE'.

$\underline{\mathcal{A}'_1(R_{\mathsf{dec}}, cvk, pk)}$

1. Run $R_{\mathsf{adv}}, m_0, m_1 \leftarrow \mathcal{A}_{\mathsf{intr}}(R_{\mathsf{dec}}, cvk, pk)$.[17]
2. Run $R_{\mathsf{cert}} \leftarrow \mathsf{PKE}'.\mathsf{Cert}(R_{\mathsf{dec}})$.
3. Output $R_{\mathsf{cert}}, R_{\mathsf{adv}}, m_0, m_1$.

$\underline{\mathcal{A}'_2(R, m_0, m_1, cvk, pk, ct)}$

1. Run $b \leftarrow \mathcal{A}_{\mathsf{Main}}(R, m_0, m_1, cvk, pk, ct)$.
2. Output b.

Consider the secure key leasing game played by $\mathcal{A}'$ and the intrusion-detection game played by $\mathcal{A}$. Observe that they are exactly the same, except for the following differences. We rewind $\mathsf{TestIntrusion}$ in the intrusion detection game, while in the key leasing game, the challenger does not apply the same rewinding. However, crucially, in the intrusion-detection game, we are effectively tracing out R_{dec} after testing for leakage, since we never use it again. Similarly, in the key leasing game, we trace out any garbage left from testing the certificate or running the verification, and we trace out the certificate itself too (once the verification succeeds), since $\mathcal{A}_2{}'$ does not use it. Since applying a channel to a subsystem and then tracing out the resulting subsystem is equivalent to tracing out without applying the channel, conditioned on the game not terminating early on[18], we can see that the leftover state of R_{adv} produced by $\mathcal{A}$ in the intrusion-detection game is the same as the leftover state used to invoke $\mathcal{A}_{\mathsf{Main}}$ in the key leasing game played by $\mathcal{A}'$. Hence, again conditioned on the game not terminating early, we conclude that both games have the same output distribution. Finally, since we have already observed that $\mathsf{TestIntrusion}(tk, R)$ has the same output distribution as $\mathsf{PKE}'.\mathsf{Verify}(cvk, \mathsf{PKE}'.\mathsf{Cert}(R))$ for any register R, we see that the probability of terminating early on is the same for both games. Hence, combined with the previous part, we get that the probability of $\mathcal{A}'$ winning the key leasing game, $p_{leasing}$, is the same as the probability of $\mathcal{A}$ winning the intrusion-detection game. Let E_{INVALID} be the event that the output of Verify in the secure key leasing game played by $\mathcal{A}'$ is $\mathsf{INVALID}$. Then,

$$\begin{aligned} p_{leasing} &= \frac{\Pr[E_{\mathsf{INTRUSION}}]}{2} + \frac{1}{p(\lambda)} \\ &= \frac{\Pr[E_{\mathsf{INVALID}}]}{2} + \frac{1}{p(\lambda)}. \end{aligned}$$

[17] Note that the state of R_{dec} has been altered by applying $\mathcal{A}_{\mathsf{intr}}$, and references to this register afterwards are with regards to its updated state.

[18] The game terminates early on when $\mathsf{TestIntrusion}$ outputs $\mathsf{INTRUSION}$ in the intrusion-detection game or when Verify outputs $\mathsf{INVALID}$ in the key leasing game.

This violates the key leasing security of PKE', which is a contradiction. □

Theorem 35. *Suppose there exists a public-key encryption scheme with intrusion-detection. Then, it is also a public-key encryption scheme with publicly verifiable secure key leasing with quantum certificates.*

Proof. We define $\mathsf{PKE.Cert}(R_{\mathsf{dec}})$ to output R_{dec} and we define $\mathsf{PKE.Verify}$ to be the same as $\mathsf{PKE}'.\mathsf{TestIntrusion}$. Observe that if an adversary can break the secure key leasing, then the same adversary can also break the intrusion-detection game by producing a fake certificate and placing it instead of the key register. □

We remark that, as discussed in Section 5.3, assuming the existence of the quantum hardness of LWE, and post-quantum sub-exponentially secure iO and OWFs, one can construct PKE schemes with unclonable keys, which is a stronger security notion than PKE with publicly-verifiable key leasing. Further, [BGG+23] also construct functional encrpytion with publicly verifiable key leasing (which implies public-key encryption with the same property), based on indistinguishability obfuscation and injective one- way functions. While the post-quantum iO is a strong assumption, we show that intrusion detection implies public-key quantum money, whose only known construction in the plain model relies is also based on post-quantum iO. We also note that a recent work [AKN+23] also builds PKE with key leasing from any post-quantum PKE. However, this construction lacks public-verifiability, which is crucially needed for intrusion-detection since the leakage adversary gets access to the complete state of the honest party, which includes the intrusion-detection key.

Theorem 36. *Suppose there exists a public-key encryption scheme with intrusion-detection. Then, there exists a public-key quantum money scheme.*

Proof. We will first construct a quantum money scheme that is only $1/2 + \mathsf{negl}(\lambda)$ secure but has perfect correctness. We can achieve negligible security by parallel amplification: A banknote consists of λ banknotes from λ independent instances of the weak scheme and it is verified to be valid if all the smaller banknotes are valid.

Let PKE be a public-key encryption scheme with intrusion-detection. Then, let PKE' be the scheme where we modify the intrusion detection algorithm to be the following.

$\underline{\mathsf{PKE}'.\mathsf{TestIntrusion}(tk, R_{\mathsf{dec}})}$

1. Sample $m_0, m_1 \leftarrow \mathcal{M}$.
2. Sample $b \leftarrow \{0,1\}$.
3. Sample $ct \leftarrow \mathsf{PKE.Enc}(pk, m_b)$.
4. Sample $m' \leftarrow \mathsf{PKE.Dec}(R_{\mathsf{dec}}, ct)$.
5. Sample $tb \leftarrow \mathsf{PKE.TestIntrusion}(tk, R_{\mathsf{dec}})$.
6. Output 1 if and only $m' = m_b$ and $tb = \mathsf{NO\ INTRUSION}$.

One caveat is that the algorithm above uses pk. However, without loss of generality, we can assume that the secret key includes pk, and therefore we can include pk in tk too.

We now argue that PKE' still satisfies intrusion detection correctness: The decryption algorithm on an honest key only negligibly disturbs it, therefore, the steps before running $\mathsf{PKE.TestIntrusion}$ do not disturb the key.

Now, we argue that PKE' also satisfies intrusion detection security. Suppose for a contradiction that an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ wins the intrusion detection security game against PKE' with non-negligible advantage. Then we can construct $\mathcal{A}' = (\mathcal{A}'_{\mathsf{Main}}, \mathcal{A}'_{\mathsf{intr}})$ for PKE as follows. $\mathcal{A}'_{\mathsf{intr}}$ first runs $\mathcal{A}_{\mathsf{intr}}$ on the keys to obtain the updated register R_{dec}, then it applies the first four steps of $\mathsf{PKE}'.\mathsf{TestIntrusion}$. If $b' = b$, $\mathcal{A}'_{\mathsf{intr}}$ outputs $\bot$ as the leftover register. Now observe that $\mathsf{PKE}.\mathsf{TestIntrusion}$ run on the updated register corresponds exactly to running $\mathsf{PKE}'.\mathsf{TestIntrusion}$ in the intrusion detection security game played for PKE'. Therefore, we conclude the security of PKE'.

Finally, now we move onto the weak quantum money scheme. Note that our scheme will be a mini-scheme, where there is no secret key of the bank and there is only one banknote given to the adversary. A mini-scheme can be updated to a full-fledged quantum money scheme using digital signatures [AC12].

$\underline{\mathsf{QM.Gen}(1^\lambda)}$

1. $pk, sk \leftarrow \mathsf{PKE}'.\mathsf{Setup}(1^\lambda)$.
2. $tk, R_{\mathsf{dec}} \leftarrow \mathsf{PKE}'.\mathsf{QKeyGen}(sk)$.
3. Output tk, R_{dec}.

$\underline{\mathsf{QM.Verify}(tk, R_{\mathsf{dec}})}$

1. Output $\mathsf{PKE}'.\mathsf{TestIntrusion}(tk, R_{\mathsf{dec}})$.

Now we argue security of the scheme. Suppose there exists an adversary $\mathcal{A}$ such that upon receiving tk, R_{dec}, it outputs two registers R_1, R_2 that both pass $\mathsf{PKE}'.\mathsf{Verify}(tk, R_1)$ and $\mathsf{PKE}'.\mathsf{Verify}(tk, R_2)$ simultaneously with probability $1/2 + 1/\mathrm{poly}(\lambda)$. We construct an adversary $\mathcal{A}' = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ for PKE' as follows. It runs $\mathcal{A}$ on the keys to obtain R_1, R_2. It outputs R_1 as the updated key register, and R_2 as its leakage register. It also outputs two random messages m_0, m_1 as its challenge messages. Now, observe that the intrusion detection game runs $\mathsf{PKE}'.\mathsf{TestIntrusion}$ on R_1, and a decryption test on R_2, which corresponds to the first part of $\mathsf{PKE}'.\mathsf{TestIntrusion}$. Therefore, both tests succeed simultaneously with probability $1/2 + 1/\mathrm{poly}(\lambda)$ as desired. □

8.2 Digital Signature Schemes with Intrusion-Detection

We show that digital signatures with intrusion-detection is equivalent to digital signature schemes with key leasing, using essentially the same technique as used for PKEs in Section 8.1. Note that our results hold only for uniformly sampled challenge messages. This is necessary with classical signatures, as in the case of unclonable digital signature keys. In the intrusion detection game with a selective challenge message, an adversary can sign a message of its choice (which only negligibly disturbs the key) and leak the signature.

Definition 30 (Digital signatures with intrusion-detection)**.** *A digital signature scheme with intrusion-detection is a digital signature scheme with the following additional algorithms that satisfies the reusability and security guarantees below.*

- $\mathsf{QLKeyGen}(sk)$: *Along with a quantum signing key R_{sign}, also outputs a classical intrusion-detection key tk.*
- $\mathsf{TestIntrusion}(tk, R_{sign})$: *Takes the intrusion-detection key and the signing key, outputs* INTRUSION *if* leakage is detected*,* NO INTRUSION *otherwise.*

Digital signature correctness and security *We require the usual correctness and security satisfied for the signing key generated by* QKeyGen *to now also hold for the signing key generated by* QLKeyGen*.*

Detection correctness

$$\Pr\left[\mathsf{TestIntrusion}(tk, R_{\mathsf{sign}}) = \text{NO INTRUSION} : \begin{array}{c} sk, vk \leftarrow \mathsf{Setup}(1) \\ tk, R_{\mathsf{sign}} \leftarrow \mathsf{QLKeyGen}(sk) \end{array}\right] = 1.$$

Reusability after testing *Initialize the signing register,* $R_{\mathsf{sign}}, tk \leftarrow \mathsf{QLKeyGen}(sk)$ *and let* ρ *denote its state. Run the algorithm* TestIntrusion *on* tk *and* R_{sign}*, and let* ρ' *denote the state of the register* R_{sign} *immediately afterwards. Then,*

$$\left\|\rho - \rho'\right\|_1 \leq \mathsf{negl}(\lambda).$$

We note that reusability after testing will follow from detection correctness by utilizing Lemma 1.

Intrusion-detection security *Consider the following game between the challenger and an adversary. We say that the adversary has won the game if the challenger outputs* 1 *and we require that any QPT adversary* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *wins the following game with probability at most* $\mathsf{negl}(\lambda)$*.*

1. *The challenger runs* $sk, vk \leftarrow \mathsf{DS.Setup}(1)$*, and then* $tk, R_{\mathsf{sign}} \leftarrow \mathsf{DS.QLKeyGen}(sk)$*.*
2. *The adversary* $\mathcal{A}_{\mathsf{intr}}$ *gets access to* R_{sign}, tk *and* vk *and it produces a leakage register* R_{adv}*, along with the updated register* R_{sign}*.*
3. *The challenger runs* $tb \leftarrow \mathsf{DS.TestIntrusion}(tk, R_{\mathsf{sign}})$*. If* tb *is* INTRUSION*, challenger outputs* 0 *and terminates.*
4. *The challenger samples a challenge message* $m \leftarrow \mathcal{M}$*.*
5. $\mathcal{A}_{\mathsf{Main}}$ *gets* R_{adv}, tk, vk *and* m*, it outputs a forged signature* s*.*
6. *The challenger outputs* 1 *if* $\mathsf{DS.Verify}(m, s) = 1$*.*

Similar to other primitives with secure leasing, we formally define signature schemes with secure key leasing.

Definition 31 (Digital signatures with secure key leasing)**.** *A digital signature scheme with secure key leasing is a digital signature scheme with the following additional algorithms that satisfies the correctness and security guarantees below.*

- $\mathsf{QVKeyGen}(sk)$: *Along with a quantum signing key* R_{sign}*, also outputs a classical deletion verification key* cvk*.*
- $\mathsf{Cert}(R_{sign})$: *Takes the signing key and outputs a deletion certificate.*
- $\mathsf{VerifyDeletion}(cvk, R_{\mathsf{cert}})$: *Takes the verification key and a deletion certificate, outputs* VALID *if it is a valid certificate.*

Digital signature correctness and security *We require the usual correctness and security to now also hold for keys generated by* QVKeyGen*.*

Verification correctness

$$\Pr\left[\mathsf{VerifyDeletion}(cvk, R_{\mathsf{cert}}) = \mathsf{VALID} : \begin{array}{c} sk, vk \leftarrow \mathsf{Setup}(1) \\ cvk, R_{\mathsf{sign}} \leftarrow \mathsf{QVKeyGen}(sk) \\ R_{\mathsf{cert}} \leftarrow \mathsf{Cert}(R_{\mathsf{dec}}) \end{array}\right] = 1.$$

Lessor security *Consider the following game between the challenger and an adversary. We say that the adversary has won the game if the challenger outputs* 1 *and we require that any QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *wins the following game with probability at most* $\mathsf{negl}(\lambda)$.

1. *The challenger runs* $sk, vk \leftarrow \mathsf{DS.Setup}(1)$, *and then*

$$cvk, R_{\mathsf{sign}} \leftarrow \mathsf{DS.QVKeyGen}(sk).$$

2. *The adversary* $\mathcal{A}_1$ *gets access to* R_{sign}, cvk *and* vk, *it produces a certificate* R_{cert} *and a state register* R.

3. *The challenger runs* $vb \leftarrow \mathsf{VerifyDeletion}(cvk, R_{\mathsf{cert}})$. *If* vb *is* INVALID, *challenger outputs* 0 *and terminates.*

4. *The challenger samples a challenge message* $m \leftarrow \mathcal{M}$.

5. $\mathcal{A}_2$ *gets* R, cvk, vk *and* m, *it outputs a forged signature* s.

6. *The challenger outputs* 1 *if* $\mathsf{DS.Ver}(m, s) = 1$.

Theorem 37. *Suppose there exists a digital signature scheme with publicly verifiable secure key leasing. Then, there exists a digital signature scheme with intrusion-detection.*

Proof. The construction and the reduction are essentially the same as Theorem 34. Let DS' be a public-key encryption scheme with secure key leasing. We construct DS as follows. Let $\mathsf{DS.Setup}, \mathsf{DS.Sign}, \mathsf{DS.Ver}$ be the same as those of DS', and let DS.QLKeyGen be the same as $\mathsf{DS}'.\mathsf{QVKeyGen}$. Consider the binary measurement implemented by $\mathsf{DS}'.\mathsf{Verify}(cvk, \mathsf{DS}'.\mathsf{Cert}(\cdot))$, and let DS.TestIntrusion be its rewinding version as obtained by Lemma 1. We associate NO INTRUSION with VALID and INTRUSION with INVALID.

Same arguments as in the proof of Theorem 34 show the detection correctness and reusability of DS.

Finally, we argue intrusion-detection security. For a contradiction, suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ such that $\mathcal{A}$ wins the intrusion-detection game with probability $\frac{1}{p(\lambda)}$ for some polynomial $p(\cdot)$ and infinitely many values of λ. Then, we construct the following adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ against the secure key leasing game for DS'.

$\underline{\mathcal{A}'_1(R_{\mathsf{sign}}, cvk, vk)}$

1. Run $R_{\mathsf{adv}} \leftarrow \mathcal{A}_{\mathsf{intr}}(R_{\mathsf{sign}}, cvk, vk)$.
2. Run $R_{\mathsf{cert}} \leftarrow \mathsf{DS}'.\mathsf{Cert}(R_{\mathsf{sign}})$.
3. Output $R_{\mathsf{cert}}, R_{\mathsf{adv}}$.

$\underline{\mathcal{A}'_2(R, cvk, vk, m)}$

1. Run $b \leftarrow \mathcal{A}_{\mathsf{Main}}(R, cvk, vk, m)$.
2. Output b.

The same arguments as in the proof of Theorem 34 show that $\mathcal{A}'$ wins the key leasing game for DS' with probability $\frac{1}{p(\lambda)}$, which is a contradiction. □

Theorem 38. *Suppose there exists a digital signature scheme with intrusion-detection. Then, it is also a digital signature scheme with publicly verifiable secure key leasing with quantum certificates.*

Proof. We define $\mathsf{DS.Cert}(R_{\mathsf{dec}})$ to output R_{dec}. We define $\mathsf{DS.Verify}$ to be the same as $\mathsf{DS}'.\mathsf{TestIntrusion}$. Observe that if an adversary can break the secure key leasing, then the same adversary can also break the intrusion-detection game by producing a fake certificate and placing it instead of the key register. □

As discussed in Section 6.2, assuming post-quantum subexponentially secure indistinguishability obfuscation, OWFs, and qLWE, one can construct digital signature schemes with unclonable signing keys, which is a stronger notion that digital signatures with secure key leasing.

Similar to the case of public-key encryption, digital signatures with intrusion detection implies public key quantum money.

Theorem 39. *Suppose there exists a digital signature scheme with intrusion-detection. Then, there exists a public-key quantum money scheme.*

Proof. Let DS be a public-key encryption scheme with intrusion-detection. Then, let DS' be the scheme where we modify the intrusion detection algorithm to be the following.

$\underline{\mathsf{DS}'.\mathsf{TestIntrusion}(tk, R_{\mathsf{sign}})}$

1. Sample $m \leftarrow \mathcal{M}$.
2. Sample $sg \leftarrow \mathsf{DS.Sign}(R_{\mathsf{sign}}, m)$.
3. Sample $tb \leftarrow \mathsf{DS.TestIntrusion}(tk, R_{\mathsf{sign}})$.
4. Output 1 if and only $\mathsf{DS.Ver}(vk, m, sg) = 1$ and $tb = \mathsf{NO\ INTRUSION}$.

Similar to the case of public-key encryption, we can show that DS' still satisfies intrusion detection correctness and security.

Finally, now we move onto the mini quantum money scheme. A mini-scheme can be updated to a full-fledged quantum money scheme using digital signatures [AC12].

$\underline{\mathsf{QM.Gen}(1^\lambda)}$

1. $vk, sk \leftarrow \mathsf{DS}'.\mathsf{Setup}(1^\lambda)$.
2. $tk, R_{\mathsf{dec}} \leftarrow \mathsf{DS}'.\mathsf{QKeyGen}(sk)$.
3. Output tk, R_{dec}.

$\underline{\mathsf{QM.Verify}(tk, R_{\mathsf{dec}})}$

1. Output $\mathsf{DS'.TestIntrusion}(tk, R_{\mathsf{dec}})$.

Similar to Theorem 36, we can show that the scheme above is secure. □

8.3 Functional Encryption with Intrusion-Detection

In this section, we introduce the notion of intrusion-detection for functional keys of public-key functional encryption schemes, and show that it is equivalent to functional encryption schemes with publicly verifiable certified key deletion [BGG+23]. We will have schemes for classical messages and families of classical functions, with classical public-key and quantum functional keys.

Definition 32 (Functional encryption with intrusion-detection)**.** *A functional encryption scheme with intrusion detection for a family of functions $\mathcal{F}$ is a public-key functional encryption scheme (Definition 7) for $\mathcal{F}$ with the following additional algorithms that satisfy the correctness and security guarantees below.*

- $\mathsf{QLKeyGen}(msk, f)$: *Along with a quantum functional key R_f, outputs a classical intrusion detection key tk.*
- $\mathsf{TestIntrusion}(tk, R_f)$*: Takes the intrusion detection key and functional key, outputs* INTRUSION *if leakage is detected,* NO INTRUSION *otherwise.*

FE correctness and security: *We require the usual functional encryption correctness and security (Definition 7) for keys generated by* QLKeyGen*.*

Detection correctness: *For all $f \in \mathcal{F}$,*

$$\Pr\left[\mathsf{TestIntrusion}(tk, R_f) = \text{NO INTRUSION} : \begin{array}{c} pk, msk \leftarrow \mathsf{Setup}(1) \\ tk, R_f \leftarrow \mathsf{QLKeyGen}(msk, f) \end{array}\right] = 1.$$

Reusability after testing: *We require the following for all $f \in \mathcal{F}$. Initialize the functional key register, $tk, R_f \leftarrow \mathsf{QLKeyGen}(msk, f)$ and let ρ denote its state. Run the algorithm* TestIntrusion *on tk and R_f, and let ρ' denote the state of the register R_f immediately afterwards. Then,*

$$\left\|\rho - \rho'\right\|_1 \leq \mathsf{negl}(\lambda).$$

We note that reusability after testing will follow from detection correctness by utilizing Lemma 1.

Intrusion detection security: *For any polynomial $p(\cdot)$ and any functions $f_1, \ldots, f_{p(\lambda)} \in \mathcal{F}$, for any (stateful) QPT adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$, the advantage of $\mathcal{A}$ in the following game is negligible.*

1. *$\mathcal{A}_{\mathsf{intr}}$ outputs two messages m_0, m_1.*
2. *The challenger runs $pk, msk \leftarrow \mathsf{FE.Setup}(1)$ and then for all $i \in [p(\lambda)]$,*

$$tk, R_{f_i} \leftarrow \mathsf{FE.QLKeyGen}(msk, f_i).$$

3. *The leakage adversary $\mathcal{A}_{\mathsf{intr}}$ gets access to $(R_{f_i})_{i\in[p(\lambda)]}, (f_i)_{i\in[p(\lambda)]}, tk$ and pk, it produces a leakage register R_{adv} along with the* updated *registers $(R_i)_{i\in[p(\lambda)]}$.*

4. *The challenger sets* $tb = 0$ *and runs the following for each* $i \in [p(\lambda)]$.
   (a) $tb_i \leftarrow \mathsf{TestIntrusion}(tk, R_{f_i})$.
   (b) *If* $tb_i = \mathsf{INTRUSION}$ *and* $f_i(m_0) \neq f_i(m_1)$, *set* $tb = 1$.
5. *If* $tb = 1$, *challenger outputs* 0 *and terminates.*
6. *The challenger samples a challenge bit* $b \leftarrow \{0,1\}$ *and prepares the ciphertext* $ct \leftarrow \mathsf{FE.Enc}(pk, m_b)$.
7. $\mathcal{A}_{\mathsf{Main}}$ *gets* $R_{\mathsf{adv}}, (f_i)_{i\in[p(\lambda)]}, m_0, m_1, tk, pk$ *and* ct, *it outputs a prediction* b'.
8. *The challenger outputs* 1 *if* $b' = b$.

Let $\mathsf{Game}_{\mathcal{A}}$ *denote the output of the above experiment. We define the advantage of* $\mathcal{A}$ *to be* $\left|\Pr[\mathsf{Game}_{\mathcal{A}} = 1] - \frac{\Pr[tb=1]}{2}\right|$.

Prior work of [BGG+23] define and construct schemes for the related model of functional encryption with certified key deletion.

Definition 33 (Functional encryption with certified key deletion [BGG+23])**.** *A functional encryption scheme with certified key deletion for a family of functions* $\mathcal{F}$ *is a public-key functional encryption scheme for* $\mathcal{F}$ *with the following additional algorithms that satisfy the correctness and security guarantees below.*

- $\mathsf{QVKeyGen}(msk, f)$: *Along with a quantum functional key* R_f, *outputs a classical verification key* vk.
- $\mathsf{Cert}(R_f)$: *Takes the functional key and outputs a classical deletion certificate.*
- $\mathsf{Verify}(vk, cert)$: *Takes the verification key and a deletion certificate, outputs* VALID *if it is a valid certificate.*

FE correctness and security: *We require the usual functional encryption correctness and security (Definition 7) for keys generated by* QVKeyGen.

Verification correctness: *For all* $f \in \mathcal{F}$,

$$\Pr\left[\mathsf{Verify}(vk, cert) = \mathsf{VALID} : \begin{array}{c} pk, msk \leftarrow \mathsf{Setup}(1) \\ vk, R_f \leftarrow \mathsf{QVKeyGen}(msk, f) \\ cert \leftarrow \mathsf{Cert}(R_f) \end{array}\right] = 1.$$

Certified deletion security: *For any (stateful) QPT adversary* $\mathcal{A}$, *the advantage of* $\mathcal{A}$ *in the following game is negligible.*

1. $\mathcal{A}$ *outputs two messages* m_0, m_1.
2. *The challenger runs* $pk, msk \leftarrow \mathsf{FE.Setup}(1)$ *and sends* pk *to the adversary.*
3. *For* $p(\lambda)$ *many times for some polynomial* $p(\cdot)$, $\mathcal{A}$ *adaptively submits a query* $f_i \in \mathcal{F}$ *and receives* $R_{f_i}, vk_i \leftarrow \mathsf{FE.QVKeyGen}(msk, f_i)$.
4. *The adversary sends a list of deletion proofs* $cert_1, \ldots, cert_{p(\lambda)}$.

5. *The challenger sets* $vb = 0$ *and runs the following for each* $i \in [p(\lambda)]$.
   (a) $vb_i \leftarrow \mathsf{Verify}(vk, cert_i)$.
   (b) *If* $vb_i = \mathsf{INVALID}$ *and* $f_i(m_0) \neq f_i(m_1)$, *set* $vb = 1$.
6. *If* $vb = 1$, *challenger outputs* 0 *and terminates.*
7. *The challenger samples a challenge bit* $b \leftarrow \{0,1\}$ *and prepares the ciphertext* $ct \leftarrow \mathsf{FE.Enc}(pk, m_b)$.
8. *The adversary receives* ct *and for polynomially many times,* $\mathcal{A}$ *adaptively submits a query* $f \in \mathcal{F}$. *For each query* f, *if* $f(m_0) = f(m_1)$, *challenger samples* $R_f, vk \leftarrow \mathsf{FE.QVKeyGen}(msk, f)$ *and sends* R_f, vk *to the adversary.*
9. *Adversary outputs a guess* b'.
10. *Output* 1 *if* $b' = b$.

Let $\mathsf{Game}_{\mathcal{A}}$ *denote the output of the above experiment. We define the advantage of* $\mathcal{A}$ *to be* $\left|\Pr[\mathsf{Game}_{\mathcal{A}} = 1] - \frac{\Pr[vb=1]}{2}\right|$.

Theorem 40 ([BGG+23])**.** *Assuming post-quantum indistinguishability obfuscation, public key encryption, and injective one-way functions, there exists a functional encryption scheme with selective secret key publicly verifiable certified deletion with classical certificates.*

We show that publicly verifiable certified key deletion implies intrusion-detection, similar to the previous primitives.

Theorem 41. *Suppose there exists a functional encryption scheme for a family of functions* $\mathcal{F}$ *with publicly verifiable certified key deletion. Then, there exists a functional encryption scheme for* $\mathcal{F}$ *with intrusion detection.*

Proof. Construction and the security proof are mostly the same as Theorem 34, so we only sketch it. Let FE' be a functional encryption scheme with certified deletion for $\mathcal{F}$ as in the theorem statement. We construct a functional encryption scheme FE for $\mathcal{F}$ with intrusion detection as follows. Define $\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec}$ to be the same as those of FE', and define $\mathsf{FE.QLKeyGen}$ to be the same as $\mathsf{FE}'.\mathsf{KeyGen}$. Consider the binary measurement implemented by $\mathsf{FE}'.\mathsf{Verify}(cvk, \mathsf{FE}'.\mathsf{Cert}(\cdot))$, and let $\mathsf{FE.TestIntrusion}$ be its rewinding version as obtained by Lemma 1. We associate $\mathsf{NO\ INTRUSION}$ with VALID and $\mathsf{INTRUSION}$ with $\mathsf{INVALID}$.

It is easy to see that usual FE correctness and security, along with detection correctness and reusability after testing are satisfied by FE. Finally, we argue intrusion detection security. Suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ that wins intrusion detection game with non-negligible advantage. Then, we define an adversary $\mathcal{A}'$ for the certified deletion game as follows. $\mathcal{A}'$ runs $\mathcal{A}_{\mathsf{intr}}$ to obtain m_0, m_1 and outputs them. Then, it asks for keys for $f_1, \ldots, f_{p(\lambda)}$, and it runs $\mathcal{A}_{\mathsf{intr}}$ on these keys. It runs Cert on all of the updated functional key registers, outputs the resulting certificates and keeps R_{adv} as its state. In the second query stage, it does not make any queries. Finally, when it receives the challenge ciphertext, it runs $\mathcal{A}_{\mathsf{Main}}$ on the ciphertext and R_{adv}. An argument similar to Theorem 34 shows that $\mathcal{A}'$ wins the certified deletion game with non-negligible advantage. Crucially note that we trace out the certificates in the certified deletion game, and the after-the-leakage states of the functional keys in the intrusion detection game. Hence, the rewinding of $\mathsf{TestIntrusion}$ has no effect. □

Theorem 42. *Suppose there exists exists a functional encryption scheme for a family of functions $\mathcal{F}$ with intrusion detection. Then, it is also a functional encryption scheme for $\mathcal{F}$ with publicly verifiable certified key deletion with quantum certificates.*

Proof. We define $\mathsf{FE.Cert}(R_{\mathsf{dec}})$ to output R_{dec}. We define $\mathsf{FE.Verify}$ to be the same as $\mathsf{FE}'.\mathsf{TestIntrusion}$. Observe that if an adversary can break the secure key leasing, then the same adversary can also break the intrusion-detection game by producing a fake certificate and placing it instead of the key register. □

Corollary 8. *Assuming post-quantum indistinguishability obfuscation, public key encryption, and injective one-way functions, there exists a functional encryption scheme with intrusion detection.*

Remark 9. *It is easy to see that when the underlying functional encryption scheme has adaptive-function*[19] *security for certified deletion, the intrusion detection scheme constructed in Theorem 41 will have intrusion detection security even when the functional keys possessed by the honest party are for functions (adaptively) chosen by the adversary. Similarly, adaptive-message security of the certified deletion scheme would imply adaptive-message intrusion detection security.*

Since public-key encryption with intrusion-detection implies public-key quantum money, so does functional encryption.

Theorem 43. *Suppose there exists a public-key functional encryption scheme with intrusion-detection. Then, there exists a public-key quantum money scheme.*

8.4 Indistinguishability Obfuscation with Intrusion-Detection

In this section, we define intrusion-detection for differing-inputs obfuscation with intrusion-detection and show that it is equivalent to obfuscation with certified deletion [BGG+23].

Definition 34 (Differing-inputs circuit family [BGG+23])**.** *Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ be a family of circuits and let $\mathcal{D}$ be an efficiently sampleable ensemble associated with $\mathcal{C}$. We say that $(\mathcal{C}, \mathcal{D})$ is a* differing-inputs circuit family *if for every QPT adversary $\mathcal{A}$, we have*

$$\Pr\left[C_0(x) \neq C_1(x) : \begin{array}{c}(C_0, C_1, aux) \leftarrow \mathcal{D} \\ x \leftarrow \mathcal{A}(C_0, C_1, aux)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

If C_0, C_1 differ on at most polynomially many inputs for all C_0, C_1 in the support of $\mathcal{D}$, we say that $(\mathcal{C}, \mathcal{D})$ is a differing-inputs circuit family with polynomially many differing inputs*.*

Definition 35 (Differing-inputs obfuscation [BCP14, BGG+23])**.** *Let $(\mathcal{C}, \mathcal{D})$ be a differing-inputs circuit family. An obfuscation scheme $i\mathcal{O}$ for $(\mathcal{C}, \mathcal{D})$ consists of the following algorithms satisfying the correctness and security guarantees below.*

- *$i\mathcal{O}.\mathsf{Gen}(C)$: Takes a circuit C and outputs an (possibly quantum) obfuscation of C.*
- *$i\mathcal{O}.\mathsf{Eval}(R_{\mathsf{obf}}, x)$: Takes an obfuscated program and evaluates it on x.*

Functionality preservation: *For all $C \in \mathcal{C}$ and all inputs x,*

$$\Pr[i\mathcal{O}.\mathsf{Eval}(R_{\mathsf{obf}}, x) = C(x) : R_{\mathsf{obf}} \leftarrow i\mathcal{O}.\mathsf{Gen}(C)] = 1.$$

[19] Functions adaptively chosen by the adversary by interacting with the functional key generator

Obfuscation security *For any QPT adversary $\mathcal{A}$,*

$$\Pr\left[b' = b : \begin{array}{c}(C_0, C_1, aux) \leftarrow \mathcal{D} \\ b \leftarrow \{0,1\} \\ R_{\mathsf{obf}} \leftarrow i\mathcal{O}.\mathsf{Gen}(C_b) \\ b' \leftarrow \mathcal{A}(C_0, C_1, aux, R_{\mathsf{obf}})\end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Definition 36 (Differing-inputs obfuscation with intrusion-detection)**.** *Let $(\mathcal{C}, \mathcal{D})$ be a differing-inputs circuit family. An obfuscation scheme with intrusion detection for $(\mathcal{C}, \mathcal{D})$ is an obfuscation scheme for $(\mathcal{C}, \mathcal{D})$ with the following additional algorithms that satisfy the security and correctness guarantees below.*

- $\mathsf{QLGen}(C)$*: Along with a quantum obfuscation of C, also outputs a intrusion detection key tk.*
- $\mathsf{TestIntrusion}(tk, R_{\mathsf{obf}})$*: Takes the intrusion detection key and the obfuscation, outputs* INTRUSION *if leakage is detected,* NO INTRUSION *otherwise.*

Obfuscation correctness and security *We require the usual correctness and security satisfied by the obfuscation scheme to now also hold for obfuscations generated by* QLGen*.*

Detection correctness *For all circuits C in the support of $\mathcal{D}$, we require that*

$$\Pr[\mathsf{TestIntrusion}(tk, R_{\mathsf{obf}}) = \text{NO INTRUSION} : tk, R_{\mathsf{obf}} \leftarrow \mathsf{QLGen}(C)] \geq 1 - \mathsf{negl}(\lambda).$$

Reusability after testing *For all circuits C in the support of $\mathcal{D}$, we require the following. Initialize the obfuscation register $R_{\mathsf{obf}} \leftarrow \mathsf{QLGen}(C)$, and let ρ denote its states. Run the algorithm* TestIntrusion *on R_{obf} and tk, and let ρ' denote the state of the register R_{obf} immediately afterwards. Then,*

$$\left\|\rho - \rho'\right\|_1 \leq \mathsf{negl}(\lambda).$$

We note that reusability after testing will follow from detection correctness by utilizing Lemma 1.

Intrusion detection security *Any QPT adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ has at most negligible advantage in the following game.*

1. *The challenger runs $C_0, C_1, aux \leftarrow \mathcal{D}$.*
2. *The challenger samples a challenge bit $b \leftarrow \{0,1\}$ and prepares the obfuscation $R_{\mathsf{obf}} \leftarrow i\mathcal{O}.\mathsf{QLGen}(\mathsf{C_b})$.*
3. *$\mathcal{A}_{\mathsf{intr}}$ gets access to $R_{\mathsf{obf}}, C_0, C_1, aux$ and tk, it produces a leakage R_{adv} along with the updated register R_{obf}.*
4. *The challenger runs $tb \leftarrow \mathsf{TestIntrusion}(tk, R_{\mathsf{obf}})$. If tb is* INTRUSION*, it outputs 0 and terminates.*
5. *$\mathcal{A}_{\mathsf{Main}}$ gets $R_{\mathsf{adv}}, C_0, C_1, aux$ and tk, it produces a guess b'.*
6. *Output 1 if $b' = b$.*

Let $\mathsf{Game}_{\mathcal{A}}$ *denote the output of the above experiment. We define the advantage of* $\mathcal{A}$ *to be* $\left|\Pr[\mathsf{Game}_{\mathcal{A}}=1]-\frac{\Pr[tb=\mathsf{NO\ INTRUSION}]}{2}\right|$.

[BGG+23] define differing-inputs obfuscation with certified deletion and construct primitives secure in this model.

Definition 37 (Differing-inputs obfuscation with certified deletion [BGG+23])**.** *Let* $(\mathcal{C},\mathcal{D})$ *be a differing-inputs circuit family. An obfuscation scheme with certified deletion for* $(\mathcal{C},\mathcal{D})$ *is an obfuscation scheme for* $(\mathcal{C},\mathcal{D})$ *with the following additional algorithms that satisfy the security and correctness guarantees below.*

- $\mathsf{QGen}(C)$*: Along with a quantum obfuscation of* C*, also outputs a verification detection key* vk*.*
- $\mathsf{Cert}(R_{\mathsf{obf}})$*: Takes the obfuscation and produces a deletion certificate.*
- $\mathsf{Verify}(vk,cert)$*: Takes the verification key and a deletion certificate, outputs* VALID *if the certificate is valid,* $\mathsf{INVALID}$ *otherwise.*

Obfuscation correctness and security *We require the usual correctness and security satisfied by the obfuscation scheme to now also hold for obfuscations generated by* QGen*.*

Deletion correctness *For all circuits* C *in the support of* $\mathcal{D}$*, we require that*

$$\Pr\left[\mathsf{Verify}(vk,cert)=\mathsf{VALID}:\begin{array}{c}vk,R_{\mathsf{obf}}\leftarrow\mathsf{QGen}(C)\\ cert\leftarrow\mathsf{Cert}(vk,R_{\mathsf{obf}})\end{array}\right]\geq 1-\mathsf{negl}(\lambda).$$

Deletion security *Any QPT adversary* $\mathcal{A}$ *has at most negligible advantage in the following game.*

1. *The challenger runs* $C_0,C_1,aux\leftarrow\mathcal{D}$*.*
2. *The challenger samples a challenge bit* $b\leftarrow\{0,1\}$ *and prepares the obfuscation* $R_{\mathsf{obf}}\leftarrow i\mathcal{O}.\mathsf{QGen}(\mathsf{C_b})$*.*
3. $\mathcal{A}$ *gets access to* $R_{\mathsf{obf}},C_0,C_1,aux$ *and* vk*, it produces a state* R *and a deletion certificate* $cert$*.*
4. *The challenger runs* $vb\leftarrow\mathsf{Verify}(vk,cert)$*. If* vb *is* $\mathsf{INVALID}$*, it outputs* 0 *and terminates.*
5. $\mathcal{A}$ *gets* R*, it produces a guess* b'*.*
6. *Output* 1 *if* $b'=b$*.*

Let $\mathsf{Game}_{\mathcal{A}}$ *denote the output of the above experiment. We define the advantage of* $\mathcal{A}$ *to be* $\left|\Pr[\mathsf{Game}_{\mathcal{A}}=1]-\frac{\Pr[vb=\mathsf{VALID}]}{2}\right|$.

Theorem 44 ([BGG+23])**.** *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists a differing inputs obfuscation scheme with (publicly-verifiable) certified deletion for polynomially many differing inputs.*

Similar to the previous primitives, we show that certified deletion implies intrusion-detection.

Theorem 45. *Let $(\mathcal{C}, \mathcal{D})$ be a differing-inputs circuit family. Suppose there exists an obfuscation scheme with publicly verifiable certified deletion for $(\mathcal{C}, \mathcal{D})$. Then, there exists an obfuscation scheme with intrusion detection for $(\mathcal{C}, \mathcal{D})$.*

Proof. The construction and the proof are mostly the same as Theorem 34 and Theorem 41, so we only sketch it. Let $i\mathcal{O}'$ be an obfuscation scheme as in the theorem statement. We construct a intrusion-detection scheme $i\mathcal{O}$ for $(\mathcal{C}, \mathcal{D})$ as follows. Define $i\mathcal{O}.\mathsf{Eval}$ to be the same as $i\mathcal{O}'.\mathsf{Eval}$ and $i\mathcal{O}.\mathsf{QLGen}$ to be the same as $i\mathcal{O}'.\mathsf{QGen}$. Consider the binary measurement implemented by $i\mathcal{O}'.\mathsf{Verify}(cvk, i\mathcal{O}'.\mathsf{Cert}(\cdot))$, and let $i\mathcal{O}.\mathsf{TestIntrusion}$ be its rewinding version as obtained by Lemma 1. We associate NO INTRUSION with VALID and INTRUSION with INVALID.

It is easy to see that $i\mathcal{O}$ satisfies the obfuscation security and correctness, along with reusuability after testing and detection correctness. Finally, we argue intrusion detection security. Suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ that wins intrusion detection game with non-negligible advantage. Then, we define an adversary $\mathcal{A}'$ for the certified deletion game as follows. $\mathcal{A}'$ runs $\mathcal{A}_{\mathsf{intr}}$ on R_{obf} to obtain a leakage, then runs Cert on the updated register. It outputs the resulting certificate and keeps the leakage as its state. Finally, when it receives the challenge, it runs $\mathcal{A}_{\mathsf{Main}}$ on the challenge and R_{adv}. An argument similar to Theorem 34 and Theorem 41 shows that $\mathcal{A}'$ wins the certified deletion game with non-negligible advantage. □

Theorem 46. *Let $(\mathcal{C}, \mathcal{D})$ be a differing-inputs circuit family. Suppose there exists an obfuscation scheme with intrusion detection for $(\mathcal{C}, \mathcal{D})$. Then, it is also an obfuscation scheme with publicly verifiable certified deletion for $(\mathcal{C}, \mathcal{D})$.*

Proof. We define $i\mathcal{O}.\mathsf{Cert}(R_{\mathsf{dec}})$ to output R_{dec}. We define $i\mathcal{O}.\mathsf{Verify}$ to be the same as $i\mathcal{O}'.\mathsf{TestIntrusion}$. Observe that if an adversary can break the secure key leasing, then the same adversary can also break the intrusion-detection game by producing a fake certificate and placing it instead of the key register. □

Corollary 9. *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists a differing inputs obfuscation scheme with intrusion detection for polynomially many differing inputs.*

Similar to public-key encryption with intrusion detection, we can show that $i\mathcal{O}$ with intrusion-detection implies public-key quantum money.

Theorem 47. *Suppose there exists an indistinguishability obfuscation scheme with intrusion-detection. Then, there exists a public-key quantum money scheme.*

Proof. Follows from an argument similar to Theorem 36. □

8.5 Intrusion-Detection for Software

In this section, we introduce the notion of intrusion-detection for *software*[20], and show a construction of such schemes from any publicly verifiable, strong secure software leasing (SSL) scheme [AL21, KNY21, BGG+23] with only finite-term lessor security. This also gives the first natural use case for SSL schemes that only satisfy the weaker notion of finite-term lessor security, in which the lessee cannot keep the software forever and has to return it for the security guarantee to hold.

[20] Modeled as a sample from a distribution on a family of circuits.

Definition 38 (intrusion-detection for software)**.** *Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ be a family of classical circuits, where $C : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ for all $C \in \mathcal{C}_\lambda$, and let $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$ be an ensemble on $\mathcal{C}$. A β-perfect intrusion-detection scheme for $(\mathcal{C}, \mathcal{D})$ consists of the following QPT algorithms, with the correctness and security guarantees below.*

- $\mathsf{Gen}(C)$*: Takes a circuit C, outputs a intrusion-detection key and the protected version of C, a quantum state.*
- $\mathsf{Eval}(R_{\mathsf{prog}}, x)$ *: Takes the protected version of C and an input x, evaluates C on x.*
- $\mathsf{TestIntrusion}(tk, R_{\mathsf{prog}})$*: Takes the intrusion-detection key and the program register, outputs* INTRUSION *if leakage is detected,* NO INTRUSION *otherwise.*

Evaluation correctness *For all $C \in \mathcal{C}$,*

$$\Pr\Big[\forall x \in \{0,1\}^{n(\lambda)}\ \mathsf{Eval}(R_{\mathsf{prog}}, x) = C(x) : \ R_{\mathsf{prog}}, tk \leftarrow \mathsf{Gen}(C)\ \Big] \geq 1 - \mathsf{negl}(\lambda).$$

Detection correctness *For all $C \in \mathcal{C}$,*

$$\Pr\big[\mathsf{TestIntrusion}(tk, R_{\mathsf{prog}}) = \text{NO INTRUSION} : \ R_{\mathsf{prog}}, tk \leftarrow \mathsf{Gen}(C)\ \big] \geq 1 - \mathsf{negl}(\lambda).$$

Reusability after testing *We require the following for all $C \in \mathcal{C}$. Initialize the program register, $R_{\mathsf{prog}}, tk \leftarrow \mathsf{Gen}(C)$ and let ρ denote its state. Run the algorithm* TestIntrusion *on tk and R_{prog}, and let ρ' denote the state of the register R_{prog} immediately afterwards. Then, $\|\rho - \rho'\|_1 \leq \mathsf{negl}(\lambda)$.*

We note that reusability after testing will follow from detection correctness by utilizing Lemma 1.

β-intrusion-detection security. *For all QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$, we require that*

$$\Pr\left[\begin{array}{c} \mathsf{TestIntrusion}(tk, R'_{\mathsf{prog}}) = \text{NO INTRUSION} \\ \wedge \\ \forall x \in \{0,1\}^{n(\lambda)}\ \ \Pr[\mathcal{A}_{\mathsf{Main}}(tk, R_{\mathsf{adv}}, x) = C(x)] \geq \beta \end{array} : \begin{array}{c} C \leftarrow \mathcal{D} \\ R_{\mathsf{prog}}, tk \leftarrow \mathsf{Gen}(C) \\ R_{\mathsf{adv}}, R'_{\mathsf{prog}} \leftarrow \mathcal{A}_{\mathsf{intr}}(R_{\mathsf{prog}}, tk) \end{array}\right]$$

is upper bounded by $\mathsf{negl}(\lambda)$.

We also introduce the related model of secure software leasing.

Definition 39 (Strong secure software leasing [BGG+23])**.** *Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ be a family of classical circuits, where $C : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$ for all $C \in \mathcal{C}_\lambda$, and let $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$ be an ensemble on $\mathcal{C}$. A software leasing scheme for $(\mathcal{C}, \mathcal{D})$ consists of the following QPT algorithms, with the correctness and security guarantees below.*

- $\mathsf{Gen}(C)$*: Takes a circuit C, outputs a verification key and the protected version of C, a quantum state.*
- $\mathsf{Eval}(R_{\mathsf{prog}}, x)$ *: Takes the protected version of C and an input x, returns $C(x)$.*
- $\mathsf{Verify}(vk, R_{\mathsf{prog}})$*: Takes the verification key and the program register, outputs* VALID *if the returned program is valid,* INVALID *otherwise.*

Evaluation correctness *For all $C \in \mathcal{C}$,*

$$\Pr\Big[\forall x \in \{0,1\}^{n(\lambda)}\ \mathsf{Eval}(R_{\mathsf{prog}}, x) = C(x) :\ \ R_{\mathsf{prog}}, vk \leftarrow \mathsf{Gen}(C)\ \Big] \geq 1 - \mathsf{negl}(\lambda).$$

Verifcation correctness *For all $C \in \mathcal{C}$,*

$$\Pr\big[\mathsf{Verify}(vk, R_{\mathsf{prog}}) = \mathsf{NO\ INTRUSION} :\ \ R_{\mathsf{prog}}, vk \leftarrow \mathsf{Gen}(C)\ \big] \geq 1 - \mathsf{negl}(\lambda).$$

β-perfect finite-term strong lessor security with public verification *For all QPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we require that*

$$\Pr\left[\begin{array}{c|c} \mathsf{Verify}(vk, R_1) = \mathsf{VALID} & C \leftarrow \mathcal{D} \\ \wedge & : \quad R_{\mathsf{prog}}, vk \leftarrow \mathsf{Gen}(C) \\ \forall x \in \{0,1\}^{n(\lambda)}\ \Pr[\mathcal{A}_2(vk, R_2, x) = C(x)] \geq \beta & R_1, R_2 \leftarrow \mathcal{A}_1(R_{\mathsf{prog}}, vk) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

[BGG+23] construct secure software leasing schemes for various classes in the plain model.

Theorem 48 ([BGG+23, Theorem 8.4, Corollary 8.18])**.** *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists a finite-term publicly-verifiable strong secure software leasing scheme for pseudorandom functions, evasive functions, random point functions, and compute-and compare circuits.*

Finally, we show that secure software leasing schemes imply intrusion-detection for the same class.

Theorem 49. *Let $\mathcal{C}$ be a family of classical circuits and let $\mathcal{D}$ be an ensemble on $\mathcal{C}$. Suppose there exists a β-perfect finite-term publicly-verifiable strong secure software leasing scheme for $(\mathcal{C}, \mathcal{D})$. Then, there exists a β-perfect intrusion detection scheme for $(\mathcal{C}, \mathcal{D})$.*

Proof. The proof is mostly the same as that of Theorem 34, so we only sketch it.

Let SSL be a secure leasing scheme as in the theorem statement. We construct a intrusion-detection scheme SLD for $(\mathcal{C}, \mathcal{D})$ as follows. Define $\mathsf{SLD.Gen}$ and $\mathsf{SLD.Eval}$ to be the same as $\mathsf{SSL.Gen}$ and $\mathsf{SSL.Eval}$, respectively. Define $\mathsf{SLD.TestIntrusion}$ to be the rewinding version of $\mathsf{SSL.Verify}$, as obtained from Lemma 1, where we associate VALID with $\mathsf{NO\ INTRUSION}$ and $\mathsf{INVALID}$ with $\mathsf{INTRUSION}$.

It is easy to see that SLD satisfies evaluation and detection correctness. By verification correctness of SSL and Lemma 1, SLD satisfies reusability. Finally, we argue detection security as follows. Suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{intr}})$ that violates the leakage security with probability $\frac{1}{p(\lambda)}$ for some polynomial $p(\cdot)$. Then, define the adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ as follows.

$\underline{\mathcal{A}'_1(R_{\mathsf{prog}}, vk)}$

1. Run $R_{\mathsf{adv}}, R'_{\mathsf{prog}} \leftarrow \mathcal{A}_{\mathsf{intr}}(R_{\mathsf{prog}}, vk)$.
2. Output $R'_{\mathsf{prog}}, R_{\mathsf{adv}}$.

$\underline{\mathcal{A}'_2(vk, R_2, x)}$

1. Output $\leftarrow \mathcal{A}_{\mathsf{Main}}(vk, R_2, x)$.

As in Theorem 34, one can show that $\mathcal{A}'$ violates the lessor security with probability $\frac{1}{p(\lambda)}$. Crucially, note that R_1 in the lessor security condition for $\mathcal{A}'$ will have the same distribution as R'_{prog} in the intrusion detection condition for $\mathcal{A}$, therefore probability of $\mathsf{SLD.TestIntrusion}$ outputting $\mathsf{NO\ INTRUSION}$ for $\mathcal{A}$ is the same $\mathsf{SSL.Verify}$ outputting VALID for $\mathcal{A}$. Further, conditioned on the event $\mathsf{NO\ INTRUSION}$, the adversary $\mathcal{A}_{\mathsf{Main}}$ does not use R'_{prog}, hence rewinding applied by $\mathsf{TestIntrusion}$ has no effect. Therefore, R_2 in the lessor security for $\mathcal{A}'$ (conditioned on VALID) will have the same distribution as R_{adv} in the intrusion detection security (conditioned on $\mathsf{NO\ INTRUSION}$). □

Theorem 50. *Let $\mathcal{C}$ be a family of classical circuits and let $\mathcal{D}$ be an ensemble on $\mathcal{C}$. Suppose there exists a β-perfect intrusion detection scheme for $(\mathcal{C}, \mathcal{D})$. Then, it is also a β-perfect finite-term publicly-verifiable strong secure software leasing scheme for $(\mathcal{C}, \mathcal{D})$.*

Proof. We define $\mathsf{SLD.Cert}(R_{\mathsf{dec}})$ to output R_{dec} and we define $\mathsf{SLD.Verify}$ to be the same as $\mathsf{SLD'.TestIntrusion}$. Observe that if an adversary can break the secure key leasing, then the same adversary can also break the intrusion-detection game by producing a fake certificate and placing it instead of the key register. □

Corollary 10. *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exist intrusion detection schemes for pseudorandom functions, evasive functions, random point functions, and compute-and compare circuits.*

Proof. Invoke Theorem 48 and Theorem 49. □

9 Acknowledgements

We thank Prabhanjan Ananth, Thiago Bergamaschi, Shafi Goldwasser, and Bhaskar Roberts for helpful comments and discussions.

Alper Çakan and João Ribeiro were supported by the following grants of Vipul Goyal: NSF award 1916939, DARPA SIEVE program, a gift from Ripple, a DoE NETL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cylab seed funding award. João Ribeiro's research was also supported by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT - Fundação para a Ciência e a Tecnologia.

References

[Aar05] Scott Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1(1):1–28, 2005.

[Aar16] Scott Aaronson. The complexity of quantum states and transformations: From quantum money to black holes, 2016.

[AARR03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM side—channel(s). In Burton S. Kaliski, Çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 29–45, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[AC02] Mark Adcock and Richard Cleve. A quantum goldreich-levin theorem with cryptographic applications. In Helmut Alt and Afonso Ferreira, editors, *STACS 2002*, pages 323–334, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 41–60, 2012.

[ADN+19] Divesh Aggarwal, Ivan Damgård, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, João Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 510–539, Cham, 2019. Springer International Publishing.

[AGLL23] Prabhanjan Ananth, Vipul Goyal, Jiahui Liu, and Qipeng Liu. Unpublished manuscript. 2023.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography*, pages 474–495, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[AKL23] Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. Cloning games: A general framework for unclonable primitives. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 66–98, Cham, 2023. Springer Nature Switzerland.

[AKN+23] Shweta Agrawal, Fuyuki Kitagawa, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Public key encryption with secure key leasing. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 581–610, Cham, 2023. Springer Nature Switzerland.

[AL21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 501–530, Cham, 2021. Springer International Publishing.

[ALP22] Adil Ahmad, Sangho Lee, and Marcus Peinado. Hardlog: Practical tamper-proof system auditing using a novel audit device. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1791–1807, 2022.

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *Theory of Cryptography*, pages 52–73, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[BDIR18] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 531–561, 2018.

[BFO+21] Gianluca Brian, Antonio Faonio, Maciej Obremski, João Ribeiro, Mark Simkin, Maciej Skórski, and Daniele Venturi. The mother of all leakages: How to simulate noisy leakages via bounded leakage (almost) for free. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021*, pages 408–437. Springer, 2021.

[BGG+23] James Bartusek, Sanjam Garg, Vipul Goyal, Dakshita Khurana, Giulio Malavolta, Justin Raizes, and Bhaskar Roberts. Obfuscation and outsourced computation with certified deletion. Cryptology ePrint Archive, Paper 2023/265, 2023. https://eprint.iacr.org/2023/265.

[BK23] James Bartusek and Dakshita Khurana. Cryptography with certified deletion. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 192–223, 2023.

[BKKV10] Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 501–510, 2010.

[BSW11] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 89–108, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[CGG+20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, Ashutosh Kumar, Xin Li, Raghu Meka, and David Zuckerman. Extractors and secret sharing against bounded collusion protocols. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1226–1242, 2020.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 398–412, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[CKOS22] Nishanth Chandran, Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Short leakage resilient and non-malleable secret sharing schemes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022*, pages 178–207. Springer, 2022.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 556–584, Cham, 2021. Springer International Publishing.

[CLM+21] Matthias Christandl, Felix Leditzky, Christian Majenz, Graeme Smith, Florian Speelman, and Michael Walter. Asymptotic performance of port-based teleportation. *Communications in Mathematical Physics*, 381(1):379–451, 2021.

[CLW14] Kai-Min Chung, Xin Li, and Xiaodi Wu. Multi-source randomness extractors against quantum side information, and their applications. *arXiv e-prints*, page arXiv:1411.2315, November 2014.

[DD10] Simon Pierre Desrosiers and Frédéric Dupuis. Quantum entropic security and approximate quantum encryption. *IEEE Transactions on Information Theory*, 56(7):3455–3464, 2010.

[DDF19] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. *J. Cryptol.*, 32(1):151–177, 2019.

[DFS15] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 401–429, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[DGT+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *Theory of Cryptography*, pages 361–381, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[DHLW10] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 511–520, 2010.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 621–630, New York, NY, USA, 2009. Association for Computing Machinery.

[DLT02] David P. DiVincenzo, Debbie W. Leung, and Barbara M. Terhal. Quantum data hiding. *IEEE Transactions on Information Theory*, 48(3):580–598, 2002.

[DLW06] Giovanni Di Crescenzo, Richard Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 225–244, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 227–237, 2007.

[DPVR12] Anindya De, Christopher Portmann, Thomas Vidick, and Renato Renner. Trevisan's extractor in the presence of quantum side information. *SIAM Journal on Computing*, 41(4):915–940, 2012.

[Dzi06] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 207–224, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[EW02] T. Eggeling and R. F. Werner. Hiding classical data in multipartite quantum states. *Phys. Rev. Lett.*, 89:097905, Aug 2002.

[FKPR10] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N Rothblum. Leakage-resilient signatures. In *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7*, pages 343–360. Springer, 2010.

[FNV15] Antonio Faonio, Jesper Buus Nielsen, and Daniele Venturi. Mind your coins: Fully leakage-resilient signatures with graceful degradation. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming*, pages 456–468, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[GK18] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, page 685–698, 2018.

[HLAWW16] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. *Journal of Cryptology*, 29(3):514–551, 2016.

[HLS05] Patrick Hayden, Debbie Leung, and Graeme Smith. Multiparty data hiding of quantum information. *Phys. Rev. A*, 71:062339, Jun 2005.

[HLSW04] Patrick Hayden, Debbie Leung, Peter W Shor, and Andreas Winter. Randomizing quantum states: Constructions and applications. *Communications in Mathematical Physics*, 250:371–391, 2004.

[HSR03] Michael Horodecki, Peter W. Shor, and Mary Beth Ruskai. Entanglement breaking channels. *Reviews in Mathematical Physics*, 15(06):629–641, aug 2003.

[IH08] Satoshi Ishizaka and Tohya Hiroshima. Asymptotic teleportation scheme as a universal programmable quantum processor. *Physical review letters*, 101(24):240501, 2008.

[IH09] Satoshi Ishizaka and Tohya Hiroshima. Quantum teleportation scheme by selecting one of multiple output ports. *Phys. Rev. A*, 79:042306, Apr 2009.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 463–481, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[KK12] Roy Kasher and Julia Kempe. Two-source extractors secure against quantum adversaries. *Theory of Computing*, 8(21):461–486, 2012.

[KN22] Fuyuki Kitagawa and Ryo Nishimaki. Functional encryption with secure key leasing. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 569–598, Cham, 2022. Springer Nature Switzerland.

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography*, pages 31–61, Cham, 2021. Springer International Publishing.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Advances in Cryptology — CRYPTO '96*, pages 104–113, 1996.

[KR19] Yael Tauman Kalai and Leonid Reyzin. A survey of leakage-resilient cryptography. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 727–794. ACM, 2019.

[KRS09] Robert König, Renato Renner, and Christian Schaffner. The operational meaning of min- and max-entropy. *IEEE Trans. Inf. Theor.*, 55(9):4337–4347, sep 2009.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, pages 703–720. Springer, 2009.

[LAF+10] Andrew Lutomirski, Scott Aaronson, Edward Farhi, David Gosset, Jonathan A. Kelner, Avinatan Hassidim, and Peter W. Shor. Breaking and making quantum money: Toward a new quantum cryptographic protocol. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010*, pages 20–31, 2010.

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography, TCC 2022*, pages 294–323. Springer, 2022.

[LPW18] Ludovico Lami, Carlos Palazuelos, and Andreas Winter. Ultimate data hiding in quantum mechanics and beyond. *Communications in Mathematical Physics*, 361:661–708, 2018.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *Theory of Cryptography Conference*, pages 278–296. Springer, 2004.

[MWW09] William Matthews, Stephanie Wehner, and Andreas Winter. Distinguishability of quantum states under restricted families of measurements with an application to quantum data hiding. *Communications in Mathematical Physics*, 291:813–843, 2009.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, pages 18–35, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[PGMP19] Thomas Prest, Dahmun Goudarzi, Ange Martinelli, and Alain Passelègue. Unifying leakage models on a Rényi day. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 683–712, Cham, 2019. Springer International Publishing.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 142–159, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[QS01] Jean-Jacques Quisquater and David Samyde. ElectroMagnetic Analysis (EMA): Measures and counter-measures for smart cards. In Isabelle Attali and Thomas Jensen, editors, *Smart Card Programming and Security*, pages 200–210, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[RRV02] Ran Raz, Omer Reingold, and Salil P. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002.

[SJEL14] Arunesh Sinha, Limin Jia, Paul England, and Jacob R. Lorch. Continuous tamper-proof logging using TPM 2.0. In Thorsten Holz and Sotiris Ioannidis, editors, *Trust and Trustworthy Computing*, pages 19–36, Cham, 2014. Springer International Publishing.

[SV19] Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 480–509, 2019.

[SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. Cryptology ePrint Archive, Paper 2013/454, 2013. https://eprint.iacr.org/2013/454.

[SYC04] Richard T. Snodgrass, Shilong Stanley Yao, and Christian Collberg. Tamper detection in audit logs. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, page 504–515. VLDB Endowment, 2004.

[TDL01] Barbara M. Terhal, David P. DiVincenzo, and Debbie W. Leung. Hiding bits in Bell states. *Phys. Rev. Lett.*, 86:5807–5810, Jun 2001.

[TFKW13] Marco Tomamichel, Serge Fehr, Jedrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, oct 2013.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, 2001.

[VZ21] Thomas Vidick and Tina Zhang. Classical proofs of quantum knowledge. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 630–660, Cham, 2021. Springer International Publishing.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611, 2017.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 408–438, Cham, 2019. Springer International Publishing.

[Zha23] Mark Zhandry. Quantum money from abelian group actions. Cryptology ePrint Archive, Paper 2023/1097, 2023. https://eprint.iacr.org/2023/1097.

A Quantum Information Preliminaries

Here we present some quantum information theory concepts that are needed in Appendix B. We refer the reader to the book of Nielsen and Chuang [NC10] and Watrous [Wat18] for an overview of qunatum information theory. We briefly mention some additional concepts, and refer the reader to the same references for details.

Trace Distance. We will make use of the notion of trace distance between states.

Definition 40 (Trace distance)**.** *The* trace distance *between two mixed states with associated density matrices ρ and σ, denoted by $D(\rho,\sigma)$, is given by*

$$D(\rho,\sigma)=\frac{1}{2}\|\rho-\sigma\|_1,$$

where $\|\rho\|_1=\mathrm{Tr}\left[\sqrt{\rho^\dagger\rho}\right]$ is the trace norm*. We write $\rho\approx_\varepsilon\sigma$ whenever $D(\rho,\sigma)\leq\varepsilon$.*

The trace distance is a metric and has the following useful interpretation: If $D(\rho,\sigma)\leq\varepsilon$, then any POVM applied to states with density matrices ρ and σ yields classical measurement outcome distributions, say $(p_1,\ldots,p_m)$ and $(q_1,\ldots,q_m)$, which are ε-close in statistical distance, i.e., $\frac{1}{2}\sum_{i=1}^m|p_i-q_i|\leq\varepsilon$. Therefore, when ρ and σ are classical mixed states, the trace distance corresponds exactly to the statistical distance between the two probability distributions inducing ρ and σ.

Definition 41 (Quantum channel [Wat18])**.** *A quantum channel is a linear map $\Phi:\mathcal{X}\to\mathcal{Y}$ that is both trace preserving and completely positive.*

Lemma 16 (Post-processing lemma for trace distance [NC10, Theorem 9.2])**.** *Let Φ be a quantum channel and ρ,σ density matrices. Then,*

$$D(\Phi(\rho,\sigma))\leq D(\rho,\sigma).$$

Definition 42 (Completely dephasing channel [Wat18])**.** *Let X be a register with alphabet Σ. The* completely dephasing channel *over X, denoted by Δ^X, is defined as follows.*

$$\Delta^X(\rho)=\sum_{a\in\Sigma}\mathrm{Tr}(E_{a,a}\rho)E_{a,a}$$

Definition 43 (Quantum-to-classical channel)**.** *A quantum channel $\Phi:\mathcal{X}\to\mathcal{A}\otimes\mathcal{B}$ is called* classical over A *if*

$$(\Delta^A\otimes I^B)\Phi=\Phi$$

Lemma 17 (Tracing out commutes with channel on the traced out system)**.** *Let $\Phi:\mathcal{X}\to\mathcal{Y}$ be a quantum channel. Then, for any register A and any joint state ρ of (A,X), we have*

$$\mathrm{Tr}_Y((I^A\otimes\Phi^X)(\rho))=\mathrm{Tr}_X(\rho).$$

Proof. Let $\rho=\sum_i p_i|\psi_i\rangle\langle\psi_i|$ be a diagonalization of ρ. Let $|\psi_i\rangle=\sum_{a,x}\alpha_{i,a,x}\alpha_{i,a,x}|a\rangle|x\rangle$. Then,

$$\rho=\sum_{i,a,x,a',x'}p_i\alpha_{i,a,x}\alpha^*_{i,a',x'}|a\rangle\langle a'|\otimes|x\rangle\langle x'|.$$

Since Φ is trace preserving and $\mathrm{Tr}(|x\rangle\langle x'|)=\delta_{x,x'}$, we get

$$\begin{aligned}\mathrm{Tr}_Y((I^A\otimes\Phi^X))\rho&=\sum_{i,a,x,a',x'}p_i\alpha_{i,a,x}\alpha^*_{i,a',x'}\,\mathrm{Tr}\big(\Phi(|x\rangle\langle x'|)\big)|a\rangle\langle a'|\\&=\sum_{i,a,x,a'}p_i\alpha_{i,a,x}\alpha^*_{i,a',x}|a\rangle\langle a'|.\end{aligned}$$

We also have

$$\begin{aligned}\mathrm{Tr}_X(\rho) &= \sum_{i,a,x,a',x'} p_i \alpha_{i,a,x} \alpha^*_{i,a',x'} \mathrm{Tr}\big(|x\rangle\langle x'|\big) |a\rangle\langle a'| \\ &= \sum_{i,a,x,a'} p_i \alpha_{i,a,x} \alpha^*_{i,a',x} |a\rangle\langle a'|,\end{aligned}$$

which completes the proof. □

A.1 Min-Entropy and Randomness Extractors

As one of our main tools, we will require explicit constructions of seeded randomness extractors that are secure against quantum side information and multi-source randomness extractors which are resilient to quantum adversaries with shared entanglement. These objects have been studied under many different models. For seeded extractors we will use the model of De, Portmann, Vidick, and Renner [DPVR12] and for multi-source extractors we focus on the model of Kasher and Kempe [KK12] and Chung, Li, and Wu [CLW14].

We first start with entropy definitions and useful lemmas.

Definition 44 (Min-entropy)**.** *The* min-entropy *of a random variable X supported on a finite set $\mathcal{X}$, denoted by $\mathbf{H}_\infty(X)$, is given by*

$$\mathbf{H}_\infty(X) = -\log \max_{x \in \mathcal{X}} \Pr[X = x].$$

Definition 45 (k-source)**.** *A random variable X is said to be a* k-source *if $\mathbf{H}_\infty(X) \geq k$.*

Definition 46 (Average conditional min-entropy [DORS08])**.** *Let X, Y be two (possibly correlated) random variables. We define the average conditional min-entropy of X given Y as*

$$\mathbf{H}_\infty(X|Y) = -\log \mathbb{E}_{y \leftarrow Y} \max_x \Pr[X = x | Y = y].$$

Definition 47 (Quantum min-entropy)**.** *Let X be a register in the state ρ. We define the min-entropy of X to be*

$$\mathbf{H}_\infty(X)_\rho = -\log(\lambda_{max}(\rho)).$$

where $\lambda_{max}(\rho)$ denotes the largest eigenvalue of the density matrix ρ. When the state ρ is clear from context, we will simply write $\mathbf{H}_\infty(X)$

Definition 48 (Quantum conditional min-entropy)**.** *Let X, Y be registers with state space $\mathcal{X}, \mathcal{Y}$ and joint state ρ. We define the conditional min-entropy of X given Y as*

$$\mathbf{H}_\infty(X|Y)_\rho = -\log \min_{\sigma \in \mathcal{Y}} \{ \min_{\lambda \in \mathbb{R}} \lambda I \otimes \sigma \geq \rho \}.$$

When ρ is a cq-state, $\mathbf{H}_\infty(X|Y)$ has an operational meaning in terms of the optimal guessing probability for X given Y.

Definition 49. *Let X, Y be two registers with state spaces $\mathcal{X}, \mathcal{Y}$ and joint cq-state $\rho = \sum_x |x\rangle\langle x| \otimes \sigma_x^Y$. Then, the guessing probability of X given Y, denoted by $p_{\mathsf{guess}}(X|Y)$, is given by*

$$p_{guess}(X|Y) = \max_{\{\mu_x\}_x \ POVM} \mathrm{Tr}\big(\mu_x \rho^Y\big).$$

Lemma 18 ([KRS09, Theorem 1]). *Let X, Y be two registers in a cq-state. Then,*

$$\mathbf{H}_\infty(X|Y) = -\log p_{guess}(X|Y).$$

We will also utilize the following lemma.

Lemma 19 (Separable chain rule for quantum min-entropy [DD10, Lemma 7]). *Let A, B, C be registers with some joint, separable state $\rho = \sum_i p_i \tau_i^{AB} \otimes \sigma_i^C$. Then,*

$$\mathbf{H}_\infty(A|B,C) \geq \mathbf{H}_\infty(A|B) - \log|C|.$$

Now we move to extractors.

Definition 50 (Strong seeded extractor). *A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is said to be a (k,ε)-*strong seeded extractor *if for every pair of random variables (X, W) with $X \in \{0,1\}^n$ and $\widetilde{\mathbf{H}}_\infty(X|W) \geq k$ it holds that*

$$\mathsf{Ext}(X, U_d), U_d, W \approx_\varepsilon U_m, U_d, W.$$

A seeded extractor Ext is said to be linear *if $\mathsf{Ext}(\cdot, s)$ is a linear function for every $s \in \{0,1\}^d$.*

Definition 51 (Quantum-proof seeded extractor [DPVR12]). *A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is said to be a (k,ε)-*strong quantum-proof seeded extractor *if for any cq-state $\rho \in \mathcal{H}^{\otimes n} \otimes \mathcal{Y}$ of the registers X, Y with $\mathbf{H}_\infty(X|Y) \geq k$, we have*

$$\mathsf{Ext}(X, S), Y, S \approx_\varepsilon U_m, Y, S$$

where $S \leftarrow \{0,1\}^d$.

Note that any quantum-proof seeded extractor is also a classical seeded extractor with the same parameters. We will use the following explicit linear strong seeded extractor due to Trevisan [Tre01] with improvements by Raz, Reingold, and Vadhan [RRV02], which was later shown to be quantum-proof by De, Portmann, Vidick, and Renner [DPVR12].

Lemma 20 ([Tre01, RRV02, DPVR12]). *There exists an explicit linear (k,ε)-strong quantum-proof seeded extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log^3(n/\varepsilon))$ and $m = k - O(d)$.*

When we do not insist on linearity, we can use the following extractor with slightly improved parameters.

Theorem 51 ([DPVR12]). *There exists an explicit (k,ε)-strong quantum-proof seeded extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log^2(n/\varepsilon)\log m)$ and $m = k - 4\log(1/\varepsilon) - O(1)$.*

B BB84-based Cryptographic Schemes Resilient to Unbounded Classical Leakage

In this section, we define the weaker notion of unbounded classical leakage-resilience and use BB84 states to construct secret key encryption and secret sharing schemes that satisfy this notion. While this security notion is weaker than LOCC leakage-resilience, we believe constructions based on BB84 states are nevertheless useful since they are vastly more practical than highly entangled states like coset states.

We first present some results regarding unbounded classical leakage on BB84 states that will be useful in most of our schemes.

B.1 Monogamy-of-Entanglement Games

In this section we introduce the notion of a Monogamy-of-Entanglement game (MoE game), as first studied by Tomamichel, Fehr, Kaniewski, and Wehner [TFKW13], along with useful games and associated results.

An MoE game is played by three parties, Alice, Bob, and Charlie and is parameterized by a list Θ of possible POVM measurements performed by Alice. The game proceeds as follows:

1. Bob and Charlie select a tripartite quantum state ρ_{ABC}. Alice has access to the contents of register A, Bob has access to the contents of register B, and Charlie has access to the contents of register C.

2. Alice samples a POVM measurement $\theta \leftarrow \Theta$ and measures the contents of register A according to θ. Let x denote the measurement outcome. Alice reveals θ to Bob and Charlie.

3. Bob and Charlie win the game if they *both* guess x given their quantum registers and knowledge of θ.

A quantity of interest in an MoE game is the winning probability of Bob and Charlie, maximized over the choice of the tripartite quantum state ρ_{ABC} and strategies of Bob and Charlie. In our work we will use bounds on the winning probability for the basic n-qubit "BB84" MoE game already studied in [TFKW13], where register A contains n qubits and for each $i \in [n]$ Alice measures the i-th qubit with respect to the computational or Hadamard basis independently with probability $1/2$. The following result was established in [TFKW13].

Lemma 21 ([TFKW13, Theorem 3])**.** *The winning probability of the n-qubit BB84 MoE game is $\left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^n$.*

Lemma 22 (Entropy loss of BB84 states with unbounded classical leakage)**.** *Let X, θ be independent and uniformly distributed over $\{0,1\}^\lambda$ and consider the BB84 state $H^\theta|X\rangle$. Let Leak be any quantum-to-classical channel. Then, we have that*

$$\mathbf{H}_\infty(X|\mathsf{Leak}(H^\theta|X\rangle), \theta) \geq C_{BB84} \cdot \lambda$$

where $C_{BB84} = -\log\left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right) > 0.22$.

Proof. The desired statement follows by framing the task of guessing X as an instance of the λ-qubit BB84 MoE game from Appendix B.1. To see this, consider the tripartite quantum state ρ_{ABC} constructed as follows:

1. Generate λ EPR pairs $|\Phi_1\rangle, \ldots, |\Phi_\lambda\rangle$. Store the first half of each pair in Alice's register A, and the second half in another register A'.

2. Compute the classical leakage L by applying Leak to the contents of A'.

3. Store L in Bob's and Charlie's registers, B and C.

Note that if Alice samples θ uniformly at random from $\{0,1\}^\lambda$ and measures the i-th qubit in A according to the computational basis if $\theta_i = 0$ and the Hadamard basis if $\theta_i = 1$ obtaining the measurement outcome $X \in \{0,1\}^\lambda$, then, after these measurements, the register A' holds the state $H^\theta|X\rangle$. Moreover, since the measurements above and the leakage function Leak are applied to disjoint sets of registers, these operations commute and so $L \leftarrow \mathsf{Leak}(H^\theta|X\rangle)$. As Bob and Charlie

both have access to (L, θ), the winning probability of this MoE game equals the optimal probability of guessing X given (L, θ). According to Lemma 21, this probability is exactly

$$\left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{\lambda},$$

and so

$$\mathbf{H}_{\infty}(X|L,\theta) = -\log p_{\mathsf{guess}}(X|L,\theta) = -\lambda \cdot \log\left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right) = C_{BB84} \cdot \lambda,$$

where the first equality uses Lemma 18. □

We can also extend Lemma 22 against attacks that are composed of unbounded classical leakage and bounded quantum leakage.

Lemma 23 (Entropy loss of BB84 states with unbounded classical and bounded quantum leakage)**.** *Let X, θ be independent and uniformly distributed over $\{0,1\}^{\lambda}$ and consider the BB84 state $H^{\theta}|X\rangle$. Let $\ell_c(\lambda), \ell_q(\lambda)$ be functions denoting the classical leakage and qubit leakage size, respectively. Then, for any quantum channel* Leak *with $\ell_c(\lambda)$ bit classical output and $\ell_q(\lambda)$ qubit output, we have*

$$\mathbf{H}_{\infty}(X|\mathsf{Leak}(H^{\theta}|X\rangle),\theta) \geq C_{BB84} \cdot \lambda - \ell_q(\lambda).$$

We first need a technical lemma that will help show that the quantum leakage will be unentangled from the rest of the system.

Lemma 24 ([HSR03, Theorem 1])**.** *Any channel of the form*

$$\Phi(\rho) = \sum_k R_k \operatorname{Tr}(F_k \rho)$$

where $\{F_k\}_k$ is a POVM and each R_k is a density matrix, is entanglement breaking.

More formally, for such a channel Φ on register X, for any other register Y and any state σ of (X, Y), we have that $(\Phi^X \otimes I^Y)(\sigma)$ is separable.

Proof. Without loss of generality assume that the first $\ell_c(\lambda)$ registers of the output of Leak are classical. Define the registers A, B, C, D where C will contain the classical leakage and D will contain the quantum leakage, and consider the *cccq* state

$$\rho = \sum_x \frac{1}{2^{\lambda}} |x\rangle\langle x| \otimes \left(\sum_{\theta} \frac{1}{2^{\lambda}} |\theta\rangle\langle\theta| \otimes \mathsf{Leak}(H^{\theta}|x\rangle)\right)$$

over these registers.

We have $(\Delta^C \otimes I^D)\mathsf{Leak} = \mathsf{Leak}$ by Definition 43. We also have $\Delta^C(\sigma) = \sum_a Tr(E_{a,a}\rho)E_{a,a}$ by Definition 42 where each $E_{a,a}$ is a density matrix while $\{E_{a,a}\}_a$ form a POVM. Hence, by Lemma 24, is Δ^C is an entanglement breaking channel and therefore

$$\mathsf{Leak}(H^{\theta}|x\rangle) = \sum_i p_i^{\theta,x} (\tau_i^{\theta,x})^C \otimes (\xi_i^{\theta,x})^D.$$

for some density matrices $\{\tau_i^{\theta,x}\}_i, \{\xi_i^{\theta,x}\}_i$ and probability distribution $\{p_i^{\theta,x}\}$ for each θ, x. Then,

$$\rho = \sum_{x,\theta,i} \frac{p_i^{\theta,x}}{4^{\lambda}} |x\rangle\langle x| \otimes |\theta\rangle\langle\theta| \otimes (\tau_i^{\theta,x})^C \otimes (\xi_i^{\theta,x})^D.$$

Hence, D is separable from rest of the system, and then by Lemma 19 we have

$$\mathbf{H}_\infty(A,(B,C),D) \geq \mathbf{H}_\infty(A|B,C) - \ell_q(\lambda) \tag{4}$$

since D consists of $\ell_q(\lambda)$ qubits.

Observe that $\mathbf{H}_\infty(A|B,C)$ is $\mathbf{H}_\infty(X|\theta,C)$ and C is classical. Hence, by Lemma 22 we have $\mathbf{H}_\infty(A|B,C) \geq C_{BB84} \cdot \lambda$. Finally combining this with Equation (4) yields the result. □

B.2 Private-Key Encryption

We introduce private-key encryption schemes that are resilient against unbounded classical leakage from encryption and decryption keys. We then show how to construct using our results regarding leakage from BB84 states (Lemma 22, Lemma 23).

Definition 52 (Unbounded classical leakage-resilient private-key encryption)**.** *A private-key encryption scheme* SKE *is said to be* $(*,\ell_q(\lambda))$*-leakage-resilient if for all polynomials* $\ell_c(\cdot)$ *and* $p(\cdot)$*, for all tuples of QPT adversaries* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak,enc}}, \mathcal{A}_{\mathsf{Leak,dec}})$ *such that the output of* $\mathcal{A}_{\mathsf{Leak,enc}}$ *and* $\mathcal{A}_{\mathsf{Leak,dec}}$ *consists of* $\ell_c(\lambda)$ *classical bits and* $\ell_q(\lambda)$ *qubits, respectively, the advantage of* $\mathcal{A}$ *in the following game is negligible.*

1. *The challenger runs* $R_{\mathsf{enc}}, R_{\mathsf{dec}} \leftarrow \mathsf{SKE.KeyGen}(1)$.
2. *Adversary outputs messages*
$$(m_0^{(1)}, m_1^{(1)}), \ldots, (m_0^{(p(\lambda))}, m_1^{(p(\lambda))}) \leftarrow \mathcal{A}_{\mathsf{Main}}(1).$$
3. *Challenger samples a challenge bit* $b \leftarrow \{0,1\}$.
4. *For* $i = 1$ *to* $p(\lambda)$*, challenger sets* $R_{i,\mathsf{ct}} \leftarrow \mathsf{SKE.Enc}(R_{\mathsf{enc}}, m_b^{(i)})$.
5. *The leakage adversaries get access to their keys and produce leakages*
$$R_{\mathsf{leak}} \leftarrow \mathcal{A}_{\mathsf{Leak,enc}}(R_{\mathsf{enc}}), \mathcal{A}_{\mathsf{Leak,dec}}(R_{\mathsf{dec}}).$$
6. *Given the leakage, adversary outputs a guess* $b' \leftarrow \mathcal{A}_{\mathsf{Main}}(R_{\mathsf{ct}}, R_{\mathsf{leak}})$.
7. *Output* 1 *iff* $b = b'$.

We define the advantage of $\mathcal{A}$ *to be* $\left|\Pr[b' = b] - \frac{1}{2}\right|$.

We also require overwhelming correctness in the natural way, and assume that Dec *is rewound after each use.*

Our definition can be seen as an everlasting security for messages encrypted before the leakage attack. This is necessary since the leakage attack will collapse the key to a state known by the adversary, after which it is impossible to satisfy security. However, one can use intrusion-detection schemes (Section 8) to throw away the key after a leakage attack and establish new keys.

Theorem 52. *Let* $m(\cdot)$ *be a polynomial denoting the message size and* $\ell_q(\cdot)$ *be any polynomial denoting the qubit leakage size. Let* PKE *be a public-key encryption scheme for message of size* $m(\lambda)$ *whose public-key length is* $k(\lambda)$ *and key generation algorithm* $\mathsf{PKE.KeyGen}$ *has randomness complexity* $r(\lambda)$*. Let* Ext *be the extractor obtained by instantiating Theorem 51 with* $n = N$*,* $\varepsilon = (\log(\lambda))^{-\log(\lambda)}$ *and* $k = C_{BB84} \cdot N - \ell_q(\lambda)$ *where we define* $N(\lambda) = \frac{1}{C_{BB84}}(\ell_q(\lambda) + r(\lambda) + 4\log(\lambda)\log(\log(\lambda))$*. Then, the following private-key encryption scheme* SKE *is* $(*,\ell_q(\lambda))$*-leakage-resilient and*

- *its encryption key consists of* $N(\lambda) + k(\lambda) + O(\log^2(n)\log^2(\log(n))\log(r(\lambda)))$ *classical bits,*
- *its decryption key consists of* $N(\lambda) + O(1)$ *qubits.*

SKE

- $\underline{\mathsf{SKE.KeyGen}(1)}$
  1. *Sample* $x, \theta, s \leftarrow \{0,1\}^{N(\lambda)}$.
  2. $pk, sk \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; \mathsf{Ext}(x, s))$.
  3. $R_{\mathsf{enc}} \leftarrow (\theta, s, pk)$.
  4. $R_{\mathsf{dec}} \leftarrow H^\theta |x\rangle$.
  5. *Output* $(R_{\mathsf{enc}}, R_{\mathsf{dec}})$.
- $\underline{\mathsf{SKE.Enc}((\theta, s, pk), m)}$
  1. $ct \leftarrow \mathsf{PKE.Enc}(pk, m)$.
  2. *Output* ct, θ, s.
- $\underline{\mathsf{SKE.Dec}(R_{\mathsf{dec}}, (ct, \theta, s))}$
  1. *Apply* $H^{-\theta}$ *to* R_{dec}.
  2. *Measure* R_{dec} *in computational basis to obtain* x.
  3. $pk, sk \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; \mathsf{Ext}(x, s))$.
  4. *Restore the key as* $R_{\mathsf{dec}} \leftarrow H^\theta |x\rangle$.
  5. *Output* $\mathsf{PKE.Dec}(sk, ct)$.

Proof. It is straightforward to show that correctness holds with probability 1.

We will prove the security using a hybrid argument. Observe that, by Theorem 51, the output length of Ext is $r(\lambda)$ as required. Define the first hybrid, Hyb_0 to be the original security game. Define the second hybrid Hyb_1 by replacing the line

$$pk \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; \mathsf{Ext}(x, s))$$

with

$$pk \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; U_{r(\lambda)})$$

in $\mathsf{SKE.KeyGen}$. By the entropy lemma for BB84 states given unbounded classical leakage (Lemma 23), we have that $\mathbf{H}_\infty(x|\mathcal{A}_{\mathsf{Leak,dec}}(R_{dec}), \theta) \geq C_{BB84} \cdot N - \ell(\lambda)$. Hence, since $\mathcal{A}_{\mathsf{Leak,dec}}(R_{dec}), \theta$ are independent of the seed s and Ext is a strong quantum-proof extractor (Definition 51), we get

$$\mathsf{Ext}(x, s), s, \mathcal{A}_{\mathsf{Leak,dec}}(R_{dec}), \theta \approx U_{r(\lambda)}, s, \mathcal{A}_{\mathsf{Leak,dec}}(R_{dec}), \theta$$

Then, by post-processing (Lemma 16), it follows that

$$\mathsf{PKE.KeyGen}(1^\lambda; \mathsf{Ext}(x, s)), s, \mathcal{A}_{\mathsf{Leak,dec}}(R_{dec}), \theta \approx \mathsf{PKE.KeyGen}(1^\lambda; U_{r(\lambda)}), s, \mathcal{A}_{\mathsf{Leak,dec}}(R_{dec}), \theta$$

which implies $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$. Now, for a contradiction, suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak,enc}}, \mathcal{A}_{\mathsf{Leak,dec}})$ that wins the leakage-resilience game, Hyb_0, with non-negligible advantage. By $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$, the adversary $\mathcal{A}$ wins the game Hyb_1 also with non-negligible advantage. We construct the following adversary $\mathcal{A}'$ for the security game of the public-key encryption scheme PKE.

$\underline{\mathcal{A}'}$

1. Output $\mathcal{A}_{\mathsf{Main}}(1)$ as the selected plaintexts.
2. On input (pk, ct), sample $x, \theta, s \leftarrow \{0,1\}^{N(\lambda)}$ and output

$$\mathcal{A}_{\mathsf{Main}}(ct, (\mathcal{A}_{\mathsf{Leak,enc}}(\theta, s, pk), \mathcal{A}_{\mathsf{Leak,dec}}(H^\theta|x\rangle)))$$

It is easy to see that Hyb_1 is exacty the same as the public-key encryption indistinguishability game as played by $\mathcal{A}'$. Hence, $\mathcal{A}'$ breaks the security of PKE, which is a contradiction. □

B.3 Secret Sharing Schemes Resilient to Unbounded Classical Leakage based on BB84 States

B.3.1 Secret Sharing Schemes

We introduce basic definitions of access structures and secret sharing schemes.

Definition 53 (Access structure)**.** *We say that $\Gamma \subseteq 2^{\mathcal{S}}$ is an* access structure *if $A \in \Gamma$ and $A \subseteq B$ implies that $B \in \Gamma$. We call sets $A \in \Gamma$* authorized*.*

Definition 54 (Secret sharing)**.** *A family of functions $(\mathsf{Share}, (\mathsf{Rec}_T)_{T\in\Gamma})$ is an ε-*secret sharing scheme *for an access structure $\Gamma \subseteq 2^{[n]}$ on n parties with message space $\mathcal{X}$ and share space $\mathcal{S}$ if $\mathsf{Share} : \mathcal{X} \to \mathcal{S}^{[n]}$ and $\mathsf{Rec}_T : \mathcal{S}^T \to \mathcal{X}$ are quantum channels and the following two properties are satisfied:*

- ***Correctness:*** *If $T \in \Gamma$ (i.e., T is authorized) it holds that*

$$\mathrm{Tr}(|x\rangle\langle x|\mathsf{Rec}_T(\mathsf{Share}(x)_T)) = 1$$

  for any message $x \in \mathcal{X}$.

- ***ε-Privacy:*** *If $T \notin \Gamma$ (i.e., T is unauthorized) it holds that*

$$\mathsf{Share}(x)_T \approx_\varepsilon \mathsf{Share}(x')_T$$

  for any two messages $x, x' \in \mathcal{X}$.

*In the special case where $T \in \Gamma$ if and only if $|T| \geq t$ for some threshold t, we say that $(\mathsf{Share}, \mathsf{Rec})$ is a t-*out-of-n *ε-*secret sharing scheme*.*

B.3.2 Leakage-Resilient Secret Sharing for General Access Structures

We describe and analyze an efficient compiler that takes as input an appropriate secret sharing scheme realizing an access structure without singletons[21] and outputs a secret sharing scheme for the same access structure which is additionally leakage-resilient against unbounded classical local leakage and bounded quantum leakage. The compiler is inspired by the approach of Chandran, Kanukurthi, Obbattu, and Sekar [CKOS22] for *bounded* classical leakage (which itself improves a previous compiler of [ADN+19]) and uses the entropic monogamy-of-entanglement properties of random BB84 states, as shown in Lemmas 22 and 23.

[21] Local leakage-resilience is trivially unachievable for access structures with singletons.

Definition 55 (Unbounded-leakage-resilient secret sharing)**.** *We say that a secret sharing scheme* $(\mathsf{Share}, (\mathsf{Rec}_T)_{T\in\Gamma})$ *is* ε-unbounded-leakage-resilient *if for any unauthorized set* $T \notin \Gamma$*, any family of leakage functions* $\{\mathsf{Leak}_i\}_{i\notin T}$ *with possibly quantum input but classical output (but not sharing entangled states), and any two messages* $m, m' \in \mathcal{M}$ *we have that*

$$(\mathsf{Sh}_i)_{i\in T}, (\mathsf{Leak}_i(\mathsf{Sh}_i))_{i\notin T} \approx_\varepsilon (\mathsf{Sh}'_i)_{i\in T}, (\mathsf{Leak}_i(\mathsf{Sh}'_i))_{i\notin T}, \tag{5}$$

where $(\mathsf{Sh}_i)_{i\in[n]} \leftarrow \mathsf{Share}(m)$ *and* $(\mathsf{Sh}'_i)_{i\in[n]} \leftarrow \mathsf{Share}(m')$*.*

Similarly, we say that the scheme is ε*-leakage-resilient to* $(*, \ell)$*-leakage if it satisfies Equation (5) for any family of leakage functions* $\{\mathsf{Leak}_i\}_{i\notin T}$ *whose outputs each consist of arbitrary size classical bits and* ℓ *qubits.*

Now we move to our construction. Let n be the number of parties and Γ be the access structure. We will assume access to the following objects:

- A secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ for the access structure Γ, mapping a u-bit message m to w-bit shares $Z_1^m, \ldots, Z_n^m$ with $\varepsilon_{\mathsf{priv}}$-privacy, i.e.,

  $$(Z_i^m)_{i\in T} \approx_{\varepsilon_{\mathsf{priv}}} (Z_i^{m'})_{i\in T}$$

  for any unauthorized set $T \notin \Gamma$ and any two secrets m and m'. We additionally enforce the marginal uniformity property that $Z_i^m \approx_{\varepsilon_{\mathsf{unif}}} U_w$ for all $i \in [n]$ and $m \in \{0,1\}^u$. We also require that the access structure $\Gamma \subseteq 2^{[n]}$ realized by $(\mathsf{Share}, \mathsf{Rec})$ contains no singletons.[22] For the special case of threshold access structures, Shamir's secret sharing scheme satisfies these properties with $\varepsilon_{\mathsf{priv}} = \varepsilon_{\mathsf{unif}} = 0$.

- An explicit linear $(k = C_{BB84} \cdot N - \ell, \varepsilon_{\mathsf{ext}})$-strong quantum-proof seeded extractor $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^w$, such as Trevisan's extractor from Lemma 20 with seed length $d = O(\log^3(w/\varepsilon_{\mathsf{ext}}))$ and such that

  $$w \geq k - O(d) = C_{BB84} \cdot N - \ell - O(d).$$

  As already shown in [CKOS22, Lemma 2], every such linear extractor Ext is equipped with an efficient inversion procedure $\mathsf{InvExt}(z, s)$ which either samples x uniformly at random from the preimage $\{x \in \{0,1\}^N : \mathsf{Ext}(x, s) = z\}$ or outputs $\perp$ if this set is empty. If $S \leftarrow \{0,1\}^d$ and $Z \approx_{\varepsilon_{\mathsf{unif}}} U_m$ are independent, it holds that

  $$\Pr[\mathsf{InvExt}(Z, S) = \perp] \leq \varepsilon_{\mathsf{ext}} + \varepsilon_{\mathsf{unif}}. \tag{6}$$

  To see this, note that $\mathsf{InvExt}(Z, S) \approx_{\varepsilon_{\mathsf{unif}}} \mathsf{InvExt}(U_m, U_d)$, and that at most an $\varepsilon_{\mathsf{ext}}$-fraction of output-seed pairs $(z, s) \in \{0,1\}^w \times \{0,1\}^d$ can have an empty preimage with respect to Ext. Then, a union bound yields Equation (6).

- 2-out-of-n Shamir secret sharing schemes $(\mathsf{Share}_{2\text{-}n}, \mathsf{Rec}_{2\text{-}n})$ for N-bit and d-bit secrets.[23]

We construct a leakage-resilient secret sharing scheme $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$ realizing Γ using the objects above. On input a secret $m \in \{0,1\}^u$, the sharing procedure $\mathsf{Share}^\star$ proceeds as follows:

1. Compute $(Z_1, \ldots, Z_n) \leftarrow \mathsf{Share}(m)$.

[22] Note that locally leakage-resilient secret sharing is unachievable over any access structure with singletons.

[23] For the sake of simplicity, we avoid parameterizing these schemes by the secret length.

2. Sample a basis $\theta \leftarrow \{0,1\}^N$ and a seed $S \leftarrow \{0,1\}^d$. Compute the 2-out-of-n Shamir shares $(\theta_1, \ldots, \theta_n) \leftarrow \mathsf{Share}_{2\text{-}n}(\theta)$ and $(S_1, \ldots, S_n) \leftarrow \mathsf{Share}_{2\text{-}n}(S)$.

3. For each $i \in [n]$, sample $X_i \leftarrow \mathsf{InvExt}(Z_i, S)$.

4. If $X_i \neq \perp$, set $\mathsf{Sh}_i = (H^\theta|X_i\rangle, S_i, \theta_i)$. Else, if $X_i = \perp$ set $\mathsf{Sh}_i = (\perp, Z_i)$.

The reconstruction procedure $\mathsf{Rec}^\star$ is straightforward. Moreover, it is easy to show that $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$ realizes Γ and satisfies $\varepsilon_{\mathsf{priv}}$-privacy. To conclude the argument, we proceed to show that $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$ is resilient to local unbounded classical leakage.

Theorem 53. *The secret sharing scheme* $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$ *for the access structure* Γ *is* $\varepsilon_{\mathsf{leak}}$*-unbounded-leakage-resilient with*

$$\varepsilon_{\mathsf{leak}} = 5n(\varepsilon_{\mathsf{ext}} + \varepsilon_{\mathsf{unif}}) + \varepsilon_{\mathsf{priv}}.$$

Proof. We prove Theorem 53 via a hybrid argument. Fix an unauthorized set $T \notin \Gamma$ of size t. Without loss of generality we may assume that $T = \{1, \ldots, t\}$. For a secret $m \in \{0,1\}^u$ and local quantum-to-classical leakage functions

$$\mathsf{Leak}_{t+1}, \ldots, \mathsf{Leak}_n,$$

let Leak^m denote the output of the leakage experiment on m, i.e.,

$$\mathsf{Leak}^m = (\mathsf{Sh}_i)_{i\in[t]}, (\mathsf{Leak}_j(\mathsf{Sh}_j))_{j\in\{t+1,\ldots,n\}},$$

where $(\mathsf{Sh}_1, \ldots, \mathsf{Sh}_n) \leftarrow \mathsf{Share}^\star(m)$. The desired result follows if we show that $\mathsf{Leak}^m \approx_{\varepsilon_{\mathsf{leak}}} \mathsf{Leak}^{m'}$ for any two secrets $m, m' \in \{0,1\}^u$. By Equation (6) and a union bound over all n shares, it follows that the probability that there is at least one share of the form $(\perp, Z_i)$ is at most

$$n(\varepsilon_{\mathsf{ext}} + \varepsilon_{\mathsf{unif}}).$$

Consequently, from here onwards we assume that no inversion procedure fails in the sharing phase, and will add this term to the final leakage error $\varepsilon_{\mathsf{leak}}$.

Towards this end, we consider hybrids Hyb_i^m for $i = t, \ldots, n$ which behave like Leak^m, but where $X_j \leftarrow \{0,1\}^N$ for every $j \in \{t+1, \ldots, i\}$. Note that $\mathsf{Leak}^m \equiv \mathsf{Hyb}_t^m$ and, by $\varepsilon_{\mathsf{priv}}$-privacy of the underlying scheme $(\mathsf{Share}, \mathsf{Rec})$, we also have

$$\mathsf{Hyb}_n^m \approx_{\varepsilon_{\mathsf{priv}}} \mathsf{Hyb}_n^{m'}.$$

Therefore, it suffices to establish the following.

Claim 9. *For every secret* $m \in \{0,1\}^u$ *and* $i \in \{t+1, \ldots, n\}$ *it holds that*

$$\mathsf{Hyb}_{i-1}^m \approx_{2(\varepsilon_{\mathsf{ext}}+\varepsilon_{\mathsf{unif}})} \mathsf{Hyb}_i^m.$$

Assuming Claim 9, repeated application of the triangle inequality yields

$$\mathsf{Leak}^m \equiv \mathsf{Hyb}_0^m \approx_{2n(\varepsilon_{\mathsf{ext}}+\varepsilon_{\mathsf{unif}})} \mathsf{Hyb}_{n-t}^m \approx_{\varepsilon_{\mathsf{priv}}} \mathsf{Hyb}_{n-t}^{m'} \approx_{2n(\varepsilon_{\mathsf{ext}}+\varepsilon_{\mathsf{unif}})} \mathsf{Hyb}_0^{m'} \equiv \mathsf{Leak}^{m'}.$$

The triangle inequality applied to this chain leads to Theorem 53. We proceed to prove Claim 9, which concludes our argument.

Proof of Claim 9. Note that Hyb_{i-1}^m and Hyb_i^m only differ in the computation of the i-th share Sh_i. We begin by observing that we may write Hyb_{i-1}^m and Hyb_i^m as

$$\mathsf{Hyb}_{i-1}^m = f(i, Z_i, S_i, S, \theta_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(Z_i, S)\rangle, S_i, \theta_i))$$

and

$$\mathsf{Hyb}_i^m = f(i, Z_i, S_i, S, \theta_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, S_i, \theta_i))$$

for the same randomized function f (with randomness independent of the input). Therefore, by the post-processing property of trace distance, it suffices to show that

$$Z_i, S_i, S, \theta_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(Z_i, S)\rangle, S_i, \theta_i) \approx_{2(\varepsilon_{\mathsf{ext}}+\varepsilon_{\mathsf{unif}})} Z_i, S_i, S, \theta_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, S_i, \theta_i). \quad (7)$$

We claim that we can replace the Shamir shares S_i and θ_i by Shamir shares of 0 and Z_i by the uniform distribution on $\{0,1\}^w$. Let $\widetilde{S}_i$ and $\widetilde{\theta}_i$ denote the i-th Shamir secret sharing of 0. Since $Z_i \approx_{\varepsilon_{\mathsf{unif}}} U_w$ and Z_i and X are independent of each other and of S_i, S, θ_i, θ, the 0-privacy of Shamir secret sharing implies that

$$Z_i, S_i, S, \theta_i, \theta, X \approx_{\varepsilon_{\mathsf{unif}}} U_w, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, X.$$

Since both sides of Equation (7) are randomized functions of $Z_i, S_i, S, \theta_i, \theta, X$, in order to show Equation (7) it is enough to argue that

$$U_w, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(U_m, S)\rangle, \widetilde{S}_i, \widetilde{\theta}_i) \approx_{2\varepsilon_{\mathsf{ext}}} U_w, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, \widetilde{S}_i, \widetilde{\theta}_i). \quad (8)$$

As $\widetilde{S}_i$ and $\widetilde{\theta}_i$ are independent of X and θ, Lemma 22 guarantees that

$$\mathbf{H}_\infty(X|\mathsf{Leak}_i(H^\theta|X\rangle, \widetilde{S}_i, \widetilde{\theta}_i), \widetilde{\theta}_i, \theta) \geq C_{BB84} \cdot N.$$

Therefore, invoking the strong extractor properties of Ext and the fact that $\widetilde{S}_i$, $\widetilde{\theta}_i$, and θ are independent of X and the seed S yields

$$\begin{aligned}
&U_m, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, \widetilde{S}_i, \widetilde{\theta}_i) \\
&\approx_{\varepsilon_{\mathsf{ext}}} \mathsf{Ext}(X, S), \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, \widetilde{S}_i, \widetilde{\theta}_i) \\
&\equiv \mathsf{Ext}(X, S), \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(\mathsf{Ext}(X, S), S)\rangle, \widetilde{S}_i, \widetilde{\theta}_i) \\
&\approx_{\varepsilon_{\mathsf{ext}}} U_m, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(U_m, S)\rangle, \widetilde{S}_i, \widetilde{\theta}_i),
\end{aligned}$$

and so Equation (8) follows by the triangle inequality. □

□

Setting parameters in the compiler

In this section we show how to instantiate the compiler from Theorem 53 to obtain efficient threshold secret sharing schemes resilient against unbounded classical leakage and a constant rate of quantum leakage with exponentially small error. To be more precise, we obtain the following corollary.

Corollary 11. *Given a security parameter λ, there is an efficient t-out-of-n secret sharing scheme for u-bit secrets with the following properties:*

- *Its share length $w^\star$ satisfies $w^\star = O(u + \lambda^3)$;*

- *It is* $(\varepsilon_{\mathsf{leak}} = O(n2^{-\lambda}))$*-unbounded-leakage-resilient to* $(*, \ell)$*-leakage for* $\ell = \Omega(w^\star)$ *qubits of leakage from each share.*

Proof. Let λ be a security parameter. Our goal is to instantiate the compiler so that the resulting scheme $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$ achieves leakage error $\varepsilon_{\mathsf{leak}} = O(n2^{-\lambda})$, where n is the number of parties, while withstanding unbounded classical leakage and $\ell = \Omega(w^\star)$ qubits of leakage from each share, where $w^\star$ denotes the share length of $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$, and so that $w^\star$ is not much larger than the original share size w of the underlying (non-leakage-resilient) secret sharing scheme.

Choose the underlying scheme $(\mathsf{Share}, \mathsf{Rec})$ to be a t-out-of-n Shamir secret sharing scheme with secret size u and share size $w = u$. Note that $(\mathsf{Share}, \mathsf{Rec})$ satisfies $(\varepsilon_{\mathsf{priv}} = 0)$-privacy and $(\varepsilon_{\mathsf{unif}} = 0)$-uniformity. Lemma 20 guarantees the existence of an efficient linear $(k, \varepsilon_{\mathsf{ext}})$-strong quantum-proof seeded extractor $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^w$ with error $\varepsilon_{\mathsf{ext}} = 2^{-\lambda}$, input source length $N = C(w + \lambda^3)$ for a sufficiently large constant $C > 0$, min-entropy requirement $k = cN$ for an arbitrary constant $c > 0$, and seed length $d \leq C' \log^3(N/\varepsilon_{\mathsf{ext}}) = C'(\lambda^3 + \log^3 N)$ for a sufficiently large constant $C' > 0$.

Combining the objects above with the compiler of Theorem 53 yields an efficient threshold secret sharing scheme $(\mathsf{Share}^\star, \mathsf{Rec}^\star)$ with share size $w^\star = 2N + d = O(w + \lambda^3)$. It remains to argue that we may set $\ell = \Omega(w^\star)$ and $\varepsilon_{\mathsf{leak}} = O(n2^{-\lambda})$. Note that under these constraints we may assume that $C_{BB84}N - \ell \geq cN$ for some constant $c > 0$, thus satisfying the min-entropy requirement of the extractor Ext above, and obtaining final leakage error

$$\varepsilon_{\mathsf{leak}} = 5n(\varepsilon_{\mathsf{ext}} + \varepsilon_{\mathsf{unif}}) + \varepsilon_{\mathsf{priv}} = O(n2^{-\lambda}).$$

□

B.3.3 An impossibility result for leakage-resilient secret sharing

We have designed secret sharing schemes which are resilient against unbounded classical leakage. To complement this result, relying on ideas from a recent result of Ananth, Goyal, Liu and Liu [AGLL23], we show that such schemes are unachievable if we additionally allow arbitrary entangled states to be shared between local leakage adversaries, even if these adversaries only output classical leakage and share no entanglement with the distinguisher. For simplicity, we present the result for 2-out-of-2 threshold secret sharing schemes, but the argument is easily generalizable to quantum secret sharing schemes realizing arbitrary access structures.

Theorem 54. *Given any quantum secret sharing scheme which encodes a secret* $m \in \{0,1\}$ *into* w*-dimensional shares* $\mathsf{Sh}^m = (\mathsf{Sh}_1^m, \mathsf{Sh}_2^m)$*, there exist quantum-to-classical local leakage functions* Leak_1 *and* Leak_2 *outputting* $\ell_c = \ell_c(w)$ *classical bits each, for* N *and* ℓ_c *sufficiently large functions of* w*, and a distinguisher* $\mathcal{D}$ *such that*

$$\Pr\left[\mathcal{D}(\mathsf{Sh}_i^1, R_i, \mathsf{Leak}_{3-i}(\mathsf{Sh}_{3-i}^1, R_{3-i})) = 1\right] - \Pr\left[\mathcal{D}(\mathsf{Sh}_i^0, R_i, \mathsf{Leak}_{3-i}(\mathsf{Sh}_{3-i}^0, R_{3-i})) = 1\right] \geq 0.99$$

for any $i \in \{1,2\}$*, where* R_1, R_2 *is initialized to* $N = N(w)$ *EPR pairs shared between them.*

Before we proceed to the proof of Theorem 54, we introduce port-based teleportation.

Port-based teleportation Port-Based Teleportation (PBT), introduced by Ishizaka and Hiroshima [IH08, IH09] is a quantum teleportation protocol between two parties, Alice and Bob, with special properties. More precisely, assuming that Alice and Bob share a large number N of EPR

pairs[24], Alice can teleport a d-dimensional quantum state to Bob by performing a joint measurement and communicating its outcome (determining which EPR pairs contain the teleported state) to Bob, who does not perform any operation. PBT necessarily incurs some failure probability or non-perfect fidelity between the original and the teleported states. In contrast, vanilla quantum teleportation has no failure probability and has perfect fidelity, but requires Bob to perform some corrective operations to its state.

In the *probabilistic* version of PBT which we will be using, the protocol may fail with some probability $p(d, N)$, and otherwise simulates an identity channel perfectly. It is known that $p(d, N) \to 0$ as $N \to \infty$ for every fixed dimension d. In fact, the asymptotics of this failure probability are well studied [CLM+21], although we will not need them here.

This discussion is summarized in the following theorem.

Theorem 55 (Probabilistic PBT [IH08, CLM+21])**.** *Fix a dimension $d > 0$. Suppose that Alice and Bob share N EPR pairs indexed in some prespecified manner. There exists a protocol between Alice and Bob through which Alice can teleport a d-dimensional quantum state to Bob by performing a measurement and sending its classical outcome i to Bob. To obtain the teleported state, Bob does not apply any operations to its state and simply selects its EPR halves indexed by the received measurement outcome i. The protocol fails with some probability $p(d, N)$ which satisfies $p(d, N) \to 0$ as $N \to \infty$, and otherwise perfectly simulates an identity channel.*

Proof of Theorem 54

Suppose that a secret $m \in \{0, 1\}$ is secret-shared into w-dimensional shares $\mathsf{Sh}^m = (\mathsf{Sh}_1^m, \mathsf{Sh}_2^m)$. Consider the local leakage functions $\mathsf{Leak}_1, \mathsf{Leak}_2$ sharing N EPR pairs and where Leak_i has access to Sh_i^m defined as follows:

1. Using the halves of their first w shared EPR pairs, Leak_1 teleports Sh_1^m to Leak_2. Let $k, k' \in \{0, 1\}^w$ denote the measurement outcome of the quantum teleportation protocol, so that the halves of the w EPR pairs held by Leak_2 now contain the state
$$\overline{\mathsf{Sh}_1}^m = \left(X^{k_i} Z^{k'_i} (\mathsf{Sh}_1^m)_i\right)_{i \in [w]}$$

2. Leak_2 now has access to Sh_2^m and the hidden share $\overline{\mathsf{Sh}_1}^m$. Exploiting probabilistic PBT (see Appendix B.3.3), Leak_2 teleports $(\overline{\mathsf{Sh}_1}^m, \mathsf{Sh}_2^m)$ to Leak_1 using the remaining EPR pairs. Let $i^\star$ denote the measurement outcome of the PBT protocol.

3. Using the measurement outcomes (k, k') from the initial teleportation step, Leak_1 applies $\bigotimes_{i=1}^w X^{k_i} Z^{k'_i}$ to the EPR halves of each port corresponding to $\overline{\mathsf{Sh}_1}^m$, and then applies the reconstruction algorithm of the given quantum secret sharing scheme to the EPR halves corresponding to each port. Finally, Leak_1 leaks the classical output of the reconstruction algorithm on each port.

4. Leak_2 leaks the PBT measurement outcome $i^\star$.

Conditioned on the probabilistic PBT protocol succeeding, the EPR halves held by Leak_1 corresponding to port $i^\star$ contain the state $(\overline{\mathsf{Sh}_1}^m, \mathsf{Sh}_2^m)$. Therefore, the output of Leak_1's operations on port $i^\star$, call it $L_{i^\star}$, satisfies $L_{i^\star} = m$. By Theorem 55, if the number N of EPR pairs shared

[24] Here we focus on the setting where Alice and Bob share EPR pairs. Settings where Alice and Bob share entangled states optimized for PBT have also been studied. See, e.g., [IH08, IH09, CLM+21].

by Leak_1 and Leak_2 is large enough then it holds that the probabilistic PBT protocol succeeds with probability at least 0.99. As a result, the distinguisher $\mathcal{D}$ which outputs $L_{i^\star}$, which can be computed given the classical outputs of Leak_1 and Leak_2, succeeds with the desired advantage.