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

Alper Çakan
Carnegie Mellon University
acakan@andrew.cmu.edu

Vipul Goyal
NTT Research & CMU
vipul@cmu.edu

Chen-Da Liu-Zhang
NTT Research
chen-da.liuzhang@ntt-research.com

João Ribeiro*
NOVA LINCS
Universidade Nova de Lisboa
joao.ribeiro@fct.unl.pt

Abstract

Side-channel attacks, which aim to leak side information on secret system components, are ubiquitous. Even simple attacks, such as measuring time elapsed or radiation emitted during encryption and decryption procedures, completely break textbook versions of many cryptographic schemes. This has prompted the study of *leakage-resilient* cryptography, which remains secure in the presence of side-channel attacks.

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, it is unclear if such a bound is realistic in practice since many practical side-channel attacks cannot be captured by bounded leakage.

In this work, we investigate the possibility of designing cryptographic schemes that provide guarantees against *arbitrary* side-channel attacks:

- Using techniques from uncloneable quantum cryptography, we design several basic leakage-resilient primitives, such as secret sharing, (weak) pseudorandom functions, digital signatures, and public- and private-key encryption, 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.
- In the even stronger adversarial setting where the adversary is allowed to obtain unbounded *quantum* leakage (and thus leakage-resilience is impossible), we design schemes for many cryptographic tasks which support *leakage-detection*. This means that we can efficiently check whether the security of such a scheme has been compromised by a side-channel attack. These schemes are based on techniques from cryptography with certified deletion.
- We also initiate a study of *classical* cryptographic schemes with (bounded) *post-quantum* leakage-resilience. These schemes resist side-channel attacks performed by adversaries with quantum capabilities which may even share arbitrary *entangled* quantum states. That is, even if such adversaries are non-communicating, they can still have "spooky" communication via entangled states.

*Part of the work was done while at Carnegie Mellon University.

Contents

1 Introduction

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. 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 ℓ 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. Quantum information behaves in a fundamentally different way compared to its classical counterpart. In particular, 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. In particular, we design a host of cryptographic schemes such as public-key encryption, digital signatures, (weak) pseudorandom functions, and secret sharing schemes which *can tolerate any (polynomially) unbounded classical leakage*. In our view, leakage is the result of *observations* of quantities such as power consumption, time elapsed, and temperature fluctuations. Hence, leakage is most naturally viewed as classical information. If this view is indeed correct, *our schemes can even be seen as leakage-proof rather than just leakage-resilient*. Many of our results are obtained by using techniques from uncloneable cryptography.

Leakage-detection in a quantum world. What if an adversary can get quantum leakage after all? We show that most of our constructions can tolerate a bounded amount of quantum leakage (in addition to unbounded classical leakage). However, if an adversary can get unbounded quantum leakage, it can simply leak the whole secret and, similarly to the classical setting, all bets are off. This raises the following question:

> Can we still achieve meaningful security guarantees in the face of unbounded quantum leakage?

We design a host of cryptographic scheme which can offer *leakage-detection* in this case. While leakage-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 and considered an 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 however, we are able to design public-key encryption and digital signature schemes supporting leakage-detection. More precisely, our schemes provide the following guarantee: Suppose that an adversary was able to leak sufficient (quantum or classical) information to break the security of the primitive (e.g., break indistinguishability in the case of public-key encryption). Then, there exists a procedure called TestLeakage which takes the residual secret (e.g., the residual secret key in the case of public-key encryption), and outputs LEAKED with overwhelming probability, indicating that a leakage attack was performed and security has been compromised. On the other hand, if the procedure TestLeakage outputs NO LEAKAGE, then, with overwhelming probability, *either there has been either no leakage attack, or any possible leakage 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 certified deletion.

In the quantum setting, the boundary between leakage and tampering is blurred and *our results, in fact, also offer security against tampering attacks.* In particular, we allow the adversary to obtain arbitrary (quantum) leakage, tamper with any non-public secret and even then, if the security has been compromised, our TestLeakage will output LEAKED. For example in case of public key encryption, we allow the adversary to arbitrarily tamper with the secret key before the TestLeakage algorithm is run.

Classical schemes with (bounded) post-quantum leakage resilience. In the settings above we (necessarily) considered quantum schemes. However, classical schemes are friendlier to everyday users, and it is thus interesting to investigate whether one can design classical schemes which tolerate either classical or even quantum leakage by quantum adversaries in the post-quantum setting. As a first natural question:

Is a classical leakage-resilient scheme automatically "post-quantum leakage-resilient" if it is based on post-quantum assumptions?

We argue that, unfortunately, the above is not true, as adversaries with access to quantum computers may be able to carry out more devastating leakage attacks. Notably, non-communicating parties with quantum capabilities can make use of shared *entangled* quantum states (even if they are physically far apart from each other, and even if they only output classical bits) to obtain a strict advantage over fully classical parties in many settings. One early example of this phenomenon is the CHSH game [CHSH69], where a small amount of entanglement allows for a higher success probability versus non-communicating classical parties with arbitrary correlated classical randomness.

For example, in the context of public-key encryption one can consider a leakage adversary and a "main adversary" which tries to break the indistinguishability given the ciphertext and the output of the leakage adversary. If these adversaries can share unbounded entanglement, then the leakage adversary can teleport the whole secret to the main adversary without any communication. Thus, here one must rely on the inability of the leakage adversary to communicate the full set of classical correction bits to the main adversary. In fact, one can consider settings where many local leakage adversaries now share entangled quantum states, for example in the context of locally

leakage-resilient secret sharing [BDIR18, GK18, ADN+19, SV19, CKOS22]. While in the classical setting such adversaries cannot communicate, in the post-quantum setting, *they can have "spooky communication" via shared entanglement.*

Our main contribution in this direction is to again design schemes for several cryptographic tasks such as secret sharing, pseudorandom functions, public-key encryption, digital signatures, and general computation which are resilient to bounded post-quantum leakage with shared entanglement.

1.1 Our results

1.1.1 Leakage-resilience against unbounded classical leakage

We first show quantum schemes that tolerate any arbitrary (polynomial) amount of classical leakage, for a wide range of primitives including secret sharing, PRFs, MACs, signatures, public-key encryption and private-key encryption. Details can be found in Section 3.

Secret sharing. We consider the model for secret sharing where the adversary can obtain from each share any unbounded classical leakage. Our first result is an efficient threshold secret sharing scheme resilient against unbounded classical leakage and in addition a constant rate of quantum leakage.

Theorem 1 (informal)**.** *Given a security parameter λ, there is an efficient t-out-of-n secret sharing scheme for u-bit secrets with share length $w^\star = O(u+\lambda^3)$ that tolerates unbounded classical leakage, and $\ell = \Omega(w^\star)$ qubits of leakage from each share.*

To complement our result, we show that such schemes are unachievable if we additionally allow arbitrary entangled states to be shared between local leakage adversaries, even if we these adversaries only output classical leakage.

Theorem 2 (informal)**.** *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 exists a quantum-to-classical local leakage functions Leak_1 and Leak_2 sharing $N = N(w)$ EPR pairs and outputting $\ell_c = \ell_c(w)$ classical bits each, that breaks the security of the sharing scheme.*

We then move to basic cryptographic primitives, including PRFs, MACs, digital signatures, public-key encryption and private-key encryption. These results are obtained using techniques from the area of uncloneable and copy-protection cryptography [CLLZ21, LLQZ22].

Basic cryptographic primitives. We first consider the notion of leakage-resilient schemes that grant security even when the adversary can leak unbounded classical leakage for various basic cryptographic primitives, including weak pseudorandom functions, message authentication codes, digital signatures, public-key encryption and secret-key encryption. We consider a main adversary that participates in the respective security game, but can in addition obtain an unbounded classical leakage computed by a leakage adversary on the respective keys.

Theorem 3 (informal)**.** *Assuming the existence of post-quantum sub-exponentially secure iO and one-way functions, and the quantum hardness of LWE, there exists unbounded-classical-leakage-resilient schemes for weak PRFs, weak MACs, digital signatures, public-key encryption and secret-key encryption.*

Moreover, we also show that we can obtain secret-key encryption schemes from public-key assumptions.

Theorem 4 (informal)**.** *Assuming the existence of a post-quantum public-key encryption, there exists an unbounded-classical-leakage-resilient scheme for secret-key encryption.*

1.1.2 Leakage-detection against arbitrary unbounded quantum leakage

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 leakage-detection. More specifically, we aim to design cryptographic primitives with a leakage-detection algorithm, which can detect if a *useful* leakage has been obtained on the secret key. We show that this is possible for public-key encryption and digital signatures.

Theorem 5. *Suppose there exists a {public-key encryption, digital signature, functional encryption, obfuscation, secure software leasing} scheme with certified deletion. Then, there exists a {public-key encryption, digital signature, functional encryption, obfuscation, software protection} scheme with leakage-detection.*

1.1.3 Classical schemes with bounded post-quantum leakage resilience

The schemes from previous sections are (inherently) quantum schemes, so we then turn our attention to classical schemes that tolerate classical leakage in the post-quantum setting.

Basic cryptographic primitives. We investigate several cryptographic primitives, including weak PRFs, PKE and digital signatures, in the setting where there is a main adversary (attempting to win the corresponding PRF/PKE/signature security game), obtains ℓ bits of leakage from a leakage adversary. Both adversaries may share arbitrary entanglement. We denote schemes resilient against such adversaries as ℓ-spooky-classic-leakage resilient schemes. See Section 5.1 for details.

Known constructions for weak PRFs, PKE and signatures [HLAWW16, KV09, FKPR10] can be proven secure in this setting, by making use of a new min-entropy drop lemma.

Theorem 6 (informal)**.** *Assuming the existence of post-quantum one-way functions, for any polynomial $\ell(\cdot)$, there exists $\ell(\lambda)$-spooky-classic-LR wPRFs.*

Theorem 7 (informal)**.** *Assuming the existence of post-quantum public-key encryption schemes, for any polynomial $\ell(\cdot)$, there exists a $\ell(\lambda)$-spooky-classic-LR public-key encryption scheme.*

Theorem 8 (informal)**.** *Let n be the size of the secret key. Assuming post-quantum secure universal one-way hash functions, there exists:*

- *A one-time signature scheme that tolerates $(1/4-\epsilon)n$ spooky-classical-leakage;*
- *A t-time signature scheme that tolerates $\theta(n/t^2)$ spooky-classical-leakage;*
- *A stateful signature scheme that tolerates $\theta(n/9)$ spooky-classical-leakage.*

Secret Sharing. We then study secret sharing schemes with *spooky local leakage*, where each local leakage adversary $\mathcal{A}_i$ can leak ℓ bits from each share S_i, and the adversaries may be arbitrarily entangled with each other. See Section 5.2 for details.

In the classical setting we know schemes with shares of size N which tolerate $\ell = (1-\delta)N$ bits of local leakage per share, for any constant $\delta > 0$ [ADN+19, SV19, CKOS22]. We first show that spooky leakage-resilience is impossible if the leakage rate is at least $1/2$, via superdense coding.

Theorem 9 (informal)**.** *If there exists an (ℓ, ε)-spooky locally leakage-resilient secret sharing scheme with share space $\{0,1\}^N$ and error $\varepsilon < 1$, then $\ell < N/2$. Moreover, the adversary can guess the secret with probability δ whenever $\ell \geq \frac{1}{2}(N - \log(1/\delta))$.*

We then construct a simple and nearly optimal 2-out-of-2 spooky leakage-resilient secret sharing scheme via quantum-proof two-source extractors.

Theorem 10 (informal)**.** *Let $\ell \geq 0$. There exists an efficient 2-out-of-2 ℓ-spooky leakage-resilient secret sharing scheme which shares one bit into two shares of size $N = 2(\ell + \log(1/\varepsilon) - 1)$. In other words, this scheme withstands local leakage of $\ell = N/2 + 1 - \log(1/\varepsilon)$ qubits from each share.*

General leakage-resilient computations. Finally, we show that the compiler by Goldwasser and Rothblum [GR12] also works if the leakage functions share arbitrary entanglement. More concretely, we show that the compiler transforms a general computation into an algorithm that withstands spooky leakage attacks.

The leakage model considered is the *only computation leaks* model [MR04], where the algorithm is composed of a sequence of instructions, which are basic subcomputations coming from a fixed universal set of instructions. The adversary is allowed to learn a bounded local classical leakage on each operand to an instruction when it is executed, and the leakage functions are decided in advance and have access to arbitrary entanglement.

Theorem 11 (informal)**.** *There exists a compiler and a leakage bound function $L = \Theta(\lambda)$ such that for every λ, the compiler outputs an algorithm that tolerates $L(\lambda)$ spooky-leakage.*

1.2 Technical overview

A common theme across several of our results is that they are based on techniques from quantum copy-protection and uncloneability. The connection between leakage resilience and copy-protection is as follows. Suppose one can obtain classical leakage on a (quantum) secret which is "functionally equivalent" to the secret itself (e.g., this leakage allows one to decrypt in case the secret is a secret key). But then, since any classical information can be cloned, this gives us a way of essentially obtaining multiple states having the same functionality as the quantum secret. If the quantum secret was "uncloneable", we arrive at a contradiction. While this basic observation is our starting point, this is not enough due to our limited understanding of quantum uncloneable cryptography. For example, we are not aware of any constructions of uncloneable secret sharing in the plain model. Additionally even for primitives such as PRF or public-key encryption where an uncloneable counterpart exists, new ideas are required to get a leakage-resilient construction. We provide a more detailed overview of a selected subset of our results.

1.2.1 Cryptography resilient to unbounded classical leakage

Leakage-resilient secret sharing via random Wiesner encodings. In this section we provide an overview of our cryptographic schemes resilient to unbounded classical leakage. For the sake

of exposition, we construct here a more modest 2-out-of-2 secret sharing scheme for 1-bit messages which is only resilient to unbounded classical leakage on a fixed share. This allows us to illustrate some of our main underlying ideas. Then, we briefly discuss how we can extend this approach to construct secret sharing schemes realizing a very general classe of access structures resilient to unbounded classical leakage on all shares.

More precisely, our more modest goal is to construct a "one-sided leakage-resilient" secret sharing scheme which maps a secret $m \in \{0,1\}$ into two possibly quantum shares $(\mathsf{Sh}_1^m, \mathsf{Sh}_2^m)$ such that (1) we can perfectly reconstruct m given $(\mathsf{Sh}_1^m, \mathsf{Sh}_2^m)$, and (2) for any quantum-to-classical leakage function Leak chosen a priori it holds that

$$\mathsf{Sh}_1^0, \mathsf{Leak}(\mathsf{Sh}_2^0) \approx \mathsf{Sh}_1^1, \mathsf{Leak}(\mathsf{Sh}_2^1), \tag{1}$$

where $\approx$ means that these distributions are appropriately close in statistical distance. By "one-sided", we mean that if the adversary instead gets leakage on Sh_1^0 and Sh_2^0 in the clear, the scheme becomes insecure.

The main tool we employ towards this goal are random *Wiesner encodings* [Wie83]: The *Wiesner encoding* of a classical bitstring X is given by

$$\rho_{X,\theta} = H^\theta|X\rangle = H^{\theta_1}|X_1\rangle \otimes H^{\theta_2}|X_2\rangle \otimes \cdots \otimes H^{\theta_n}|X_n\rangle$$

for a uniformly random string $\theta \leftarrow \{0,1\}^n$. In words, $\rho_{X,\theta}$ is obtained by independently encoding each bit of X in either the computational or Hadamard basis (represented by θ) with probability $1/2$. This encoding has been used in many other contexts within quantum cryptography, including recent examples such as uncloneable encryption [BL20] and cryptography with certified deletion [BK22].

The key property we require from the random Wiesner encoding $\rho_{X,\theta}$ is that X is hard to guess even given the basis θ and any unbounded classical leakage $\mathsf{Leak}(\rho_{X,\theta})$. More precisely, if $X \leftarrow \{0,1\}^n$, then

$$\mathbf{H}_\infty(X|\theta, \mathsf{Leak}(\rho_{X,\theta})) > 0.2n, \tag{2}$$

where $\mathbf{H}_\infty(\cdot|\cdot)$ denotes the average conditional min-entropy. We show this via a connection to Monogamy-of-Entanglement (MoE) games [TFKW13], and, in particular, the BB84 game played by three parties, Alice, Bob, and Charlie: Alice holds n qubits, and Bob and Charlie hold quantum registers arbitrarily entangled with Alice's register. Alice measures each qubit according to the computational or Hadamard basis uniformly at random, yielding an outcome X, and sends the measurement basis vector θ to both Bob and Charlie. Then, Bob and Charlie win the game if they *both* guess X. Tomamichel, Fehr, Kaniewski, and Wehner [TFKW13] showed that the optimal success probability in this game is much smaller than $2^{-0.2n}$. But, note that the task of guessing X given $(\theta, \mathsf{Leak}(\rho_{X,\theta}))$ is a particular strategy in the BB84 game, since $(\theta, \mathsf{Leak}(\rho_{X,\theta}))$ is a classical string and thus can be cloned between Bob and Charlie! This yields Equation (2).

The above leads us to consider the candidate secret sharing scheme

$$\begin{aligned}\mathsf{Sh}_1^m &= (S, \theta, m + \mathsf{Ext}(X,S)),\\ \mathsf{Sh}_2^m &= \rho_{X,\theta},\end{aligned}$$

where $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ is an average-case strong seeded extractor for min-entropy $0.2n$, $S \leftarrow \{0,1\}^d$ is the seed, and $X \leftarrow \{0,1\}^n$ is the extractor's source. To see why this scheme satisfies Equation (1), note that X and S remain independent after revealing $\mathsf{Leak}(\mathsf{Sh}_2^m) = \mathsf{Leak}(\rho_{X,\theta})$, and that, by Equation (2), X retains enough min-entropy so that $\mathsf{Ext}(X,S)$ is close to uniform even given S, θ, and $\mathsf{Leak}(\rho_{X,\theta})$. This ensures that the secret m remains hidden.

In the study of uncloneable cryptography, going from unpredictability to indistinguishability has proven to be a hard problem. For example, the above style of randomness extraction argument requires the use of random oracles in the setting of uncloneable encryption [AKL+22]. Fortunately, we are able to make it work in the plain model by relying on the fact that our "cloned states" are classical.

We also take these ideas further and design secret sharing schemes realizing a large class of access structures and withstanding unbounded classical leakage on *all* shares. Roughly speaking, we combine the approach above using random Wiesner encodings with ideas from the classical compiler of [CKOS22] which transforms a (non-leakage-resilient) secret sharing scheme satisfying mild properties into a leakage-resilient secret sharing scheme for the same access structure. In particular, using Shamir secret sharing as the base scheme, we construct secret sharing schemes for any threshold access structure resilient to unbounded classical leakage. For more details, see Section 3.1.

Simultaneously handling arbitrary classical leakage attacks and bounded quantum leakage. We show that our schemes against unbounded classical leakage also withstand a combination of unbounded classical leakage and a linear amount of quantum leakage. This is enabled by the fact that the Wiesner encoding leakage min-entropy bound from Equation (2) can be extended to this combined setting separating the classical leakage and the quantum leakage, via a chain rule, yielding

$$\mathbf{H}_\infty(X|\theta, \mathsf{Leak}(H^\theta|X\rangle)) \geq 0.2n - \ell,$$

where ℓ denotes the number of *qubits* leaked by Leak. Note, however, that standard strong seeded extractors are not guaranteed to be secure against bounded quantum leakage. Nevertheless, it is well known that Trevisan's extractor, which we use in our schemes, remains secure even against quantum side information [DPVR12].

Impossibility of LRSS with Unbounded Shared Entanglement. The above positive result on secret sharing resilient to unbounded classical leakage on all shares holds assuming that the different local leakage adversaries targetting different shares do not have access to shared entanglement. However, say that these adversaries *do* have access to shared entanglement. Can we hope to construct secret sharing schemes that remain resilient in this setting?

We show that the answer is "no". One may wonder whether this negative result may be derived through the direct use of standard quantum teleportation to teleport one share being held by a local leakage adversary to another local leakage adversary. If this was possible, then a single local leakage adversary would be able to access an authorized subset of shares and thus leak the secret. However, this strategy does not work because it would require the local leakage adversaries to communicate classical bits among themselves *before leaking occurs*, since the leakage is fully classical. Instead, our leakage attack relies on ideas from a recent result of Ananth, Goyal, Liu and Liu [AGLL23] and applies standard quantum teleportation followed by *port-based teleportation* [IH08, IH09]. This is a quantum teleportation protocol from Alice to Bob with the useful property that Bob need not perform any error-correction operations on his entangled state – Alice's measurement outcome simply tells Bob which of the EPR pairs (the *ports*) to look at. Although this protocol incurs some error probability, it is known that it can be made arbitrarily close to 0 if the two parties share a large enough number of EPR states [IH08, IH09, CLM+21].

Suppose that we wish to attack shares $(\mathsf{Sh}_1^m, \mathsf{Sh}_2^m)$ of some 2-out-of-2 secret sharing scheme with local leakage adversaries Leak_1 and Leak_2 having access to shared entanglement. At a high level, we first proceed by having Leak_1 teleport its share Sh_1^m in the standard manner to Leak_2, who

receives an encoded version, call it $\overline{\mathsf{Sh}_1^m}$. Then, Leak_2 uses port-based teleportation to send back to Leak_1 both $\overline{\mathsf{Sh}_1^m}$ and Sh_2^m. This information becomes available to Leak_1 in some port whose index is unknown to him. Nevertheless, crucially, Leak_1 knows the measurement outcomes necessary to recover Sh_1^m from $\overline{\mathsf{Sh}_1^m}$! Therefore, we can have Leak_1 apply the decoding procedure of standard teleportation to *every* port, then apply the reconstruction procedure of the secret sharing scheme also to every port, and finally leak *all* the candidates for the classical secret. At the same time, Leak_2 simply leaks the measurement outcome of the port-based teleportation procedure, which (with low error probability) identifies the correct port, and thus the correct candidate secret. See Section 3.1 for more details.

Leakage-resilient PKE from PKE with uncloneable decryption. Our goal is to construct PKE schemes which withstand arbitrary classical leakage on the decryption key. More precisely, the scheme first generates a public encryption key pk and a (quantum) decryption key R_{dec}, and the adversary is allowed to obtain any polynomial amount of classical leakage $\mathsf{Leak}(pk, R_{\mathsf{dec}})$. Then, the goal is to ensure that the adversary cannot distinguish between $\mathsf{Enc}(pk, m_0)$ and $\mathsf{Enc}(pk, m_1)$ given the leakage $\mathsf{Leak}(pk, R_{\mathsf{dec}})$ for any two messages m_0 and m_1.

We begin by considering the recent notion of PKE with *uncloneable decryption*, studied by Coladangelo, Liu, Liu, and Zhandry [CLLZ21]. Roughly speaking, a PKE scheme with uncloneable decryption satisfies the following property: Suppose that the decryption key R_{dec} is split across two adversaries in an arbitrary manner, and that two ciphertexts are sent to these adversaries. Then, the probability that *both* adversaries are able to correctly decrypt their ciphertexts is negligibly close to $1/2$.

It is natural to wonder whether PKE schemes with uncloneable decryption immediately give leakage-resilient PKE. However, although PKE schemes with indistinguishability-based uncloneable decryption guarantees are known, there is an issue that precludes a direct reduction from PKE with uncloneable decryption to leakage-resilient PKE: On the one hand, note that the adversaries' baseline success probability in the uncloneable 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. 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 uncloneable decryption game, should be close to $1/4$, instead of close to $1/2$ as guaranteed by the uncloneable decryption property.

Fortunately, this issue dissipates if we move to the weaker *unpredictability-based* security notion for leakage-resilient PKE, where we only require that the adversary cannot guess a random plaintext m^* given the encoding $\mathsf{Enc}(pk, m^*)$ and the leakage $\mathsf{Leak}(pk, R_{\mathsf{dec}})$. We are able to show that every PKE scheme with (unpredictability-based) uncloneable decryption is also (unpredictability-based) leakage-resilient.

To obtain our final indistinguishability-based leakage-resilient PKE scheme, we show how to compile an arbitrary unpredictability-based leakage-resilient PKE scheme for random messages PKE' into a leakage-resilient PKE scheme for worst-case messages PKE. This is done by combining the underlying PKE scheme with a seeded randomness extractor. More precisely, if we wish to encrypt a fixed 1-bit message $m \in \{0,1\}$, we first sample fresh randomness $k \leftarrow \{0,1\}^\ell$ and encrypt it using the underlying PKE scheme PKE'. Then, one uses this randomness k along with a public seed r to mask m, yielding the encryption

$$\mathsf{PKE}.\mathsf{Enc}(pk, m) = (\mathsf{PKE}'.\mathsf{Enc}(pk, k), r, \langle k, r\rangle \oplus m).$$

This encoding can be extended to multi-bit messages by encoding each bit separately. Using a standard argument, guessing between $\mathsf{PKE.Enc}(pk, m_0)$ and $\mathsf{PKE.Enc}(pk, m_1)$ for any two messages m_0 and m_1 reduces to guessing k, which is randomly chosen, from $\mathsf{PKE'.Enc}(pk, k)$ and the leakage $\mathsf{Leak}(pk, R_{\mathsf{dec}})$. See Section 3.5 for more details.

Leakage-resilient (weak) PRFs from copy-protected PRFs. We focus on weak PRFs in our setting with leakage, since strong PRFs are impossible to construct even under 1 bit of leakage. Our goal is to design a PRF that withstands any polynomial amount of classical leakage on its secret key R_{key}. Namely, an adversary has access to R_{key} and produces a classical string $\mathsf{Leak}(R_{\mathsf{key}})$. Then, given this classical leakage, the adversary must distinguish between the case where it receives $\mathsf{PRF.Eval}(R_{\mathsf{key}}, x^*)$ for a random message x^*, or an independent random string y^*.

Our starting point is the notion of copy-protected PRFs studied in [CLLZ21], which is reminiscent of the notion of PKE with uncloneable decryption that we used to construct leakage-resilient PKE: An adversary with access to the PRF secret key R_{key} cannot *clone* it in a way that allows two parties to distinguish the PRF outputs from random strings. We know how to construct such objects from post-quantum sub-exponential iO and one-way functions along with quantum-hard LWE.

We face issues similar to those found in the PKE setting when constructing leakage-resilient PRFs. As above, although there is no direct reduction from copy-protected PRFs to leakage-resilient PRFs, we are able to show that copy-protected PRFs do satisfy a weaker notion of unpredictability-based leakage-resilience. We can then compile PRFs satisfying unpredictability-based leakage-resilience into the desired (indistinguishability-based) leakage-resilient PRFs by combining the former objects with seeded randomness extractors. One difficulty in this compiler is figuring out where to obtain the random seed from. In the PKE setting, we could use part of the randomness from the encryption procedure as a seed. However, in the PRF setting, there is no ciphertext randomness nor a public key, and the seed clearly cannot be part of the secret key. Fortunately, the input for a weak PRF is guaranteed to be random, and so we are able to use part of it as our seed.

Finally, using standard connections between weak PRFs and message authentication codes, we are able to obtain message authentication codes which are resilient against unbounded classical leakage. The same holds for secret key encryption resilient against unbounded classical leakage. See Section 3.2 for more details.

Leakage-resilient SKE from Weaker Assumptions. We can obtain leakage-resilient SKE by combining our construction of leakage-resilient PRFs with the usual construction of SKE from PRFs (which does go through even with leakage). However, we investigate whether we can construct leakage-resilient SKE by other means which allow us to avoid the strong hardness assumptions such as post-quantum indistinguishability obfuscation which are needed to construct leakage-resilient PRFs.

In our leakage-resilient SKE, the encryption and the decryption keys may be different and we allow the adversary to obtain any polynomial amount of classical leakage separately on the encryption key R_{enc} and the decryption R_{dec}, yielding $\mathsf{Leak}(R_{\mathsf{enc}})$ and $\mathsf{Leak}(R_{\mathsf{dec}})$.

Our insight is that we can combine any (non-leakage-resilient post-quantum) PKE scheme PKE with random Wiesner encodings and a strong seeded extractor to construct leakage-resilient SKE. The idea is as follows: Suppose that PKE uses r random bits to generate its public and secret keys. Sample an extractor source $X \leftarrow \{0,1\}^n$ and seed $S \leftarrow \{0,1\}^d$ for a strong seeded extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^r$. Intuitively, we will (1) use $\mathsf{Ext}(X, S)$ as the randomness for $\mathsf{PKE.KeyGen}$, and (2) use a random Wiesner encoding to mask X so that leakage on the encryption

and decryption keys of the SKE scheme still leaves X with decent min-entropy.

More precisely, we set $R_{\mathsf{enc}} = (\theta, S, pk)$ for $\theta \leftarrow \{0,1\}^n$ the random basis vector and $R_{\mathsf{dec}} = \rho_{X,\theta} = H^\theta|X\rangle$. To encrypt a message m, we compute $ct = \mathsf{PKE.Enc}(pk, m)$ and output the ciphertext (θ, S, ct), while to decrypt we first recover X from $R_{\mathsf{dec}} = \rho_{X,\theta}$ and θ, and then recover (pk, sk) by computing $\mathsf{PKE.KeyGen}(1^\lambda; \mathsf{Ext}(X, S))$, after which we can compute $m = \mathsf{PKE.Dec}(sk, ct)$.

To see why this SKE scheme is leakage-resilient, note that, by our earlier min-entropy bound on random Wiesner encodings with leakage (Equation (2)), we know that X retains a decent amount of min-entropy after $\mathsf{Leak}(R_{\mathsf{dec}}) = \mathsf{Leak}(\rho_{X,\theta})$ is revealed. Since Ext is a strong seeded extractor, this means that $\mathsf{Ext}(X, S)$ is close to uniformly random from the adversary's view, and so the PKE secret key sk remains secret.

Finally, we are also able to obtain unbounded leakage resilient digital signatures by relying on copy protection of digital signatures [LLQZ22]. For more details, see Section 3.3.

1.2.2 Detecting unbounded quantum leakage

We discuss how construct a leakage-detection scheme from any publicly verifiable certified deletion scheme for a primitive. As an example, we will elaborate on public-key encryption, see Section 4 for the other primitives.

We discuss how to construct public-key encryption schemes that support leakage-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 leakage occurred), and a (quantum) decryption key R_{dec}. An adversary is allowed to produce quantum leakage R_{leak} and two challenge messages m_0 and m_1 based on $(pk, tk, R_{\mathsf{dec}})$. Note that this may change the state in register R_{dec}. Before the distinguishing game proceeds, a leakage-detection step is run and the adversary automatically loses if its presence is detected, i.e., $\mathsf{TestLeakage}(tk, R_{\mathsf{dec}}) = \mathsf{LEAKED}$. If no leakage 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

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

We construct PKE schemes with these guarantees by establishing a connection to secure software leasing and cryptography with certified deletion [AL21, BK22, KN22]. 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 correctly 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].

We show that we can construct a PKE scheme that supports leakage-detection from a PKE scheme with secure key leasing. Starting with a PKE scheme with secure key leasing, we construct a $\mathsf{TestLeakage}$ procedure which essentially tries to produce a deletion certificate for the secret decryption key R_{dec}, and outputs $\mathsf{NO\ LEAKAGE}$ if it succeeds. Intuitively, we can argue leakage-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 pretends to leak 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

that 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 9), 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 leakage-detection procedure will skip this measurement and will instead run the verification procedure coherently. Furthermore, the leakage-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 supporting leakage-detection. See Section 4 for more details.

1.2.3 Classical schemes with post-quantum leakage-resilience

Generalizing the "min-entropy drop lemma" to side information with entanglement, and applications. The bulk of the security analysis of many classical leakage-resilient cryptographic schemes is based on the combination of randomness extractors with the following well-known (and quite general) fact proved by Dodis, Ostrovsky, Reyzin, and Smith [DORS08]: Let X be an arbitrary random variable. Suppose that an adversary computes some bounded leakage $L = \mathsf{Leak}(X) \in \{0,1\}^\ell$. Then, it holds that

$$\mathbf{H}_\infty(X|L) \geq \mathbf{H}_\infty(X) - \ell.$$

In words, the optimal guessing probability for X grows (on average) by a factor of at most 2^ℓ after the leakage L is revealed.

We would like to be able to use a similar result in the setting of *post-quantum* leakage, where adversaries have quantum capabilities and access to shared entanglement. Consider the augmented setting where the leakage adversary (which computes the leakage L on X) and the distinguisher (which attempts to guess X given the leakage L) have access to arbitrarily entangled quantum registers R_1 and R_2, respectively. We show a *post-quantum* analogue of the min-entropy drop lemma of [DORS08], stating that if $L = \mathsf{Leak}(X, R_1) \in \{0,1\}^\ell$, then

$$\mathbf{H}_\infty(X|L, R_2) \geq \mathbf{H}_\infty(X) - \ell.$$

In words, the optimal guessing probability for X grows (on average) by a factor of at most 2^ℓ after the leakage L is revealed also when the leakage adversary and the distinguisher share arbitrary entanglement. We note that because of shared entanglement, the entire secret X could have been teleported to the state R_2. However the correction bits are only subject to bounded leakage L.

Using our post-quantum min-entropy drop lemma, we are able to show that several existing constructions in the classical setting can also be proven secure in the post-quantum setting if we replace the randomness extractors being used by *quantum-proof* randomness extractors (of which we know several constructions with good parameters [DPVR12, KK12, CLW14]). This includes PRFs and PKE schemes [HLAWW16], digital signatures [KV09], and general leakage-resilient computation [GR12]. See Section 5 for more details.

2 Notation and preliminaries

2.1 Notation

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 $\rho \geq \sigma$. 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. $x \leftarrow \mathcal{A}(\ldots)$ means sample x from the distribution induced by the randomized algorithm $\mathcal{A}$.

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. Algorithm will mean a quantum algorithm, and our schemes will be uniform QPT algorithms. Adversaries will be stateless, and separate adversaries will unentangled. In the computational setting, negligible means negligible in the security parameter, λ and for two ensembles, $\approx$ means $\approx_{\mathsf{negl}(\lambda)}$. Sizes and bounds, such as leakage bounds, will 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. Finally, for a quantum algorithm Φ with no input, we will write $\Phi(1)$ to denote the execution of Φ_λ, since 1 is the only normalized element of $\mathbb{C}^{\{0\}}$.

Definition 1 (Computational model)**.** *We fix a universal set of unitary gates, such as Hadamard, phase, CNOT, $\frac{\pi}{8}$ gates. We define a quantum polynomial time (QPT) algorithm to be a uniform family of generalized quantum circuits $\{\Phi_\lambda\}_\lambda$ with some fixed polynomial $p(\lambda)$ where each Φ_λ is constructed by introducing an ancilla register of size at most $p(\lambda)$, 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.*

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.*

2.2 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 the book of Nielsen and Chuang [NC10] and Wa-

trous [Wat18] for an overview of these concepts. 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 2 (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 3 (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 1 (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 4 (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 5 (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 2 (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. □

2.3 Port-based teleportation of quantum states

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[1], 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 12 (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.*

2.4 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 6 (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*

$$\mathbf{H}_\infty(X) = -\log \max_{x \in \mathcal{X}} \Pr[X = x].$$

[1] 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].

Definition 7 (k-source)**.** *A random variable X is said to be a* k-source *if* $\mathbf{H}_\infty(X) \geq k$.

Definition 8 (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 9 (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 10 (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 11. *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 3 ([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 lemmas that can be considered quantum variants of chain rule for classical random variables.

Lemma 4 ([DD10, Lemma 1])**.** *Let X, Y be two registers in an independent state $\rho = \sigma \otimes \tau$. Then,*

$$\mathbf{H}_\infty(X|Y) = \mathbf{H}_\infty(X).$$

Lemma 5 (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 12 (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 13 (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 6 ([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 13 ([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* $m = k - 4\log(1/\varepsilon) - O(1)$*.*

We will also need quantum-proof multi-source extractors in some of our constructions.

Definition 14 (Quantum-proof two-source extractor [KK12], adapted)**.** *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is said to be a* strong $(\ell_1, \ell_2, k, \varepsilon)$-quantum-proof two-source extractor *if for any two independent* k*-sources* X*,* Y *supported on* $\{0,1\}^n$ *and local adversaries* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *sharing arbitrary entanglement, with access to* X *and* Y *and with* ℓ_1 *and* ℓ_2 *qubit output, respectively, we have*

$$\mathsf{Ext}(X,Y), \rho^{\mathcal{A}}_{X,Y}, Y \approx_\varepsilon U_m, \rho^{\mathcal{A}}_{X,Y}, Y,$$

where $\rho^{\mathcal{A}}_{X,Y}$ *denotes the entangled state produced by the adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *on input* X *and* Y*, respectively.*

Additionally, we say that Ext *supports* efficient preimage sampling *if given* $z \in \{0,1\}^m$ *we can efficiently sample uniformly at random from the preimage* $\mathsf{Ext}^{-1}(z)$*.*

Kasher and Kempe [KK12] showed that the well-known inner product extractor [CG88] is also quantum-proof with good parameters.

Lemma 7 ([KK12, Corollary 14])**.** *The inner product extractor* $\mathsf{IP} : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}$ *given by* $\mathsf{IP}(x,y) = \langle x, y\rangle$ *is a strong* $(\ell_1, \ell_2, k, \varepsilon)$*-quantum-proof two-source extractor whenever*

$$k - \ell_1 \geq N/2 + \log(1/\varepsilon) - 1.$$

The inner product extractor can be extended to output multiple bits in a standard manner, leading to the following result. To this end, we use the following result of [KK12], which states the multibit generalization of the inner product extractor from [DEOR04] (which supports efficient preimage sampling) is quantum-proof.

Lemma 8 ([KK12], adapted)**.** *There is an explicit strong* $(\ell_1, \ell_2 = \infty, k, \varepsilon)$*-quantum-proof two-source extractor* $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}^M$ *provided that*

$$k - \ell_1 \geq N/2 + M + \log(1/\varepsilon) - 1.$$

Moreover, Ext *supports efficient preimage sampling.*

2.5 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 9 (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.

2.6 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 10 ([TFKW13, Theorem 3])**.** *The winning probability of the n-qubit BB84 MoE game is $\left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^n$.*

2.7 Secret sharing schemes

We introduce basic definitions of access structures and secret sharing schemes.

Definition 15 (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 16 (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*.*

2.8 Weak pseudorandom functions

Definition 17. *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 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* $\left|\Pr[\mathsf{Game}_\mathcal{A} = 1] - \frac{1}{2}\right|$*.*

2.9 Digital signatures

Definition 18. *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 $m \in \mathcal{M}$.*

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

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.*

2.10 Functional encryption

Definition 19 (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.*
- $\mathsf{QKeyGen}(msk, f)$*: Takes in the master secret key and a function f, outputs a functional key R_f for f.*
- $\mathsf{Enc}(pk, m)$*: Takes in the public key and a message m, outputs an encryption of m.*
- $\mathsf{Dec}(R_f, ct)$*: Takes in a functional key R_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.*

$$\Pr\left[\mathsf{Dec}(R_f, ct) = f(m) : \begin{array}{c} msk, pk \leftarrow \mathsf{Setup}(1) \\ R_f \leftarrow \mathsf{QKeyGen}(msk, f) \\ ct \leftarrow \mathsf{Enc}(pk, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

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 $R_f \leftarrow\!\leftarrow \mathsf{QKeyGen}(msk, f)$.*
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 $\left|\Pr[\mathsf{Game}_{\mathcal{A}} = 1] - \frac{1}{2}\right|$. *If the adversary outputs the challenge messages before the keys are sampled, we call it* selective indistinguishability security.

2.11 Hash-proof systems

We reproduce the definitions of symmetric- and public-key weak hash-proofs systems from [HLAWW16] for convenience.

Definition 20 (Symmetric-key weak hash-proof system (wHPS) [HLAWW16], almost verbatim)**.** *Let* $\mathcal{X}, \mathcal{Y}, \mathcal{K}$ *be some efficient ensembles and let* $\mathcal{F} = \{F_K : \mathcal{X} \to \mathcal{Y}\}_{K \in \mathcal{K}}$ *be some efficient function family with the following PPT algorithms.*

- $\mathsf{samK} \leftarrow \mathsf{SamGen}(K)$ *takes an input* $K \in \mathcal{K}$ *and outputs a sampling key* samK.
- $X \leftarrow \mathsf{Dist}_1(\mathsf{samK})$, $X \leftarrow \mathsf{Dist}_2(\mathsf{samK})$ *are two distributions that sample* $X \in \mathcal{X}$ *using the sampling key* samK. *For convenience, we also define the distribution* $\mathsf{Dist}_0(\mathsf{samK})$ *which just samples a uniformly random* $X \leftarrow \mathcal{X}$ *and ignores the sampling key* samK.

We say that $\mathcal{F}$ *is a* symmetric-key wHPS *if it satisfies the following two properties:*

- *Input indistinguishability. For any polynomial* $q = q(\lambda)$ *and any choice of* $(b_1, \ldots, b_q), (b'_1, \ldots, b'_q) \in \{0,1,2\}^q$*, the following distributions are computationally indistinguishable:*

$$(K, X_1, \ldots, X_q) \approx (K, X'_1, \ldots, X'_q)$$

  where $K \leftarrow \mathcal{K}_\lambda$, $\mathsf{samK} \leftarrow \mathsf{SamGen}(K)$, $\{X_i \leftarrow \mathsf{Dist}_{b_i}(\mathsf{samK})\}$, $\{X'_i \leftarrow \mathsf{Dist}_{b'_i}(\mathsf{samK})\}$.

- *Smoothness. For any polynomial* $q = q(\lambda)$*, the following distributions are* statistically *equivalent:*

$$(X_1, \ldots, X_q, Y_1, \ldots, Y_q, X^*, Y^*) \equiv (X_1, \ldots, X_q, Y_1, \ldots, Y_q, X^*, U)$$

  where the distributions are defined by $K \leftarrow \mathcal{K}_\lambda$, $\mathsf{samK} \leftarrow \mathsf{SamGen}(K)$, $\{X_i \leftarrow \mathsf{Dist}_1(\mathsf{samK}), Y_i = F_K(X_i)\}_{i \in [q]}$, $X^* \leftarrow \mathsf{Dist}_2(\mathsf{samK})$, $Y^* = F_K(X^*)$ *and* $U \leftarrow \mathcal{Y}$.

Definition 21 (Public-key weak hash-proof system [HLAWW16], almost verbatim)**.** *A weak hash-proof system (wHPS) with output space* $\mathcal{K}$ *consists of the algorithms* $\mathsf{Gen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap}^*$ *that satisfy the following properties.*

- *Correctness. For all* (pk, sk) *in the range of* $\mathsf{Gen}(1^\lambda)$,

$$\Pr\left[k = k' : (c, k) \leftarrow \mathsf{Encap}(pk), k' = \mathsf{Decap}(c, sk)\right] = 1.$$

- *Ciphertext indistinguishability. We have*

$$(pk, sk, c) \approx (pk, sk, c^*)$$

where $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda), (c, k) \leftarrow \mathsf{Encap}(pk), c^* \leftarrow \mathsf{Encap}^*(pk)$.

- *Smoothness.*

$$(pk, c^*, k^*) \equiv (pk, c^*, k).$$

where $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda), k \leftarrow \mathcal{K}, c^* \leftarrow \mathsf{Encap}^*(pk), k^* = \mathsf{Decap}^*(c^*, sk)$.

3 Cryptographic schemes resilient to unbounded classical leakage

In this section, we use quantum resources to design leakage-resilient schemes that are impossible to break using *any* classical leakage attack. For the computationally secure constructions, *unbounded* will mean any polynomial amount of classical leakage while the size of the scheme does not depend on the amount of classical leakage allowed.

We first present a lemma regarding unbounded classical leakage on BB84 states that will be useful in most of our schemes.

Lemma 11 (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 Section 2.6. 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 10, 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 3. □

We can also extend Lemma 11 against attacks that are composed of unbounded classical leakage and bounded quantum leakage.

Lemma 12 (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 13 ([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 5. We also have $\Delta^C(\sigma) = \sum_a Tr(E_{a,a}\rho)E_{a,a}$ by Definition 4 where each $E_{a,a}$ is a density matrix while $\{E_{a,a}\}_a$ form a POVM. Hence, by Lemma 13, 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 5 we have

$$\mathbf{H}_\infty(A, (B, C), D) \geq \mathbf{H}_\infty(A|B, C) - \ell_q(\lambda) \tag{3}$$

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 11 we have $\mathbf{H}_\infty(A|B, C) \geq C_{BB84} \cdot \lambda$. Finally combining this with Equation (3) yields the result. □

3.1 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[2] 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 11 and 12.

Definition 22 (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*

$$(\mathsf{Sh}_i)_{i\in T}, (\mathsf{Leak}_i(\mathsf{Sh}_i))_{i\notin T} \approx_\varepsilon (\mathsf{Sh}'_i)_{i\in T}, (\mathsf{Leak}_i(\mathsf{Sh}'_i))_{i\notin T}, \tag{4}$$

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 (4) 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.[3] 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 6 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

  $$\Pr[\mathsf{InvExt}(Z, S) = \perp] \leq \varepsilon_{\mathsf{ext}} + \varepsilon_{\mathsf{unif}}. \tag{5}$$

  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 (5).

[2] Local leakage-resilience is trivially unachievable for access structures with singletons.

[3] Note that locally leakage-resilient secret sharing is unachievable over any access structure with singletons.

- 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.[4]

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 $(Z_1, \ldots, Z_n) \leftarrow \mathsf{Share}(m)$.
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 14. *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 14 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 (5) 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 1. *For every secret* $m \in \{0,1\}^u$ *and* $i \in \{t+1, \ldots, n\}$ *it holds that*

$$\mathsf{Hyb}_{i-1}^m \approx_{2(\varepsilon_{\mathsf{ext}}+\varepsilon_{\mathsf{unif}})} \mathsf{Hyb}_i^m.$$

[4] For the sake of simplicity, we avoid parameterizing these schemes by the secret length.

Assuming Claim 1, 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 14. We proceed to prove Claim 1, which concludes our argument.

Proof of Claim 1. 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

$$Z_i, S_i, S, \theta_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(Z_i, S)\rangle, S_i, \theta_i) \approx_{2(\varepsilon_{\mathsf{ext}}+\varepsilon_{\mathsf{unif}})} Z_i, S_i, S, \theta_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, S_i, \theta_i). \quad (6)$$

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 (6) are randomized functions of $Z_i, S_i, S, \theta_i, \theta, X$, in order to show Equation (6) it is enough to argue that

$$U_w, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|\mathsf{InvExt}(U_m, S)\rangle, \widetilde{S}_i, \widetilde{\theta}_i) \approx_{2\varepsilon_{\mathsf{ext}}} U_w, \widetilde{S}_i, S, \widetilde{\theta}_i, \theta, \mathsf{Leak}_i(H^\theta|X\rangle, \widetilde{S}_i, \widetilde{\theta}_i). \quad (7)$$

As $\widetilde{S}_i$ and $\widetilde{\theta}_i$ are independent of X and θ, Lemma 11 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 (7) follows by the triangle inequality. □

□

3.1.1 Setting parameters in the compiler

In this section we show how to instantiate the compiler from Theorem 14 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 1 (Theorem 1, restated)**.** *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 6 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 14 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

$$\varepsilon_{\mathsf{leak}} = 5n(\varepsilon_{\mathsf{ext}} + \varepsilon_{\mathsf{unif}}) + \varepsilon_{\mathsf{priv}} = O(n2^{-\lambda}).$$

□

3.1.2 Breaking leakage-resilience with unbounded shared entanglement and classical leakage

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 15 (Theorem 2, restated)**.** *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 exists 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.

Proof. 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 Section 2.3), 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 12, if the number N of EPR pairs shared 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. □

These results also have further implications for communication complexity. Namely, together they imply that there is a quantum circuit with classical output satisfying the following:

- For parties not sharing entanglement, it is impossible to have a 1-round protocol with classical message that has non-negligible success probability, even with unbounded amount of communication;
- For parties that can share entanglement, there is a 1-round protocol computing it perfectly.

3.2 Pseudorandom functions

In this section, we define leakage protection schemes for weak PRFs (Definition 17) and show how to construct schemes resilient against unbounded classical leakage from copy protection schemes for PRFs. We start by generalizing the PRF copy protection definition of Coladangelo, Liu, Liu, and Zhandry [CLLZ21] to allow for other security models.

Definition 23 (PRF protection scheme)**.** *A protection scheme* PRF *for a wPRF* $\mathcal{F} = \{f_k\}_k$ *consists of the following QPT algorithms.*

- $\mathsf{Setup}(1)$: *Outputs a classical PRF key* k *for* $\mathcal{F}$.

- $\mathsf{QKeyGen}(k, 1^t)$: *Takes the key k and a collusion bound t, outputs quantum key registers $R_{\mathsf{key}} = (R_{\mathsf{key},i})_{i\in[t]}$ in some product state.*
- $\mathsf{Eval}(R_{\mathsf{key},i}, x)$*: Takes a quantum key and an input x and returns the evaluation $f_k(x)$.*

It satisfies the following correctness property.

Correctness *For all inputs x and all $i \in [t]$,*

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

As argued by [CLLZ21], correctness property implies by gentle measurement lemma [Aar05] that we can implement evaluation in a way such that we can *rewind* the algorithm after decryption, and therefore we can use it polynomially many times. We will assume that Eval is implemented this way.

Coladangelo et al. [CLLZ21] and Liu et al. [LLQZ22] give two different definitions of copy protection security. For anti-piracy security, it is required that an adversary that has access to t copies of the PRF key cannot produce $t+1$ registers that can all be used to predict the PRF output at independent random challenges. For the supposedly stronger[5] notion of indistinguishability anti-piracy security, again we provide the adversary with t copies of the PRF key and it produces $t+1$ possibly entangled registers and programs. We require that, when all presented with independent, random distinguishing challenges, the $t+1$ registers cannot all distinguish the case in which they are.

Definition 24 (t-bounded collusion resistant anti-piracy security [LLQZ22])**.** *A copy protection scheme PRF for a wPRF $\mathcal{F} = \{f_k\}_k$ is said to satisfy anti-piracy security if for all QPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game is negligible.*

1. *The challenger runs $k \leftarrow \mathsf{PRF.Setup}(1)$ and*
$$R_{\mathsf{key}} = (R_{\mathsf{key},i})_{i\in[t]} \leftarrow \mathsf{PRF.QKeyGen}(k, 1^t).$$
2. *$\mathcal{A}$ gets access to R_{key}, and it produces $t+1$ QPT programs $(\mathcal{A}_i)_{i\in[t+1]}$ and $t+1$ registers $(R_i)_{i\in[t+1]}$.*
3. *The challenger samples challenge input $x_i^* \leftarrow \mathcal{M}$ for each $i \in [t+1]$.*
4. *The challenger runs $y_i \leftarrow \mathcal{A}_i(R_i, x_i^*)$ for each $i \in [t+1]$.*
5. *Output 1 if and only if $y_i = f_k(x_i^*)$ for* all *$i \in [t+1]$.*

Definition 25 (t-bounded collusion resistant indistinguishability anti-piracy security [LLQZ22])**.** *A copy protection scheme PRF for a wPRF $\mathcal{F} = \{f_k\}_k$ is said to satisfy indistinguishability anti-piracy security if for all QPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game is negligible.*

1. *The challenger runs $k \leftarrow \mathsf{PRF.Setup}(1)$ and*
$$R_{\mathsf{key}} = (R_{\mathsf{key},i})_{i\in[t]} \leftarrow \mathsf{PRF.QKeyGen}(k, 1^t).$$

[5] While this definition seems stronger, [CLLZ21] state that it is not clear if this definition implies the previous one.

2. *$\mathcal{A}$ gets access to R_{key}, and it produces $t+1$ QPT programs $(\mathcal{A}_i)_{i\in[t+1]}$ and $t+1$ registers $(R_i)_{i\in[t+1]}$.*

3. *The challenger executes the following for each $i \in [t+1]$:*

   (a) *Sample challenge input $x_i^* \leftarrow \mathcal{M}$.*

   (b) *Sample challenge bit $b_i \leftarrow \{0,1\}$*

   (c) *If $b_i = 0$, set $y_i^* = f_k(x_i^*)$. If $b_i = 1$, sample $y_i^* \leftarrow \mathcal{Y}$.*

4. *The challenger runs $b_i' \leftarrow \mathcal{A}_i(R_i, x_i^*, y_i^*)$ for each $i \in [t+1]$.*

5. *Output 1 if and only if $b_i' = b_i$ for* all *$i \in [t+1]$.*

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

Theorem 16 ([CLLZ21], [LLQZ22])**.** *Assume the existence of post-quantum sub-exponentially secure iO and one-way functions, and the quantum hardness of LWE. Let $\mathcal{F}$ be a puncturable PRF family with error $2^{-\lambda-1}$ for min-entropy $p_1(\lambda)$, input space $\{0,1\}^{p_1(\lambda)}$ and output space $\{0,1\}^{p_2(\lambda)}$ where $p_1(\cdot), p_2(\cdot)$ are some polynomials satisfying $p_1(\lambda) \geq p_2(\lambda)+2\lambda+4$. Then, for any $t = \mathrm{poly}(\lambda)$, there exists a protection scheme PRF for $\mathcal{F}$ that satisfies both t-bounded collusion resistant anti-piracy security and t-bounded collusion resistant indistinguishability anti-piracy security.*

Note that PRFs as required by Theorem 16 exist [CLLZ21].

Now we continue with leakage-resilience. Since the adversary can always leak (parts of) the evaluation of the PRF on an input of its choice, rather than standard (strong) PRFs one generally considers weak PRFs in the setting of leakage-resilience [HLAWW16]. We also define our model in this way and we will omit the qualifier *weak* for PRFs in leakage-resilience context since it will be implicit. We first start with the further weaker setting where the adversary needs to predict the output of the PRF on a random input rather than distinguishing a random input and its evaluation from a pair consisting of a random input and a random sample from the output space. While this will be sufficient to construct leakage-resilient MACs in Section 3.4, we also show a scheme using a Goldreich-Levin type construction that satisfies the more standard, indistinguishability-based security definition.

Definition 26 (Unbounded classical leakage-resilient unpredictable pseudorandom functions)**.** *A protection scheme PRF for a wPRF $\mathcal{F} = \{f_k\}_k$ is said to be t-*copy unpredictable $(*, \ell_q(\lambda))$-leakage-resilient *if for all polynomials $\ell_c(\cdot)$ and for all QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ the advantage of $\mathcal{A}$ in the following game is negligible.*

1. *The challenger runs $k \leftarrow \mathsf{PRF.Setup}(1)$ and*

$$R_{\mathsf{key}} = (R_{\mathsf{key},i})_{i\in[t]} \leftarrow \mathsf{PRF.QKeyGen}(k, 1^t).$$

2. *The challenger samples a challenge input $x^* \leftarrow \mathcal{X}$ and computes $y^* \leftarrow \mathsf{PRF.Eval}(R_{\mathsf{key},1}, x^*)$.*

3. *The leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ gets access to R_{key} and produces a leakage register R_{leak} that consists of $\ell_q(\lambda)$ qubits and $\ell_c(\lambda)$ classical bits.*

4. *$\mathcal{A}_{\mathsf{Main}}$ gets R_{leak} and x^*, and it outputs a guess y'.*

5. *Output 1 iff $y' = y^*$.*

Note that for unpredictable wPRFs (both in the context of copy protection and leakage-resilience), we will henceforth implicitly require the key space, the input space and the output space to be all superpolynomial size since otherwise the adversary can guess the key, the challenge input or the evaluation of the challenge input with non-negligible probability.

Definition 27 (Unbounded classical leakage-resilient pseudorandom functions)**.** *A protection scheme* PRF *for a wPRF* $\mathcal{F} = \{f_k\}_k$ *is said to be* t-copy $(*, \ell_q(\lambda))$-leakage-resilient *if for all polynomials* $\ell_c(\cdot)$ *and for all QPT adversaries* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *the advantage of* $\mathcal{A}$ *in the following game is negligible.*

1. *The challenger runs* $k \leftarrow \mathsf{PRF.Setup}(1)$ *and* $R_{\mathsf{key}} = (R_{\mathsf{key},i})_{i\in[t]} \leftarrow \mathsf{PRF.QKeyGen}(k, 1^t)$.
2. *The challenger samples a challenge input* $x^* \leftarrow \mathcal{X}$ *and a challenge bit* $b \leftarrow \{0,1\}$. *If* $b = 0$, *it sets* $y^* \leftarrow \mathsf{PRF.Eval}(R_{\mathsf{key},1}, x^*)$ *and if* $b = 1$, *it samples* $y^* \leftarrow \mathcal{Y}$.
3. *The leakage adversary* $\mathcal{A}_{\mathsf{Leak}}$ *gets access to* R_{key} *and produces a leakage register* R_{leak} *that consists of* $\ell_q(\lambda)$ *qubits and* $\ell_c(\lambda)$ *classical bits.*
4. $\mathcal{A}_{\mathsf{Main}}$ *gets* x^*, y^* *and* R_{leak}. *It outputs a guess* b'.
5. *Output* 1 *iff* $b' = b$.

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

We can also define a multi-challenge variant of this security notion, which will be useful in some applications such as multi-message secure private key encryption constructed in Section 3.6.

Definition 28 (Unbounded classical leakage-resilient pseudorandom functions - multi-challenge variant)**.** *A protection scheme* PRF *for a wPRF* $\mathcal{F} = \{f_k\}_k$ *is said to be* t-copy $(*, \ell_q(\lambda))$-leakage-resilient for multiple inputs *if for all polynomials* $p(\cdot)$ *and* $\ell_c(\cdot)$, *for all QPT adversaries* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *the advantage of* $\mathcal{A}$ *in the following game is negligible.*

1. *The challenger runs* $k \leftarrow \mathsf{PRF.Setup}(1)$ *and* $R_{\mathsf{key}} = (R_{\mathsf{key},i})_{i\in[t]} \leftarrow \mathsf{PRF.QKeyGen}(k, 1^t)$.
2. *The challenger samples challenge inputs* $x_i^* \leftarrow \mathcal{X}$ *for each* $i \in [p(\lambda)]$ *and a challenge bit* $b \leftarrow \{0,1\}$.
3. *For each* $i \in [p(\lambda)]$, *the challenger sets* $y_i^* \leftarrow \mathsf{PRF.Eval}(R_{\mathsf{key},1}, x^*)$ *if* $b = 0$ *and it samples* $y_i^* \leftarrow \mathcal{Y}$ *if* $b = 1$.
4. *The leakage adversary* $\mathcal{A}_{\mathsf{Leak}}$ *gets access to* R_{key} *and produces a leakage register* R_{leak} *that consists of* $\ell_q(\lambda)$ *qubits and* $\ell_c(\lambda)$ *classical bits.*
5. $\mathcal{A}_{\mathsf{Main}}$ *gets* $(x_i^*, y_i^*)_{i\in[p(\lambda)]}$ *and* R_{leak}. *It outputs a guess* b'.
6. *Output* 1 *iff* $b' = b$.

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

Theorem 17. *Let* PRF *be a* t*-copy* $(*, \ell_q(\lambda))$*-leakage-resilient PRF protection scheme. Then,* PRF *is also* t*-copy* $(*, \ell_q(\lambda))$*-leakage-resilient for multiple inputs.*

Proof. This can be shown via a simple hybrid argument.

By overwhelming correctness of PRF and by key rewinding, in the leakage game, we can replace all calls to PRF.Eval with actual evaluations of the PRF. Then, it is easy to see that indistinguishability based security from Definition 27 is equivalent to indistinguishability of the ensembles

$$(x, f_k(x), \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) \tag{8}$$

and

$$(x, y, \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) \tag{9}$$

for all *admissible* $\mathcal{A}_{\mathsf{Leak}}$ where $x \leftarrow \mathcal{X}$ and $y \leftarrow \mathcal{Y}$. Similarly, the indistinguishability based security from Definition 28 is equivalent to indistinguishability of the ensembles

$$(x_1, \ldots, x_{p(\lambda)}, f_k(x_1), \ldots, f_k(x_{p(\lambda)}), \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) \tag{10}$$

and

$$(x_1, \ldots, x_{p(\lambda)}, y_1, \ldots, y_{p(\lambda)}, \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) \tag{11}$$

for all *admissible* $\mathcal{A}_{\mathsf{Leak}}$ where $x_i \leftarrow \mathcal{X}$ and $y_i \leftarrow \mathcal{Y}$ for all $i \in [p(\lambda)]$.

Assume (8) $\approx$ (9) for all *admissible* $\mathcal{A}_{\mathsf{Leak}}$ and we will show (10) $\approx$ (11). For $i \in [p(\lambda)]$, define Hyb_i to be

$$(x_1, \ldots, x_{p(\lambda)}, f_k(x_1), \ldots, f_k(x_{i-1}), y_i, \ldots, y_{p(\lambda)}, \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}}))$$

Then, we have $\mathsf{Hyb}_1 \equiv$ (11) and $\mathsf{Hyb}_{p(\lambda)+1} \equiv$ (10). Now suppose for a contradiction that $\mathsf{Hyb}_i \not\approx \mathsf{Hyb}_{i+1}$ for some $i \in [p(\lambda)]$. Then, it is easy to see that (8) $\not\approx$ (9) for the adversary $\mathcal{A}'_{\mathsf{Leak}}$ constructed below.

$\underline{\mathcal{A}'_{\mathsf{Leak}}(R_{\mathsf{key}})}$

1. Sample $x_1, \ldots, x_{i-1} \leftarrow \mathcal{X}$
2. For each $j \in [i-1]$, run $z_j \leftarrow \mathsf{PRF.Eval}(R_{\mathsf{key}}, x_j)$.
3. Output $x_1, \ldots, x_{i-1}, z_1, \ldots, z_{i-1}, \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})$

Therefore, we have $\mathsf{Hyb}_i \approx \mathsf{Hyb}_{i+1}$ for all $i \in [p(\lambda)]$. Applying the hybrid lemma completes proof. □

Now we show that any copy protection scheme with anti-piracy security is also unpredictable leakage-resilient against unbounded classical leakage given a constant number of copies of the key.

Theorem 18. *Let $\mathcal{F}$ be a pseudorandom function family and* PRF *a copy protection scheme for $\mathcal{F}$ that satisfies t-bounded collusion resistant anti-piracy security for some constant t. Then,* PRF *is t-copy unpredictable $(*, 0)$-leakage-resilient.*

Proof. We first start by modifying the unpredictability leakage game by changing the line $y^* \leftarrow \mathsf{PRF.Eval}(R_{\mathsf{key},1})$ to $y^* = f_k(x^*)$. We are allowed to do this modification by the overwhelming correctness of the scheme and since we are rewinding the evaluation algorithm after each use.

Let $\mathcal{F} = \{f_k\}$ and suppose there is an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the (modified) leakage game with probability $\frac{1}{p(\lambda)}$ for some polynomial $p(\cdot)$ and infinitely many values of λ. We then construct an adversary as follows and show that $\mathcal{A}'$ wins the anti-piracy game with non-negligible probability. Given R_{key}, the adversary $\mathcal{A}'$ runs $\mathcal{A}_{\mathsf{Leak}}$ on it to produce a classical string L. Then,

it outputs $(\mathcal{A}_{\mathsf{Main}}, L)_{i\in[t+1]}$. Observe that conditioned on fixed values a of the leakage and $k^{'}$ of the keys, the probability of winning the anti-piracy game is $\Pr[\mathcal{A}_{\mathsf{Main}}(a, x^*) = f_{k'}(x^*) : x^* \leftarrow \mathcal{X}]^{t+1}$ since the challenges and the piracy adversaries are all independent. Hence, we can write the winning probability for the anti-piracy game as

$$\begin{aligned} p_{\mathsf{piracy}} &= \sum_{a,k^{'}} \Pr\left[\mathcal{A}_{\mathsf{Main}}(a, x^*) = f_{k^{'}}(x^*) : x^* \leftarrow \mathcal{X}\right]^{t+1} \cdot \Pr\left[L = a, k = k^{'}\right] \\ &= \mathbb{E}_{L,k}[\Pr_{x^*}[\mathcal{A}_{\mathsf{Main}}(L, x^*) = f_k(x^*)]^{t+1}] \end{aligned}$$

Similarly we can write the probability of winning the leakage game as

$$\begin{aligned} p_{\mathsf{leak}} &= \sum_{a,k^{'}} \Pr\left[\mathcal{A}_{\mathsf{Main}}(a, x^*) = f_{k^{'}}(x^*) : x^* \leftarrow \mathcal{X}\right] \cdot \Pr\left[L = a, k = k^{'}\right] \\ &= \mathbb{E}_{L,k}[\Pr_{x^*}[\mathcal{A}_{\mathsf{Main}}(L, x^*) = f_k(x^*)]] \end{aligned}$$

Then, by Jensen's inequality we see that $p_{\mathsf{piracy}} \geq (p_{\mathsf{leak}})^{t+1} = \frac{1}{p^{t+1}(\lambda)}$, which shows that $\mathcal{A}^{'}$ has a non-negligible probability of winning the game for constant t. □

Corollary 2. *Assume the existence of post-quantum sub-exponentially secure iO and one-way functions, and the quantum hardness of LWE. Let $\mathcal{F}$ be a puncturable PRF family with error $2^{-\lambda-1}$ for min-entropy $p_1(\lambda)$, input space $\{0,1\}^{p_1(\lambda)}$ and output space $\{0,1\}^{p_2(\lambda)}$ where $p_1(\cdot), p_2(\cdot)$ are some polynomials satisfying $p_1(\lambda) \geq p_2(\lambda) + 2\lambda + 4$. Then, for any constant t, there exists a t-copy unpredictable $(*, 0)$-leakage-resilient protection scheme for $\mathcal{F}$.*

Proof. Invoke Theorem 16 and Theorem 18. □

Finally, from any leakage-resilient unpredictable wPRF, we show how to construct a leakage-resilient indistinguishable wPRF with the same leakage bound. While we phrase the result in terms of PRF protection schemes, note that it also applies to the classical leakage-resilience setting by simply setting the protection scheme for the unpredictable PRF to be the trivial scheme that stores the key in the plain. Hence, the construction also generalizes the result of Naor and Reingold [NR98] to the leakage-resilient setting.

Theorem 19. *Let $\mathcal{F}^{'} = \{f_k^{'}\}_k$ be a wPRF with input space $\{0,1\}^{p_1(\lambda)}$ and output space $\{0,1\}^{p_2(\lambda)}$. Let $\mathsf{PRF}^{'}$ be a t-copy unpredictable $(\ell(\lambda), 0)$-leakage-resilient protection scheme for $\mathcal{F}^{'}$. Then, PRF constructed below is a t-copy $(\ell(\lambda), 0)$-leakage-resilient protection scheme for the 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:* $f_k(x_1||x_2) = \langle f_k^{'}(x_1), x_2\rangle$

PRF

- $\underline{\mathsf{PRF.Setup}(1)}$
  1. *Output* $\mathsf{PRF}'.\mathsf{Setup}(1)$
- $\underline{\mathsf{PRF.QKeyGen}(k, 1^t)}$
  1. *Output* $\mathsf{PRF}'.\mathsf{QKeyGen}(k, 1^t)$
- $\underline{\mathsf{PRF.Eval}(R_{\mathsf{key},i}, x_1||x_2)}$
  1. *Output* $\langle \mathsf{PRF}'.\mathsf{Eval}(R_{\mathsf{key},i}, x_1), x_2 \rangle$

Proof. It is easy to see that PRF satisfies correctness. We move to leakage-resilience. As before, both in the prediction and indistinguishability game, we change the line $y^* \leftarrow \mathsf{PRF.Eval}(R_{\mathsf{key},1})$ to $y^* = f_k(x^*)$, which is allowed since PRF' and PRF both satisfy overwhelming correctness and since we are rewinding the evaluation algorithm. Since the leakage is only classical, we will denote the contents of R_{leak} as L. We claim that PRF is t-copy $(\ell(\lambda), 0)$-leakage-resilient. For a contradiction, suppose there is an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the indistinguishability game against PRF with non-negligible advantage $\frac{1}{2} + \frac{1}{2w(\lambda)}$ where $w(\cdot)$ is some polynomial, and let $\{\rho_\lambda\}_\lambda$ denote the (possibly non-uniform quantum) advice of $\mathcal{A}_{\mathsf{Main}}$. That is,

$$\frac{1}{2}\Pr[\mathcal{A}_{\mathsf{Main}}(x, f_k(x), \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) = 0] + \frac{1}{2}\Pr_{y \leftarrow \{0,1\}}[\mathcal{A}_{\mathsf{Main}}(x, y, \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) = 1] \geq \frac{1}{2} + \frac{1}{2w(\lambda)}$$

Then, it is easy to see that we have

$$\frac{1}{2}\Pr[\mathcal{A}_{\mathsf{Main}}(x, f_k(x), \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) = 0] + \frac{1}{2}\Pr\Big[\mathcal{A}_{\mathsf{Main}}(x, \overline{f_k(x)}, \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})) = 1\Big] \geq \frac{1}{2} + \frac{1}{w(\lambda)}.$$

Now, we construct the following algorithm $\mathcal{D}$ that guesses $f_k(x)$ with non-negligible advantage given the leakage.

$\underline{\mathcal{B}(x, L)}$

- Sample $g \leftarrow \{0, 1\}$
- Run $\mathcal{A}_{\mathsf{Main}}(x, g, L)$. If it outputs 0, output g. Otherwise, output $\overline{g}$.

We then have,

$$\begin{aligned}\Pr[\mathcal{B}(x, L) = f_k(x)] &= \frac{1}{2} \cdot \Pr[\mathcal{A}_{\mathsf{Main}}(x, g, L) = 0 | g = f_k(x)] \\ &\quad + \frac{1}{2} \cdot \Pr\Big[\mathcal{A}_{\mathsf{Main}}(x, g, L) = 1 | g = \overline{f_k(x)}\Big] \\ &\geq \frac{1}{2} + \frac{1}{w(\lambda)}.\end{aligned}$$

For the prediction game for PRF', we construct the following adversary $\mathcal{A}'_{\mathsf{Main}}$ with the advice $\{\rho_\lambda^{\otimes q(\lambda))}\}_\lambda$ where we set $q(\lambda) = \lceil \log(4p_1(\lambda) \cdot w^2(\lambda) + 1) \rceil$. While we will not explicitly state the advice fed to the adversaries henceforth; since $\mathcal{A}'_{\mathsf{Main}}$ invokes $\mathcal{A}_{\mathsf{Main}}$ for $q(\lambda)$ many times, this advice is *sufficient*.

- $\underline{\mathcal{A}'_{\mathsf{Leak}}(R_{\mathsf{key}})}$
  1. Output $\mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{key}})$
- $\underline{\mathcal{A}'_{\mathsf{Main}}(x^*, L)}$
  1. For each $j \in [q(\lambda)]$, sample $g_j \leftarrow \{0,1\}$ and $r_j \leftarrow \{0,1\}^{p_1(\lambda)}$
  2. For each index $i \in [p_1(\lambda)]$ and subset $J \subseteq [q(\lambda)]$ with $J \neq \emptyset$,
$$b_i^J \leftarrow \mathcal{B}\left(x^* \middle\| \left(\bigoplus_{j\in J} r_j \oplus e_i\right), L\right)$$
  3. For each $i \in [p_1(\lambda)]$, set b_i to be the majority bit in $\{b_i^J\}_{\substack{J\subseteq[q(\lambda)],\\ J\neq\emptyset}}$.
  4. Output $(b_i)_{i\in[p_1(\lambda)]}$

Define the set of *good* inputs and leakages GOOD as follows:

$$\mathsf{GOOD} = \left\{(x, L) : \Pr_{z\leftarrow\{0,1\}^{p_1(\lambda)}}[\mathcal{B}((x, z), L) \text{ outputs the correct bit}] \geq \frac{1}{2} + \frac{1}{2w(\lambda)}\right\}.$$

Then, an averaging argument shows that

$$\Pr[(x, L) \in \mathsf{GOOD}] \geq \frac{1}{2w(\lambda)}.$$

Further, indepedently with probability $\frac{1}{2^{q(\lambda)}} \geq \frac{1}{8p_1(\lambda)\cdot w^2(\lambda)}$, we will have that for all $j \in [q(\lambda)]$,

$$g_j = \langle f_k(x^*), r_j\rangle.$$

Now condition on $x^*, L \in \mathsf{GOOD}$ and all g_j being *correct*.

Observe that, conditioned on $x^*, L \in \mathsf{GOOD}$, for a fixed i, all calls to $\mathcal{A}_{\mathsf{Main}}$ are pairwise independent. Hence, using the fact that $\mathcal{A}_{\mathsf{Main}}$ succeeds with probability $\frac{1}{2} + \frac{1}{2w(\lambda)}$ and by Chebyshev's inequality, for any fixed i we get that

$$b_i = (f_k(x^*))_i$$

with probability at least $1-\frac{1}{4p_1(\lambda)}$. Finally, by a union bound over $i \in [p_1(\lambda)]$, we get $x^* = (b_i)_{i\in[p_1(\lambda)]}$ with probability at least $\frac{3}{4}$ conditioned on $x^*, L \in \mathsf{GOOD}$ and all g_j being correct. Hence, $\mathcal{A}'$ wins the prediction game against PRF' with probability $\frac{3}{64p_1(\lambda)\cdot w^3(\lambda)}$.

□

Our proof crucially relies on the fact that the leakage is classical, since we run the adversary $\mathcal{A}_{\mathsf{Main}}$ multiple times on the leakage. We leave it as an open question to construct qubit leakage-resilient wPRFs from qubit leakage-resilient unpredictable wPRFs.

By using multiple independent copies of this PRF, we can achieve any desired output length.

Corollary 3. *Assuming the existence of post-quantum sub-exponentially secure iO and one-way functions, and the quantum hardness of LWE, for any constant t and any polynomial $p(\cdot)$, there exists a wPRF family $\mathcal{F}$ with $p(\lambda)$-bit output and a protection scheme PRF for $\mathcal{F}$ that is t-copy $(*, 0)$-leakage-resilient.*

Proof. By invoking Corollary 2 and Theorem 19, we can obtain a wPRF $\mathcal{F}' = \{f'_k\}_k$ with 1-bit output and a t-copy $(*,0)$-leakage-resilient protection scheme PRF' for $\mathcal{F}'$. We construct $\mathcal{F} = \{f_k\}_k$ by sampling $p(\lambda)$ many independent keys $\{k_i\}_{i\in[p(\lambda)]}$ for $\mathcal{F}'$ and to evaluate f_k, we evaluate each of f'_{k_i} and concatenate the outputs. That is, we set

$$f_k(x) = f'_{k_1}(x)||\cdots||f'_{k_{p(\lambda)}}(x).$$

The protection scheme PRF is constructed the obvious way: by using independent schemes for each key. A simple hybrid argument proves the security. □

3.3 Digital signatures

In this section, we define digital signature schemes secure against unbounded classical leakage and show how to construct them from copy protection schemes [CLLZ21, LLQZ22]. We will require that the adversary sign a random challenge message (rather than one selected by it), since with classical signatures the adversary can always sign a message of its choice and leak the signature. As in Section 3.2, we first start by generalizing the definition of Liu et al. [LLQZ22] to allow for different security models.

Definition 29 (Digital signatures with quantum signing key)**.** *A digital signature scheme* DS *consists of the following QPT algorithms.*

- $\mathsf{Setup}(1)$: *Outputs a classical signing key sk and a public 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 a quantum signing key and a message m, returns a signature for m.*
- $\mathsf{Ver}(vk, m, s)$*: Takes a public verification key, a message m and a (supposed) signature s for m, returns 1 if s is a valid signature for m.*

It satisfies the following correctness property and EUF-CMA security.

Correctness *For all messages m,*

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

EUF-CMA security *We require the usual EUF-CMA security (Definition 18) to now hold for keys generated by* $\mathsf{QKeyGen}$*.*

Definition 30 (Unbounded classical leakage-resilient weak digital signatures)**.** DS *said to be* weak $(*, \ell_q(\lambda))$-leakage-resilient *if for all polynomials $\ell_c(\cdot)$ and all QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$, the advantage of $\mathcal{A}$ is negligible in the following game.*

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

$$R_{\mathsf{sign}} \leftarrow \mathsf{DS.QKeyGen}(sk).$$

2. *The leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ gets access to R_{sign} and vk, it produces a leakage register R_{leak} consisting of $\ell_c(\lambda)$ classical bits and $\ell_q(\lambda)$ qubits.*

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

4. $\mathcal{A}_{\mathsf{Main}}$ *gets* R_{leak}, vk *and* m^**, and it produces a forged signature* s^*.

5. *Output* 1 *iff* $\mathsf{DS.Ver}(vk, m^*, s^*) = 1$.

Definition 31 (Anti-piracy security for digital signatures [CLLZ21, LLQZ22])**.** *A digital signature scheme* DS *is said to satisfy anti-piracy security if for all QPT adversaries* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in the following game is negligible.*

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

$$R_{\mathsf{sign}} \leftarrow \mathsf{DS.QKeyGen}(sk).$$

2. $\mathcal{A}$ *gets access to* R_{sign} *and* vk*, and it produces two QPT programs* $\mathcal{A}_1, \mathcal{A}_2$ *and a pair of registers* (R_1, R_2).

3. *The challenger samples two challenge messages* $m_1^*, m_2^* \leftarrow \mathcal{M}$.

4. *The challenger runs* $s_1 \leftarrow \mathcal{A}_1(R_1, m_1^*)$ *and* $s_2 \leftarrow \mathcal{A}_2(R_2, m_2^*)$.

5. *Output* 1 *if and only if both* $\mathsf{DS.Ver}(vk, m_1^*, s_1) = 1$ *and* $\mathsf{DS.Ver}(vk, m_2^*, s_2) = 1$.

Theorem 20 ([CLLZ21])**.** *Assuming post-quantum subexponentially secure indistinguishability obfuscation and subexponentially secure LWE, there exists a digital signature scheme that satisfies anti-piracy security.*

We show that any digital signature scheme with anti-piracy security is also $(*, 0)$-leakage-resilient. Proof follows a similar structure to Theorem 18.

Theorem 21. *Let* DS *be a digital signature scheme satisfying anti-piracy security. Then,* DS *is also* $(*, 0)$*-leakage-resilient.*

Proof. Suppose for a contradiction that there is an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins the leakage game with probability $\frac{1}{p(\lambda)}$ for some polynomial $p(\cdot)$ and infinitely many values of λ. We then construct an adversary as follows and show that $\mathcal{A}^{'}$ wins the anti-piracy game with non-negligible probability. Given R_{sign}, the adversary $\mathcal{A}^{'}$ runs $\mathcal{A}_{\mathsf{Leak}}$ on it to produce a classical string L. Then, it outputs $(\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Main}})$ and (L, L). Observe that conditioned on fixed values of the leakage and the keys, the probability of winning the anti-piracy game is $\Pr[\mathsf{DS.Ver}(vk, m^*, \mathcal{A}_{\mathsf{Main}}(L, m^*)) = 1 : m^* \leftarrow \mathcal{M}]^2$ since the signing challenges, the piracy adversaries and calls to the verification algorithm are all independent. Hence, we can write the winning probability for the anti-piracy game as

$$p_{\mathsf{piracy}} = \mathbb{E}_{sk,vk,L}\Big[\Pr[\mathsf{DS.Ver}(vk, m^*, \mathcal{A}_{\mathsf{Main}}(L, m^*)) = 1 : m^* \leftarrow \mathcal{M}]^2\Big].$$

Similarly we can write the probability of winning the leakage game as

$$p_{\mathsf{leak}} = \mathbb{E}_{sk,vk,L}[\Pr[\mathsf{DS.Ver}(vk, m^*, \mathcal{A}_{\mathsf{Main}}(L, m^*)) = 1 : m^* \leftarrow \mathcal{M}]].$$

Then, by Jensen's inequality we see that $p_{\mathsf{piracy}} \geq (p_{\mathsf{leak}})^2 = \frac{1}{p^2(\lambda)}$, which shows that $\mathcal{A}^{'}$ has a non-negligible probability of winning the anti-piracy game. □

Corollary 4. *Assuming post-quantum subexponentially secure indistinguishability obfuscation and subexponentially secure LWE, there exists a digital signature scheme that is* $(*, 0)$*-leakage-resilient.*

Proof. Invoke Theorem 20 and Theorem 21. □

3.4 Message authentication codes

In this section, we introduce (weak) message authentication codes resilient against unbounded classical leakage. In our definition, we will require that an adversary that obtains an unbounded amount of classical leakage on both the tagging and verification keys cannot tag a random message. Note that with classical tags, we cannot hope to achieve the stronger, selective or adaptive security notions where the the message is chosen by the adversary, since then the adversary can just leak a tag for a message of its choice. We show that the well-known MAC construction from PRFs is unbounded classical leakage-resilient when the underlying PRF scheme is. Therefore, by instantiating this construction with our wPRF scheme from Section 3.2, we give unbounded classical leakage-resilient (weak) MACs.

Definition 32 (Unbounded classical leakage-resilient weak MAC)**.** *A message authentication code* MAC *consists of the following QPT algorithms.*

- $\mathsf{Setup}(1)$: *Outputs a classical key* k*.*
- $\mathsf{QKeyGen}(k)$: *Takes the key* k*, outputs tagging and verification key registers* $R_{\mathsf{tag}}, R_{\mathsf{ver}}$ *in some product state.*
- $\mathsf{Tag}(R_{\mathsf{tag}}, m)$*: Takes a tagging key and a message* m*, returns a tag for* m*.*
- $\mathsf{Ver}(R_{\mathsf{ver}}, m, s)$*: Takes a verification key, a message* m *and a (supposed) tag* s *for* m*, returns* 1 *if* s *is a valid tag for* m*.*

It satisfies the following correctness property.

Correctness *For all messages* m*,*

$$\Pr\left[\mathsf{Ver}(R_{\mathsf{ver}}, m, s) = 1 : \begin{array}{c} k \leftarrow \mathsf{Setup}(1) \\ R_{\mathsf{tag}}, R_{\mathsf{ver}} \leftarrow \mathsf{QKeyGen}(k) \\ s \leftarrow \mathsf{Tag}(R_{\mathsf{tag}}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

MAC *said to be* weak $(*, \ell_q(\lambda))$-leakage-resilient *if for all polynomials* $\ell_c(\cdot)$ *and all QPT adversaries* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak,tag}}, \mathcal{A}_{\mathsf{Leak,ver}})$*, the advantage of* $\mathcal{A}$ *is negligible in the following game.*

1. *The challenger runs* $k \leftarrow \mathsf{MAC.Setup}(1)$*, and then*

$$R_{\mathsf{tag}}, R_{\mathsf{ver}} \leftarrow \mathsf{MAC.QKeyGen}(k)$$
$$R'_{\mathsf{tag}}, R'_{\mathsf{ver}} \leftarrow \mathsf{MAC.QKeyGen}(k)$$

2. *The leakage adversaries* $\mathcal{A}_{\mathsf{Leak,tag}}$ *and* $\mathcal{A}_{\mathsf{Leak,ver}}$ *get access to* R_{tag} *and* R_{ver} *respectively and they produce leakage registers* $R_{\mathsf{leak,tag}}$ *and* $R_{\mathsf{leak,ver}}$*, each consisting of* $\ell_c(\lambda)$ *classical bits and* $\ell_q(\lambda)$ *qubits.*
3. *The challenger samples a challenge message* $m^* \leftarrow \mathcal{M}$*.*
4. $\mathcal{A}_{\mathsf{Main}}$ *gets* $R_{\mathsf{leak,tag}}, R_{\mathsf{leak,ver}}$ *and* m^**, and it produces a forged tag* s^**.*
5. *Output* 1 *iff* $\mathsf{MAC.Ver}(R'_{\mathsf{tag}}, m^*, s^*) = 1$*.*

In our security definition, we create two verification keys and require that the forged tag passes verification with respect to the undisturbed key. This is because an adversary measuring the verification key will collapse it to a state that is known to it, and hence it can produce tags that will pass the verification with respect to the collapsed key. However, one can construct schemes where leakage can be detected, after which we can reject all tags until new keys are established. See Section 4.

Theorem 22. *Assuming existence of a $(*, \ell(\lambda))$-leakage-resilient unpredictable PRF protection scheme, there exists a $(*, \ell(\lambda))$-leakage-resilient weak MAC.*

Proof. Let PRF be a $(*, \ell(\lambda))$-leakage-resilient unpredictable PRF. We claim the following construction MAC is a $(*, \ell(\lambda))$-leakage-resilient MAC.

MAC

- $\mathsf{MAC.Setup}(1)$
  1. Output $\mathsf{PRF.Setup}(1)$
- $\mathsf{MAC.KeyGen}(k)$
  1. Output $\mathsf{PRF.KeyGen}(k, 1^2)$
- $\mathsf{MAC.Tag}(R_{\mathsf{tag}}, m)$
  1. Output $\mathsf{PRF.Eval}(R_{\mathsf{tag}}, m)$
- $\mathsf{MAC.Verify}(R_{\mathsf{ver}}, m, s)$
  1. Output 1 iff $\mathsf{PRF.Eval}(R_{\mathsf{ver}}, m) = s$

It is easy to see that correctness is satisfied by the correctness of PRF. Observe that the leakage-resilient forgery game is exactly the same as the leakage-resilient wPRF unpredictibility game (Definition 27) with $t = 2$ played by a (weaker) adversary running independent circuits on the two copies of the PRF key. Therefore, security of MAC follows by the security of PRF. □

3.5 Public-key encryption

In this section, we define unbounded classical leakage resilient schemes for public-key encryption, and show that we can construct them using public-key encryption schemes with the unclonable decryption property [CLLZ21]. As a stepping stone, we first define and construct a weaker variant where we only require that an adversary cannot predict the message given the ciphertext, public key and the leakage from the secret key. Then, using extractors (more specifically, Goldreich-Levin bits), we construct a scheme that satisfies the stronger, indistinguishability-based leakage-resilience.

Definition 33 (Public-key encryption)**.** *A public-key encryption scheme PKE consists of the following algorithms.*

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

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

We require that the scheme satisfies correctness and the usual CPA security.

Correctness *For all messages m,*

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

Similar to PRFs, we assume that Dec is rewound after each use, and hence we can decrypt polynomially many messages.

Definition 34 (Unbounded classical leakage-resilient unpredictable public-key encryption)**.** PKE *is said to be* unpredictable $(*, \ell_q(\lambda))$-leakage-resilient *if for all polynomials $\ell_c(\cdot)$ and all QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$, the advantage of $\mathcal{A}$ is negligible in the following game.*

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

$$R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QKeyGen}(sk).$$

2. *The leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ gets access to R_{dec} and pk, it produces a leakage register R_{leak} consisting of $\ell_c(\lambda)$ classical bits and $\ell_q(\lambda)$ qubits.*
3. *The challenger samples a challenge message $m^* \leftarrow \mathcal{M}$ and encrypts it $ct \leftarrow \mathsf{PKE.Enc}(pk, m^*)$.*
4. *$\mathcal{A}_{\mathsf{Main}}$ gets R_{leak}, pk and ct, and it produces a prediction m'.*
5. *Output 1 iff $m' = m^*$.*

Definition 35 (Unbounded classical leakage-resilient public-key encryption)**.** *A public-key encryption scheme* PKE *is said to be* $(*, \ell_q(\lambda))$-leakage-resilient *if for all polynomials $\ell_c(\cdot)$ and all (stateful) QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$, the advantage of $\mathcal{A}$ is negligible in the following game.*

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

$$R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QKeyGen}(sk)$$

2. *The leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ gets access to R_{dec} and pk, it produces a leakage register R_{leak} consisting of $\ell_c(\lambda)$ classical bits and $\ell_q(\lambda)$ qubits.*
3. *$\mathcal{A}_{\mathsf{Main}}$ gets R_{leak}, pk, it outputs two messages m_0, m_1.*
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 ct, it outputs a prediction b'.*
6. *Output 1 iff $b' = b$.*

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

Definition 36 (Anti-piracy security for public-key encryption - random challenge plaintexts [CLLZ21])**.** *A public-key encryption scheme* PKE *is said to satisfy* anti-piracy security with random challenge plaintexts *if for all QPT adversaries* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in the following game is negligible.*

1. *The challenger runs* $sk, pk \leftarrow \mathsf{PKE.Setup}(1)$*, and then*
$$R_{\mathsf{dec}} \leftarrow \mathsf{PKE.QKeyGen}(sk).$$
2. $\mathcal{A}$ *gets access to* R_{dec} *and* pk*, and it produces two QPT programs* $\mathcal{A}_1, \mathcal{A}_2$ *and a pair of registers* (R_1, R_2)*.*
3. *The challenger samples two challenge messages* $m_1^*, m_2^* \leftarrow \mathcal{M}$*.*
4. *The challenger computes* $ct_1 \leftarrow \mathsf{PKE.Enc}(pk, m_1^*)$ *and* $ct_2 \leftarrow \mathsf{PKE.Enc}(pk, m_2^*)$*.*
5. *The challenger runs* $m_1' \leftarrow \mathcal{A}_1(R_1, ct_1)$ *and* $m_2' \leftarrow \mathcal{A}_2(R_2, ct_2)$*.*
6. *Output* 1 *if and only if both* $m_1{}' = m_1^*$ *and* $m_2{}' = m_2^*$*.*

Theorem 23 ([CLLZ21])**.** *Assuming the existence of post-quantum sub-exponentially secure iO and one-way functions, the quantum hardness of LWE, there exists a public-key encryption scheme that satisfies anti-piracy security with random challenge plaintexts.*

Theorem 24. *Let* PKE *be a public-key encryption scheme satisfying anti-piracy security with random challenge plaintexts. Then,* PKE *is unpredictable* $(*, 0)$*-leakage-resilient.*

Proof. An argument similar to proofs of Theorem 18 and Theorem 21 yields the result. Namely, if there is a classical leakage attack that can distinguish ciphertexts with non-negligible probability, then we can construct a cloning attack that *leaks* on the secret key, and *clones* the classical leakage to obtain two registers that can both distinguish ciphertext with non-negligible probability. □

Theorem 25. *Suppose there exists a public-key encryption scheme* PKE'*, encrypting plaintexts of size* $m(\lambda)$ *into ciphertexts of* $c(\lambda)$*, that is unpredictable* $(*, 0)$*-leakage-resilient. Then, for any polynomial* $p(\cdot)$*, the following public-key encryption scheme satisfies* $(*, 0)$*-leakage-resilience. Furthermore, it encrypts plaintexts of size* $p(\lambda)$ *into ciphertexts of size* $p(\lambda) \cdot (m(\lambda) + c(\lambda) + 1)$ *and it has the same key size as* PKE'*.*

$\underline{\mathsf{PKE}}$

- $\underline{\mathsf{PKE.Setup}(1)}$
  1. *Output* $\mathsf{PKE}'.\mathsf{Setup}(1)$*.*
- $\underline{\mathsf{PKE.QKeyGen}(sk)}$
  1. *Output* $\mathsf{PKE}'.\mathsf{QKeyGen}(sk)$*.*
- $\underline{\mathsf{PKE.Enc}(pk, m)}$
  1. *For each* $i \in [p(\lambda)]$*, sample* $k_i, r_i \leftarrow \{0,1\}^{m(\lambda)}$*.*
  2. *Output* $(\mathsf{PKE}'.\mathsf{Enc}(pk, k_i), r_i, \langle k_i, r_i \rangle \oplus \langle m, e_i \rangle)_{i \in [p(\lambda)]}$*.*
- $\underline{\mathsf{PKE.Dec}(R_{\mathsf{dec}}, (ct_i, r_i, b_i)_{i \in [p(\lambda)]})}$

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

Proof. 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 overwhelming correctness of $\mathsf{PKE}'.\mathsf{Dec}$ and by key rewinding, we can correctly decrypt polynomially many messages with all but negligible probability.

We claim PKE is (indistinguishable) $(*, 0)$-leakage-resilient. For each $j \in \{0, 1, \ldots, p(\lambda)\}$, define the hybrid game Hyb_j as follows by modifying the indistinguishability leakage-resilience game. In $\mathsf{PKE}.\mathsf{Enc}$, replace

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

with

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

Observe that Hyb_0 is the original indistinguishability leakage-resilience game. It is easy to see that in $\mathsf{Hyb}_{p(\lambda)}$, the advantage of any adversary is 0 since each bit of the message is encrypted with a one-time pad key that is independent from everything else. 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. Then, similar to the construction in the proof of Theorem 19, using $\mathcal{A}$ one can construct an adversary $\mathcal{B}$ that can predict $\langle k_{j+1}, r \rangle$ with probability $\frac{1}{2} + \frac{1}{w(\lambda)}$, given $\mathsf{PKE}'.\mathsf{Enc}(pk, k_{j+1})$, the leakage $\mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{dec}})$ and a random r. Finally, as in Theorem 19, using $\mathcal{B}$ we can construct an adversary that predicts a random k_{j+1} with non-negligible probability given $\mathsf{PKE}'.\mathsf{Enc}(pk, k_{j+1})$ and the leakage. This violates the random challenge anti-piracy security of PKE', which is a contradiction. Note that while we need to run $\mathcal{B}$ with the leakage multiple times, it is possible since the leakage is classical.

□

Corollary 5. *Assuming the existence of post-quantum sub-exponentially secure iO and one-way functions, the quantum hardness of LWE, there exists a public-key encryption scheme that is* $(*, 0)$*-leakage-resilience.*

Proof. Invoke Theorem 23, Theorem 24 and Theorem 25. □

3.6 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 them in two different ways: first using PKE assumptions and using our results regarding leakage from BB84 states (Lemma 11, Lemma 12), and second based on unbounded classical leakage-resilient PRFs (which we construct in Section 3.2 based on iO, LWE and one-way function assumptions).

Definition 37 (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* $\mathcal{A}_{\mathsf{Leak,enc}}, \mathcal{A}_{\mathsf{Leak,dec}}$ *consist 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* $R_{\mathsf{enc}}, R_{\mathsf{dec}} \leftarrow \mathsf{SKE}.\mathsf{KeyGen}(1)$.

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* $\left|\Pr[\mathsf{Game}_{\mathcal{A}} = 1] - \frac{1}{2}\right|$.

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. Similar to MACs (Definition 32), 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 leakage-detection schemes (Section 4) to throw away the key after a leakage attack and establish new keys.

We present two schemes that satisfy this notion. The first construction is based on public-key assumptions. The second construction relies on wPRFs that can tolerate unbounded classical leakage, which we construct in Section 3.2.

Theorem 26. *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* PKE.KeyGen *has randomness complexity* $r(\lambda)$*. Let* Ext *be the extractor obtained by instantiating Theorem 13 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

- 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 13, 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 12), 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 13), 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 1), 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. □

Theorem 27. *Let $m(\lambda)$ denote the message size. Let* PRF *be a $(*, 0)$-leakage-resilient PRF scheme with input size $g(\lambda)$ and output size $m(\lambda)$, which exists by Corollary 3. Then, the following private-key encryption scheme* SKE *is $(*, 0)$-leakage-resilient.*

- $\underline{\mathsf{SKE.KeyGen}(1)}$
  1. $k \leftarrow \mathsf{PRF.KeyGen}(1)$.
  2. $R_{\mathsf{enc}}, R_{\mathsf{dec}} \leftarrow \mathsf{PRF.QKeyGen}(k, 1^2)$.
  3. *Output* $(R_{\mathsf{enc}}, R_{\mathsf{dec}})$.
- $\underline{\mathsf{SKE.Enc}(R_{\mathsf{enc}}, m)}$
  1. *Sample* $r \leftarrow \{0,1\}^{g(\lambda)}$
  2. *Output* $(r, m \oplus \mathsf{PRF.Eval}(R_{\mathsf{enc}}, r))$.
- $\underline{\mathsf{SKE.Dec}(R_{\mathsf{dec}}, (r, a))}$
  1. *Output* $a \oplus \mathsf{PRF.Eval}(R_{\mathsf{dec}}, r)$.

Proof. It is easy to see that, by multi-challenge security of PRF (Definition 28, Theorem 17), for any polynomial $p(\cdot)$ and sequence of messages $m = (m^{(i)})_{i\in[p(\lambda)]}$, we have

$$r_1, \ldots, r_{p(\lambda)}, (m^{(i)} \oplus \mathsf{PRF.Eval}(R_{\mathsf{enc}}, r_i))_{i\in[p(\lambda)]}, \mathcal{A}'_{\mathsf{Leak}}((R_{\mathsf{enc}}, R_{\mathsf{dec}}))$$
$$\approx$$
$$r_1, \ldots, r_{p(\lambda)}, U^{(1)}, \ldots, U^{(p(\lambda))}, \mathcal{A}'_{\mathsf{Leak}}((R_{\mathsf{enc}}, R_{\mathsf{dec}})).$$

where we define the adversary

$$\mathcal{A}'_{\mathsf{Leak}}((R_{\mathsf{enc}}, R_{\mathsf{dec}})) = (\mathcal{A}_{\mathsf{Leak,enc}}(R_{\mathsf{enc}}), \mathcal{A}_{\mathsf{Leak,dec}}(R_{\mathsf{dec}})).$$

Then, by applying this to m_0 and m_1, and by hybrid lemma we get

$$r_1, \ldots, r_{p(\lambda)}, (m_0^{(i)} \oplus \mathsf{PRF.Eval}(R_{\mathsf{enc}}, r_i))_{i\in[p(\lambda)]}, \mathcal{A}_{\mathsf{Leak,enc}}(R_{\mathsf{enc}}), \mathcal{A}_{\mathsf{Leak,dec}}(R_{\mathsf{dec}})$$
$$\approx$$
$$r_1, \ldots, r_{p(\lambda)}, (m_1^{(i)} \oplus \mathsf{PRF.Eval}(R_{\mathsf{enc}}, r_i))_{i\in[p(\lambda)]}, \mathcal{A}_{\mathsf{Leak,enc}}(R_{\mathsf{enc}}), \mathcal{A}_{\mathsf{Leak,dec}}(R_{\mathsf{dec}}).$$

This is equivalent to

$$\mathsf{SKE.Enc}(R_{\mathsf{enc}}, m_0^{(1)}), \ldots, \mathsf{SKE.Enc}(R_{\mathsf{enc}}, m_0^{(p(\lambda))}), \mathcal{A}_{\mathsf{Leak,enc}}(R_{\mathsf{enc}}), \mathcal{A}_{\mathsf{Leak,dec}}(R_{\mathsf{dec}}),$$
$$\approx$$
$$\mathsf{SKE.Enc}(R_{\mathsf{enc}}, m_1^{(1)}), \ldots, \mathsf{SKE.Enc}(R_{\mathsf{enc}}, m_1^{(p(\lambda))}), \mathcal{A}_{\mathsf{Leak,enc}}(R_{\mathsf{enc}}), \mathcal{A}_{\mathsf{Leak,dec}}(R_{\mathsf{dec}}),$$

which proves the security of SKE. □

We note that the construction in Theorem 27 is still secure in the stronger, CPA-style adaptive leakage security model where both of the leakage adversaries can obtain encryptions of messages of their choice before producing their leakage. Furthermore, the construction is still secure if the key leakage adversaries are entangled, or have a quantum channel between. These results follow from an argument similar to the proof of Theorem 27, using the 2-copy leakage-resilience of the underlying PRF, so we omit them.

4 Cryptographic schemes with leakage-detection

In this section, we initiate the study of cryptographic primitives with leakage-detection, yet another set of schemes that are only possible through utilization of quantum phenomena. We show a generic way of constructing such primitives from schemes with *publicly verifiable* certified deletion, and explicitly show such constructions for public-key encryption, digital signatures, functional encryption, differing-inputs indistinguishability obfuscation, and software.

In our leakage-detection models, we will require that if our detection algorithm does not detect leakage after an attack, then the adversary should have negligible advantage in breaking 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 leakage attack, then testing for leakage only negligibly disturbs the key. This means that the honest party can test for leakage a polynomial number of times while preserving the correctness and security guarantees, as long as there is no leakage attack.

Now, we describe our transformation on a high level. Suppose there exists a scheme with certified deletion. We will construct a $\mathsf{TestLeakage}$ algorithm that essentially tries to produce a deletion certificate for the secret, and outputs $\mathsf{NO\ LEAKAGE}$ if it succeeds. Intuitively, we can argue leakage-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 pretends to leak on their secret, produces a valid deletion certificate using the leftover state, and still succeeds in breaking the security guarantee using the *leak*. 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 9, 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 leakage-detection procedure will skip this measurement, and will instead run the verification procedure *coherently*. Furthermore, the leakage-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 1. *Note that public verification in the certified deletion scheme is essential to build a leakage detection scheme, since the leakage adversary will have access to the complete state of the honest parties.*

4.1 Public-key encryption with leakage-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 how to construct them from public-key encryption schemes with secure key leasing. To avoid repetition, we will only discuss how to construct our schemes from PKE scheme with secure key leasing that have classical certificates, however, our results also hold for schemes with quantum certificates, with essentially the same construction.

Definition 38 (Public-key encryption with leakage-detection)**.** *A public-key encryption scheme with leakage-detection is a public-key encryption scheme (Definition 33) with the following additional algorithms that satisfy the reusability and security guarantees below.*

- $\mathsf{QLKeyGen}(sk)$: *Along with a quantum decryption key* R_{dec}[6]*, also outputs a classical leakage-detection key* tk*.*
- $\mathsf{TestLeakage}(tk, R_{dec})$: *Takes the leakage-detection key and the decryption key, outputs* LEAKED *if* leakage is detected*,* NO LEAKAGE *otherwise.*

PKE correctness and security: *We require the usual correctness and security (Definition 33) satisfied for the decryption key generated by* QKeyGen *to now also hold for the decryption key generated by* QLKeyGen*.*

Detection correctness:

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

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* TestLeakage *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 9.

Leakage-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 leakage adversary* $\mathcal{A}_{\mathsf{Leak}}$ *gets access to* R_{dec}, tk*, and* pk*, and produces a leakage register* R_{leak} *and two challenge messages* m_0, m_1*, along with the* updated *register* R_{dec}*.*
3. *The challenger runs* $tb \leftarrow \mathsf{PKE.TestLeakage}(tk, R_{\mathsf{dec}})$*. If* tb *is* LEAKED*, 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* $R_{\mathsf{leak}}, m_0, m_1, tk, pk$ *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* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *wins the game with probability at most*

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

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.

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

Definition 39 (Public-key encryption with secure key leasing)**.** *A public-key encryption scheme with secure key leasing is a public-key encryption scheme (Definition 33) 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.*
- $\mathsf{Verify}(cvk, cert)$: *A* classical algorithm *that 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 33) satisfied for the decryption key generated by* QKeyGen *to now also hold for the decryption key generated by* QVKeyGen*.*

Verification correctness:

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

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* $cert$*, a state register* R *and two challenge messages* m_0, m_1*.*
3. *The challenger runs* $vb \leftarrow \mathsf{PKE.Verify}(cvk, cert)$*. 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* $R, cert, m_0, m_1, cvk, pk$ *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 leakage-detection.

Theorem 28. *Suppose there exists a public-key encryption scheme with publicly verifiable secure key leasing. Then, there exists a public-key encryption scheme with leakage-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 $\mathsf{PKE}'.\mathsf{QVKeyGen}$.

Throughout the proof, associate $\mathsf{NO\ LEAKAGE}$ and VALID with 1, LEAKED and $\mathsf{INVALID}$ with 0. Now, we will show how to implement $\mathsf{TestLeakage}$ in a way that satisfies reusability. By Definition 1, a quantum algorithm with a classical output is as a unitary applied to the input and some ancilla in the state $|0\rangle^{\otimes a}$, with some of the output wires being traced out and the rest being measured in the computational basis. Therefore, let $\mathsf{PKE}'.\mathsf{Cert}$ be such that it introduces an ancilla register $R_{\mathsf{anc},1}$ of size d_1 in the state $|0\rangle^{\otimes d_1}$, it applies a unitary U_{Cert} to $R_{\mathsf{dec}}, R_{\mathsf{anc},1}$ to produce registers $R_{\mathsf{cert}}, R_{\mathsf{garbage},1}$ where $R_{\mathsf{garbage},1}$ is traced out and R_{cert} is measured. Let U_{Ver} be the unitary that implements the mapping

$$|b\rangle|tk\rangle|x\rangle|r\rangle|0\rangle^{d_2} \mapsto \Big|b \oplus \mathsf{PKE}'.\mathsf{Verify}(tk, x; r)\Big\rangle|tk\rangle|x\rangle|r\rangle|0\rangle^{d_2}.$$

Note that, since $\mathsf{PKE}'.\mathsf{Verify}$ is an efficient classical algorithm, it is possible to implement U_{Ver} efficiently using the uncomputation trick. Finally, let d_3 denote the size of the randomness used by $\mathsf{PKE}'.\mathsf{Verify}$. Then, we define $\mathsf{TestLeakagePre}$ as follows.

$\underline{\mathsf{TestLeakagePre}(\mathsf{tk}, \mathsf{R_{dec}})}$

1. Introduce $R_{\mathsf{anc},1}, R_{\mathsf{anc},2}, R_{\mathsf{rand}}, R_{\mathsf{res}}, R_{\mathsf{tk}}$ in the states $|0\rangle^{\otimes d_1}$, $|0\rangle^{\otimes d_2}$, $|0\rangle^{\otimes d_3}$, $|0\rangle$, and $|tk\rangle$.
2. Apply U_{Cert} to $(R_{\mathsf{dec}}, R_{\mathsf{anc},1})$ to get the registers $R_{\mathsf{cert}}, R_{\mathsf{garbage},1}$.
3. Apply $H^{\otimes d_3}$ to R_{rand}.
4. Apply U_{Ver} to $(R_{\mathsf{res}}, R_{\mathsf{tk}}, R_{\mathsf{cert}}, R_{\mathsf{rand}}, R_{\mathsf{anc},2})$.
5. Trace out the ancilla registers, measure R_{res} in computational basis and output the resulting bit.

$\mathsf{TestLeakagePre}$ consists of a unitary and then the projective measurement $\{|0\rangle\langle 0|^{R_{\mathsf{res}}} \otimes I, |1\rangle\langle 1|^{R_{\mathsf{res}}} \otimes I\}$. Then, consider the following algorithm $\mathsf{TestLeakage}$ we get from Lemma 9, where U_{Test} denotes the unitary part of $\mathsf{TestLeakagePre}$.

$\underline{\mathsf{TestLeakage}(\mathsf{tk}, \mathsf{R_{dec}})}$

1. Introduce $R_{\mathsf{anc},1}, R_{\mathsf{anc},2}, R_{\mathsf{rand}}, R_{\mathsf{res}}, R_{\mathsf{tk}}$ in the states $|0\rangle^{\otimes d_1}$, $|0\rangle^{\otimes d_2}$, $|0\rangle^{\otimes d_3}$, $|0\rangle$, and $|tk\rangle$.
2. Apply U_{Test}.
3. Measure R_{res} in computational basis to get a bit res.
4. Apply U_{Test}^{-1} and trace out the ancilla registers.
5. Output res.

It is easy to see that $\mathsf{TestLeakagePre}(tk, R)$ (and hence $\mathsf{TestLeakagePre}(tk, R)$) has the same output distribution as $\mathsf{PKE}'.\mathsf{Verify}(vk, \mathsf{PKE}'.\mathsf{Cert}(R))$ 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

$$\Pr[\mathsf{PKE.TestLeakage}(tk, R_{\mathsf{dec}}) = \mathsf{NO\ LEAKAGE}] = 1. \tag{12}$$

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 $\mathsf{PKE.TestLeakage}(tk, R_{\mathsf{dec}})$, respectively. Then, by Equation (12) and Lemma 9, we get

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

which shows that PKE satisfies reusability.

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

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

for some polynomial $p(\cdot)$ and infinitely many values of λ where we let E_{LEAKED} be the event that the output of $\mathsf{TestLeakage}$ is LEAKED. 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 $R_{\mathsf{leak}}, m_0, m_1 \leftarrow \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{dec}}, cvk, pk)$.[7]
2. Run $cert \leftarrow \mathsf{PKE}'.\mathsf{Cert}(R_{\mathsf{dec}})$.
3. Output $cert, R_{\mathsf{leak}}, m_0, m_1$.

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

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 leakage-detection game played by $\mathcal{A}$. Observe that they are exactly the same, except for the following differences. In the leakage-detection game, the produced certificate is not measured, but Verify is instead run coherently. Furthermore, we rewind $\mathsf{TestLeakage}$ in the leakage detection game, while in the key leasing game, the challenger does not apply the same rewinding. However, crucially, in the leakage-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[8], we can see that the leftover state of R_{leak} produced by $\mathcal{A}$ in the leakage-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 $\mathsf{TestLeakagePre}(tk, R)$ 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,

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

[8] The game terminates early on when $\mathsf{TestLeakage}$ outputs LEAKED in the leakage-detection game or when Verify outputs $\mathsf{INVALID}$ in the key leasing game.

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 leakage-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,

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

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

We remark that, as discussed in Section 3.5, assuming the existence of the quantum hardness of LWE, one-way functions and post-quantum sub-exponentially secure iO, one can construct PKE schemes with unclonable keys, which is a stronger security notion than PKE with publicly-verifiable key leasing

In addition, 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 leakage-detection since the leakage adversary gets access to the complete state of the honest party, which includes the leakage-detection key.

4.2 Digital signature schemes with leakage-detection

We construct digital signatures with leakage-detection from any digital signature scheme with key leasing, using essentially the same technique as used for PKEs. 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 leakage game with a selective challenge message, a leakage adversary can sign a message of its choice (which only negligibly disturbs the key) and leak the signature.

Definition 40 (Digital signatures with leakage-detection)**.** *A digital signature scheme with leakage-detection is a digital signature scheme (Definition 29) 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 leakage-detection key tk.*
- $\mathsf{TestLeakage}(tk, R_{sign})$: *Takes the leakage-detection key and the signing key, outputs* LEAKED *if* leakage is detected*,* $\mathsf{NO\ LEAKAGE}$ *otherwise.*

Digital signature correctness and security *We require the usual correctness and security (Definition 29) satisfied for the signing key generated by* $\mathsf{QKeyGen}$ *to now also hold for the signing key generated by* $\mathsf{QLKeyGen}$*.*

Detection correctness

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

Reusability after testing *Initialize the signing register,* $R_{\mathsf{sign}}, tk \leftarrow \mathsf{QLKeyGen}(sk)$ *and let* ρ *denote its state. Run the algorithm* TestLeakage *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 9.

Leakage-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 leakage adversary* $\mathcal{A}_{\mathsf{Leak}}$ *gets access to* R_{sign}, tk *and* vk *and it produces a leakage register* R_{leak}*, along with the updated register* R_{sign}*.*
3. *The challenger runs* $tb \leftarrow \mathsf{DS.TestLeakage}(tk, R_{\mathsf{sign}})$*. If* tb *is* LEAKED*, challenger outputs* 0 *and terminates.*
4. *The challenger samples a challenge message* $m \leftarrow \mathcal{M}$*.*
5. $\mathcal{A}_{\mathsf{Main}}$ *gets* $R_{\mathsf{leak}}, tk, vk$ *and* m*, it outputs a forged signature* s*.*
6. *The challenger outputs* 1 *if* $\mathsf{DS.Verify}(m, s) = 1$*.*

Definition 41 (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.*
- $\mathsf{VerifyDeletion}(cvk, cert)$: *A* classical algorithm *that 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 (Definition 29) satisfied for keys generated by* QKeyGen *to now also hold for keys generated by* QVKeyGen*.*

Verification correctness

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

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* $cert$ *and a state register* R.
3. *The challenger runs* $vb \leftarrow \mathsf{VerifyDeletion}(cvk, cert)$. *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* $R, cert, cvk, vk$ *and* m, *it outputs a forged signature* s.
6. *The challenger outputs* 1 *if* $\mathsf{DS.Ver}(m, s) = 1$.

Theorem 29. *Suppose there exists a digital signature scheme with publicly verifiable secure key leasing. Then, there exists a digital signature scheme with leakage-detection.*

Proof. The construction and the reduction are essentially the same as Theorem 28. 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 $\mathsf{DS.QLKeyGen}$ be the same as $\mathsf{DS}'.\mathsf{QVKeyGen}$. Let U_{Cert} be the unitary part of $\mathsf{DS}'.\mathsf{Cert}$ and let U_{Ver} be the unitary that implements the mapping
$$|b\rangle|tk\rangle|x\rangle|r\rangle|0\rangle^{d_2} \mapsto \Big|b \oplus \mathsf{DS}'.\mathsf{VerifyDeletion}(tk, x; r)\Big\rangle|tk\rangle|x\rangle|r\rangle|0\rangle^{d_2}.$$

Define $\mathsf{TestLeakagePre}$ as follows.

$\underline{\mathsf{TestLeakagePre}(\mathsf{tk}, \mathsf{R_{sign}})}$

1. Introduce $R_{\mathsf{anc},1}, R_{\mathsf{anc},2}, R_{\mathsf{rand}}, R_{\mathsf{res}}, R_{\mathsf{tk}}$ in the states $|0\rangle^{\otimes d_1}$, $|0\rangle^{\otimes d_2}$, $|0\rangle^{\otimes d_3}$, $|0\rangle$, and $|tk\rangle$.
2. Apply U_{Cert} to $(R_{\mathsf{sign}}, R_{\mathsf{anc},1})$ to get the registers $R_{\mathsf{cert}}, R_{\mathsf{garbage},1}$.
3. Apply $H^{\otimes d_3}$ to R_{rand}.
4. Apply U_{Ver} to $(R_{\mathsf{res}}, R_{\mathsf{tk}}, R_{\mathsf{cert}}, R_{\mathsf{rand}}, R_{\mathsf{anc},2})$.
5. Trace out the ancilla registers, measure R_{res} in computational basis and output the resulting bit.

Then, let U_{Test} be the unitary part of $\mathsf{TestLeakagePre}$, and construct $\mathsf{TestLeakage}$ as follows.

$\underline{\mathsf{TestLeakage}(\mathsf{tk}, \mathsf{R}_{\mathsf{sign}})}$

1. Introduce $R_{\mathsf{anc},1}, R_{\mathsf{anc},2}, R_{\mathsf{rand}}, R_{\mathsf{res}}, R_{\mathsf{tk}}$ in the states $|0\rangle^{\otimes d_1}$, $|0\rangle^{\otimes d_2}$, $|0\rangle^{\otimes d_3}$, $|0\rangle$, and $|tk\rangle$.
2. Apply U_{Test}.
3. Measure R_{res} in computational basis to get a bit res.
4. Apply U_{Test}^{-1} and trace out the ancilla registers.
5. Output res.

Same arguments as in the proof of Theorem 28 show the detection correctness and reusability of DS.

Finally, we argue leakage-detection security. For a contradiction, suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ such that $\mathcal{A}$ wins the leakage-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 $\mathsf{DS}^{'}$.

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

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

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

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

The same arguments as in the proof of Theorem 28 show that $\mathcal{A}^{'}$ wins the key leasing game for $\mathsf{DS}^{'}$ with probability $\frac{1}{p(\lambda)}$, which is a contradiction.

□

As discussed in Section 3.3, assuming post-quantum subexponentially secure indistinguishability obfuscation and subexponentially secure LWE, one can construct digital signature schemes with unclonable signing keys, which is a stronger notion that digital signatures with secure key leasing.

4.3 Functional encryption with leakage-detection

In this section, we introduce the notion of leakage-detection for functional keys of public-key functional encryption schemes, and show how to construct them from 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 function keys.

Definition 42 (Functional encryption with leakage-detection)**.** *A functional encryption scheme with leakage detection for a family of functions $\mathcal{F}$ is a public-key functional encryption scheme (Definition 19) 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 leakage detection key* tk*.*
- $\mathsf{TestLeakage}(tk, R_f)$*: Takes the leakage detection key and functional key, outputs* LEAKED *if leakage is detected,* NO LEAKAGE *otherwise.*

FE correctness and security: *We require the usual functional encryption correctness and security (Definition 19) for keys generated by* $\mathsf{QLKeyGen}$*.*

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

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

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* $\mathsf{TestLeakage}$ *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 9.

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

1. $\mathcal{A}_{\mathsf{Leak}}$ *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* $\mathcal{A}_{\mathsf{Leak}}$ *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_{leak} *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) $tb_i \leftarrow \mathsf{TestLeakage}(tk, R_{f_i})$*.*
   (b) *If* $tb_i = \text{LEAKED}$ *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* $R_{\mathsf{leak}}, (f_i)_{i\in[p(\lambda)]}, m_0, m_1, tk, pk$ *and* ct*, it outputs a prediction* b'*.*
8. *The challenger outputs* 1 *if* $b' = b$*.*

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

Definition 43 (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.*

Theorem 30 ([BGG+23])**.** *Assuming post-quantum indistinguishability obfuscation, public key encryption, and injective one-way functions, there exists functional encryption with selective secret key publicly verifiable certified deletion.*

- $\mathsf{QVKeyGen}(msk, f)$: *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.*
- $\mathsf{Verify}(vk, cert)$: *A classical algorithm that 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 19) for keys generated by* $\mathsf{QVKeyGen}$*.*

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* $b' = b$.

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

Theorem 31. *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 leakage detection.*

Proof. Construction and the security proof are mostly the same as Theorem 28, 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 leakage 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 $\mathsf{FE}'.\mathsf{KeyGen}$.

Now, as in Theorem 28, consider the following algorithm. On input tk, R_f, run the unitary associated with $\mathsf{FE}'.\mathsf{Cert}$ on R_f, and then the unitary associated with $\mathsf{FE}'.\mathsf{Verify}$ on the result, along with tk, and output the measurement outcome for the result register. Define $\mathsf{FE.TestLeakage}$ to be the rewinding version of this algorithm, obtained from Lemma 9. We associate $\mathsf{NO\ LEAKAGE}$ with VALID and LEAKED 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 leakage detection security. Suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ that wins leakage detection game with non-negligible advantage. Then, we define an adversary $\mathcal{A}'$ for the certified deletion game as follows. $\mathcal{A}'$ runs $\mathcal{A}_{\mathsf{Leak}}$ to obtain m_0, m_1 and outputs them. Then, it asks for keys for $f_1, \ldots, f_{p(\lambda)}$, and it runs $\mathcal{A}_{\mathsf{Leak}}$ on these keys. It runs Cert on all of the updated functional key registers, outputs the resulting certificates and keeps R_{leak} 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_{leak}. An argument similar to Theorem 28 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 leakage detection game. Hence, the rewinding of $\mathsf{TestLeakage}$ has no effect. □

Corollary 6 ([BGG+23])**.** *Assuming post-quantum indistinguishability obfuscation, public key encryption, and injective one-way functions, there exists functional encryption with leakage detection.*

Remark 2. *It is easy to see that when the underlying functional encryption scheme has adaptive-function*[9] *security for certified deletion, the leakage detection scheme constructed in Theorem 31 will have leakage 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 leakage detection security.*

4.4 Indistinguishability obfuscation with leakage-detection

In this section, we define leakage-detection for differing-inputs obfuscation with leakage-detection, and show how to construct such schemes from obfuscation with certified deletion [BGG+23].

Definition 44 (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*

$$\Pr\left[C_0(x) \neq C_1(x) : \begin{array}{c}(C_0, C_1, aux) \leftarrow \mathcal{D} \\ x \leftarrow \mathcal{A}(C_0, C_1, aux)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

[9] Functions adaptively chosen by the adversary by interacting with the functional key generator

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 45 (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,*

$$\Pr[i\mathcal{O}.\mathsf{Eval}(R_{\mathsf{obf}}, x) = C(x) : R_{\mathsf{obf}} \leftarrow i\mathcal{O}.\mathsf{Gen}(C)] = 1.$$

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 46 (Differing-inputs obfuscation with leakage-detection)**.** *Let $(\mathcal{C}, \mathcal{D})$ be a differing-inputs circuit family. An obfuscation scheme with leakage 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 leakage detection key tk.*
- $\mathsf{TestLeakage}(tk, R_{\mathsf{obf}})$*: Takes the leakage detection key and the obfuscation, outputs* LEAKED *if leakage is detected,* NO LEAKAGE *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 the following.*

$$\Pr[\mathsf{TestLeakage}(tk, R_{\mathsf{obf}}) = \mathsf{NO\ LEAKAGE} : tk, R_{\mathsf{obf}} \leftarrow \mathsf{QLGen}(C)] \geq 1 - \mathsf{negl}(\lambda).$$

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* TestLeakage *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 9.

Leakage detection security *Any QPT adversary* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *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. $\mathcal{A}_{\mathsf{Leak}}$ *gets access to* $R_{\mathsf{obf}}, C_0, C_1, aux$ *and* tk*, it produces a leakage* R_{leak} *along with the updated register* R_{obf}.
4. *The challenger runs* $tb \leftarrow \mathsf{TestLeakage}(tk, R_{\mathsf{obf}})$*. If* tb *is* LEAKED*, it outputs* 0 *and terminates.*
5. $\mathcal{A}_{\mathsf{Main}}$ *gets* $R_{\mathsf{leak}}, C_0, C_1, aux$ *and* tk*, it produces a guess* b'.
6. *Output* 1 *if* $b' = b$.

We define the advantage of $\mathcal{A}$ *to be* $\left|\Pr[\mathsf{Game}_{\mathcal{A}} = 1] - \frac{\Pr[tb=\text{NO LEAKAGE}]}{2}\right|$.

Definition 47 (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,* 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 the following.*

$$\Pr\left[\mathsf{Verify}(vk, cert) = \mathsf{VALID} : \begin{array}{c} vk, R_{\mathsf{obf}} \leftarrow \mathsf{QGen}(C) \\ cert \leftarrow \mathsf{Cert}(vk, R_{\mathsf{obf}}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

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* INVALID*, it outputs* 0 *and terminates.*
5. $\mathcal{A}$ *gets* R*, it produces a guess* b'.
6. *Output* 1 *if* $b' = b$.

We define the advantage of $\mathcal{A}$ to be $\left|\Pr[\mathsf{Game}_{\mathcal{A}}=1]-\frac{\Pr[vb=\mathsf{VALID}]}{2}\right|$.

Theorem 32 ([BGG+23])**.** *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists differing inputs obfuscation with (publicly-verifiable) certified deletion for polynomially many differing inputs.*

Theorem 33. *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 leakage detection for $(\mathcal{C},\mathcal{D})$.*

Proof. The construction and the proof are mostly the same as Theorem 28 and Theorem 31, so we only sketch it. Let $i\mathcal{O}'$ be an obfuscation scheme as in the theorem statement. We construct a leakage-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 $i\mathcal{O}'.\mathsf{QGen}$.

Now, as in Theorem 28, consider the following algorithm. On input tk, R_{obf}, run the unitary associated with $i\mathcal{O}'.\mathsf{Cert}$ on R_{obf}, and then the unitary associated with $i\mathcal{O}'.\mathsf{Verify}$ on the result, along with tk. Define $i\mathcal{O}.\mathsf{TestLeakage}$ to be the rewinding version of this algorithm, obtained from Lemma 9. We associate NO LEAKAGE with VALID and LEAKED 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 leakage detection security. Suppose there exists an adversary $\mathcal{A}=(\mathcal{A}_{\mathsf{Main}},\mathcal{A}_{\mathsf{Leak}})$ that wins leakage detection game with non-negligible advantage. Then, we define an adversary $\mathcal{A}'$ for the certified deletion game as follows. $\mathcal{A}'$ runs $\mathcal{A}_{\mathsf{Leak}}$ 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_{leak}. An argument similar to Theorem 28 and Theorem 31 shows that $\mathcal{A}'$ wins the certified deletion game with non-negligible advantage. □

Corollary 7. *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists differing inputs obfuscation with leakage detection for polynomially many differing inputs.*

4.5 Leakage-detection for software

In this section, we introduce the notion of leakage-detection for *software*[10], and show 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.

Definition 48 (Leakage-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 leakage-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 leakage-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.*
- $\mathsf{TestLeakage}(tk,R_{\mathsf{prog}})$*: Takes the leakage-detection key and the program register, outputs* LEAKED *if leakage is detected,* NO LEAKAGE *otherwise.*

[10] Modeled as a sample from a distribution on a family of circuits.

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 $C \in \mathcal{C}$,*

$$\Pr\big[\mathsf{TestLeakage}(tk, R_{\mathsf{prog}}) = \mathsf{NO\ LEAKAGE} :\ \ R_{\mathsf{prog}}, tk \leftarrow \mathsf{Gen}(C)\ \big] \geq 1 - \mathsf{negl}(\lambda).$$

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* TestLeakage *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 9.

β-leakage-detection security *For all QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$, we require the following.*

$$\Pr\left[\begin{array}{c} \mathsf{TestLeakage}(tk, R'_{\mathsf{prog}}) = \mathsf{NO\ LEAKAGE} \\ \wedge \\ \forall x \in \{0,1\}^{n(\lambda)}\ \ \Pr[\mathcal{A}_{\mathsf{Main}}(tk, R_{\mathsf{leak}}, x) = C(x)] \geq \beta \end{array} : \begin{array}{c} C \leftarrow \mathcal{D} \\ R_{\mathsf{prog}}, tk \leftarrow \mathsf{Gen}(C) \\ R_{\mathsf{leak}}, R'_{\mathsf{prog}} \leftarrow \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{prog}}, tk) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 49 (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 $C \in \mathcal{C}$,*

$$\Pr\big[\mathsf{Verify}(vk, R_{\mathsf{prog}}) = \mathsf{NO\ LEAKAGE} :\ \ R_{\mathsf{prog}}, vk \leftarrow \mathsf{Gen}(C)\ \big] \geq 1 - \mathsf{negl}(\lambda).$$

β-perfect finite-term strong lessor security with public verification *For all QPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we require the following.*

$$\Pr\left[\begin{array}{c} \mathsf{Verify}(vk, R_1) = \mathsf{VALID} \\ \wedge \\ \forall x \in \{0,1\}^{n(\lambda)}\ \ \Pr[\mathcal{A}_2(vk, R_2, x) = C(x)] \geq \beta \end{array} : \begin{array}{c} C \leftarrow \mathcal{D} \\ R_{\mathsf{prog}}, vk \leftarrow \mathsf{Gen}(C) \\ R_1, R_2 \leftarrow \mathcal{A}_1(R_{\mathsf{prog}}, vk) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Theorem 34 ([BGG+23, Theorem 8.4, Corollary 8.18])**.** *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists finite-term publicly-verifiable strong secure software leasing for pseudorandom functions, evasive functions, random point functions, and compute-and compare circuits.*

Theorem 35. *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 leakage detection scheme for $(\mathcal{C}, \mathcal{D})$.*

Proof. The proof is mostly the same as that of Theorem 28, so we only sketch it.

Let SSL be a secure leasing scheme as in the theorem statement. We construct a leakage-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 $\mathsf{SLD.TestLeakage}$ to be the rewinding version of $\mathsf{SSL.Verify}$, as obtained from Lemma 9, where we associate VALID with $\mathsf{NO\ LEAKAGE}$ and $\mathsf{INVALID}$ with LEAKED.

It is easy to see that SLD satisfies evaluation and detection correctness. By verification correctness of SSL and Lemma 9, SLD satisfies reusability. Finally, we argue detection security as follows. Suppose there exists an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ 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 $R_{\mathsf{leak}}, R'_{\mathsf{prog}} \leftarrow \mathcal{A}_{\mathsf{Leak}}(R_{\mathsf{prog}}, vk)$.
2. Output $R'_{\mathsf{prog}}, R_{\mathsf{leak}}$.

$\underline{\mathcal{A}'_2(vk, R_2, x)}$

1. Output $\leftarrow \mathcal{A}_{\mathsf{Main}}(vk, R_2, x)$.

As in Theorem 28, 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 leakage detection condition for $\mathcal{A}$, therefore probability of $\mathsf{SLD.TestLeakage}$ outputting $\mathsf{NO\ LEAKAGE}$ for $\mathcal{A}$ is the same $\mathsf{SSL.Verify}$ outputting VALID for $\mathcal{A}$. Further, conditioned on the event $\mathsf{NO\ LEAKAGE}$, the adversary $\mathcal{A}_{\mathsf{Main}}$ does not use R'_{prog}, hence rewinding applied by $\mathsf{TestLeakage}$ has no effect. Therefore, R_2 in the lessor security for $\mathcal{A}'$ (conditioned on VALID) will have the same distribution as R_{leak} in the leakage detection security (conditioned on $\mathsf{NO\ LEAKAGE}$). □

Corollary 8. *Assuming post-quantum indistinguishability obfuscation and one-way functions, there exists leakage detection schemes for pseudorandom functions, evasive functions, random point functions, and compute-and compare circuits.*

Proof. Invoke Theorem 34 and Theorem 35. □

5 Cryptographic schemes resilient to leakage attacks with unbounded shared entanglement

5.1 Spooky classical-leakage resilient primitives

In this section, we study various cryptographic primitives that are secure against *spooky classical-leakage*, that is, against adversaries such that the leakage adversary and the main adversary are

allowed to share unbounded entanglement. While it is known that entanglement (or more generally, non-signalling correlations) does not increase the classical (Shannon) capacity of a classical channel, for some classical tasks using classical channels, entanglement allows us to achieve higher *performance* compared to best possible classical strategy [CLMW10]. Similarly, even small amount of quantum side information can break extractors that are secure against classical side information [GKK+09]. Hence, care must be taken and security of spooky classical-leakage-resilient primitives must be proven explicitly using appropriate quantum tools.

We start by showing that the min-entropy of a source decreases by at most ℓ in presence of spooky classical-leakage of size ℓ, generalizing the result of Dodis, Ostrovsky, Reyzin, and Smith [DORS08] to the setting of spooky classical leakage. Then, we continue by proving that the constructions of Hazay, López-Alt, Wee, and Wichs [HLAWW16] can be proven secure against spooky classical-leakage[11] by using quantum-proof extractors.

Lemma 14. *Let X be a classical random variable, Z be quantum side information on X and R_1, R_2 be quantum registers initialized to some possibly entangled state that is independent of X and Z. Let the leakage adversary act on (X, R_1) using a quantum operation $\mathcal{A}_{\mathsf{Leak}}$ with classical output to produce a classical leakage register L of size ℓ. Then,*

$$p_{guess}(X|Z, L, R_2) \leq 2^{\ell} \cdot p_{guess}(X|Z)$$

where guessing probability on the left hand side is with respect to the state after the actions of leakage adversary.

Proof. Instead of a separate leakage $\mathcal{A}_{\mathsf{Leak}}$ and main adversary $\mathcal{A}_{\mathsf{Main}}$, we will instead consider the equivalent setting where the adversary has access to both X and R_1R_2, but we first apply a quantum-to-classical operation $\mathcal{A}_{\mathsf{Leak}} : \mathcal{X} \otimes \mathcal{R}_1 \to \mathcal{H}^{\otimes \ell}$ and then run $\mathcal{A}_{\mathsf{Main}}$ on the result, along with R_2. To this end, introduce a new register B that is also initialized to x. Before the leakage, we can write the state of the register $XZBR_1R_2$ as

$$\sum_x \Pr[X = x] \cdot |x\rangle\langle x| \otimes \xi_x^Z \otimes |x\rangle\langle x| \otimes \rho^{R_1R_2}.$$

Then, we have $(L, R_1, R_2) \leftarrow (\mathcal{A}_{\mathsf{Leak}}^{BR_1} \otimes I^{R_2})(|X\rangle\langle X| \otimes R_1 \otimes R_2)$ and the state of the register $XZ(LR_2)$ after the leakage becomes

$$\begin{aligned}\sigma &= \sum_x \Pr[X = x] \cdot |x\rangle\langle x| \otimes \xi_x^Z \otimes (\mathcal{A}_{\mathsf{Leak}}^{BR_1} \otimes I^{R_2})(|x\rangle\langle x| \otimes \rho^{R_1R_2}) \\ &= \sum_x \Pr[X = x] \cdot |x\rangle\langle x| \otimes \xi_x^Z \otimes \tau_x\end{aligned}$$

where we define $\tau_x = (\mathcal{A}_{\mathsf{Leak}}^{BR_1} \otimes I^{R_2})(|x\rangle\langle x| \otimes \rho^{R_1R_2})$.

Since this is a *cqq* state, by Lemma 3 we then have

$$p_{guess}(X|L, Z, R_2) = 2^{-\mathbf{H}_\infty(X|L,Z,R_2)}$$

Further, since $\mathcal{A}_{\mathsf{Leak}}$ is a quantum-to-classical channel, as argued in the proof of Lemma 12 using Lemma 13, the state of the registers is separable between L and X, Z, R_2. Hence, we can apply Lemma 5 to get

$$H_\infty(X|L, Z, R_2) \geq H_\infty(X|Z, R_2) - \ell.$$

[11] With slightly different parameters.

Now, we see that $\mathbf{H}_\infty(X|Z,R_2) = \mathbf{H}_\infty(X|Z)$ due to no-signalling. More formally, we can proceed as follows. Consider the following *cqq*-state of XZR_2:

$$\sigma^{XZR_2} = (I^{XZ} \otimes \mathrm{Tr}^L \otimes I^{R_2}) \sum_x \Pr[X=x] \cdot |x\rangle\langle x| \otimes \xi_x^Z \otimes (\mathcal{A}_{\mathsf{Leak}}^{BR_1} \otimes I^{R_2})(|x\rangle\langle x| \otimes \rho^{R_1R_2})$$
$$= \sum_x \Pr[X=x] \cdot |x\rangle\langle x| \otimes \xi_x^Z \otimes (\mathrm{Tr}^L \otimes I^{R_2})(\mathcal{A}_{\mathsf{Leak}}^{BR_1} \otimes I^{R_2})(|x\rangle\langle x| \otimes \rho^{R_1R_2})$$

By Lemma 2,

$$\sigma^{XZR_2} = \sum_x \Pr[X=x] \cdot |x\rangle\langle x| \otimes \xi_x^Z \otimes \mathrm{Tr}_{B,R_1}(|x\rangle\langle x| \otimes \rho)$$
$$= \left(\sum_x \Pr[X=x] \cdot |x\rangle\langle x| \otimes \xi_x^Z\right) \otimes \mathrm{Tr}_{R_1}(\rho).$$

Hence, by Lemma 4, $\mathbf{H}_\infty(X|Z,R_2) = \mathbf{H}_\infty(X|Z)$. Combining with above,

$$\mathbf{H}_\infty(X|L,Z,R_2) \geq \mathbf{H}_\infty(X|Z) - \ell.$$

Finally, again by applying Lemma 3 to both entropies, we get

$$p_{guess}(X|L,Z,R_2) \leq 2^\ell \cdot p_{guess}(X|Z).$$

□

In fact, we can say more.

Lemma 15. *Let* $\mathsf{LeakyGame}_{\mathcal{A}}$ *be a* leakage game *as follows, with a (possibly quantum) secret* R_{sec} *and public information* pk*, played by* entangled *adversaries* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *such that* $\mathcal{A}_{\mathsf{Main}}$ *can locally test if it has won the game.*[12]

$\mathsf{LeakyGame}_{\mathcal{A}}$

1. *Initialize* $R_1, R_2 \leftarrow \rho$.
2. *Sample the secret* R_{sec} *and the public information* pk.
3. $\mathcal{A}_{\mathsf{Leak}}$ *gets* R_{sec}, pk, R_1*, produces a classical leakage* L.
4. $\mathcal{A}_{\mathsf{Main}}$ *gets* L, R_2 *and* pk*, produces an output.*
5. *Output* 1 *iff the output of* $\mathcal{A}_{\mathsf{Main}}$ *passes the game winning test.*

Also define the non-leaky version of the game as follows.

[12] E.g., a digital signature forgery game where $\mathcal{A}_{\mathsf{Main}}$ can check using the public verification key if it has succeeded in forging a valid signature.

$\underline{\mathsf{NonLeakyGame}_{\mathcal{B}}}$

1. *Sample the secret R_{sec} and the public information pk.*
2. *$\mathcal{B}$ gets pk, produces an output.*
3. *Output 1 iff the output of $\mathcal{B}$ passes the game winning test.*

Then, for any {efficient, unbounded} adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ where the output of $\mathcal{A}_{\mathsf{Leak}}$ is ℓ bits, there is an {efficient, unbounded} adversary $\mathcal{B}$ such that

$$\Pr[\mathsf{LeakyGame}_{\mathcal{A}} = 1] \leq 2^{\ell} \cdot \Pr[\mathsf{NonLeakyGame}_{\mathcal{B}} = 1].$$

Proof. Take any $\mathcal{A}$ where the output of $\mathcal{A}_{\mathsf{Leak}}$ is ℓ bits. Define $\mathcal{B}$ as follows.

$\underline{\mathcal{B}(pk)}$

1. Initialize $R_1, R_2 \leftarrow \rho$.
2. Sample $L \leftarrow \{0,1\}^{\ell}$.
3. Output $\mathcal{A}_{\mathsf{Main}}(R_2, pk, L)$.

Define the first hybrid Hyb_0 to be $\mathsf{LeakyGame}_{\mathcal{A}}$. Then, define the second hybrid Hyb_1 by modifying Hyb_0 as follows. Instead of running $\mathcal{A}_{\mathsf{Main}}$ on the leakage L, we sample an independent string $L' \leftarrow \{0,1\}^{\ell}$ and run $\mathcal{A}_{\mathsf{Main}}$ using L'. Since we have $L = L'$ with probability $\frac{1}{2^{\ell}}$, we get $\Pr[\mathsf{Hyb}_1 = 1] \geq 2^{-\ell} \Pr[\mathsf{Hyb}_0 = 1]$. Finally, define Hyb_2 by modifying Hyb_1 so that we do not run $\mathcal{A}_{\mathsf{Leak}}$ at all. Note that in Hyb_1 and Hyb_2, (since $\mathcal{A}_{\mathsf{Main}}$ ignores L) $\mathcal{A}_{\mathsf{Leak}}$ and $\mathcal{A}_{\mathsf{Main}}$ are not communicating. By no-signalling property of entanglement, $\mathcal{A}_{\mathsf{Main}}$ cannot *detect* (without communication) if $\mathcal{A}_{\mathsf{Leak}}$ has acted on its part of the entanglement or not. Since the game is testable on the side of $\mathcal{A}_{\mathsf{Main}}$ alone, we get $\Pr[\mathsf{Hyb}_1 = 1] = \Pr[\mathsf{Hyb}_2 = 1]$. It is easy to see that Hyb_2 is the same as $\mathsf{NonLeakyGame}_{\mathcal{B}}$, which concludes the proof. □

5.1.1 Pseudorandom Functions

Definition 50 (Spooky leakage-resilient weak PRF)**.** *We say that a family of functions $\mathcal{F} = \{f_k\}_k$ is an $\ell(\lambda)$-spooky-classic-leakage-resilient (spooky-classic-LR) weak PRF if, for all polynomials $p(\cdot)$, states $\rho \in \mathcal{H}^{\mathrm{poly}(\lambda)}$ and QPT adversaries $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ where the output of $\mathcal{A}_{\mathsf{Leak}}$ consists of $\ell(\lambda)$ classical bits, advantage of $\mathcal{A}$ is negligible in the following game.*

1. *Challenger samples a key $k \leftarrow \mathcal{K}$.*
2. *Challenger samples random query inputs $x_1, \ldots, x_{p(\lambda)} \leftarrow \mathcal{X}$.*
3. *For $i \in [p(\lambda)]$, compute $y_i = f_k(x_i)$.*
4. *Initialize $R_1, R_2 \leftarrow \rho$.*
5. *The leakage adversary get access to the query outputs and the key, outputs an ℓ bit classical leakage*
$$L \leftarrow \mathcal{A}_{\mathsf{Leak}}(R_1, k, (x_i, y_i)_{i \in [p(\lambda)]}).$$
6. *The challenger samples a challenge input $x^* \leftarrow \mathcal{X}$ and a challenge bit $b \leftarrow \{0,1\}$. If $b = 0$, it sets $y^* \leftarrow f_k(x^*)$ and if $b = 1$, it samples $y^* \leftarrow \mathcal{Y}$.*

7. *The adversary gets the leakage, the query outputs and the challenge input, outputs a guess*

$$b' \leftarrow \mathcal{A}_{\mathsf{Main}}(R_2, L, (x_i, y_i)_{i\in[p(\lambda)]}, x^*, y^*).$$

8. *Output* 1 *iff* $b' = b$.

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

Theorem 36. *Assuming the existence of (post-quantum) one-way functions, for any polynomial* $\ell(\cdot)$*, there exists* $\ell(\lambda)$*-spooky-classic-LR wPRFs.*

Proof. Let $\mathcal{F}' = \{f'_k\}$ be a symmetric-key weak hash-proof system (Definition 20) with input size $p_1(\lambda)$ and output size $p_2(\lambda)$. See [HLAWW16, Section 4.2] for constructions of such schemes from any wPRF. It can be verified that the construction is post-quantum secure when the underlying wPRF is. Let Ext be the extractor from Theorem 13 instantiated with $k = p_2(\lambda) - \ell(\lambda)$ and some $\varepsilon = \mathsf{negl}(\lambda)$. Let $p_3(\lambda), p_4(\lambda)$ denote the seed length and the output size of Ext respectively. Then, we claim that the following construction from [HLAWW16, Theorem 4.3] is $\ell(\lambda)$-spooky-classic-LR.

- Key distribution: Same as $\mathcal{F}'$
- Input space: $\{0,1\}^{p_1(\lambda)} \times \{0,1\}^{p_3(\lambda)}$
- Output space: $\{0,1\}^{p_4(\lambda)}$
- Evaluation:

$$f_k(x||s) = \mathsf{Ext}(f'_k(x), s)$$

Proof is mostly the same as that of [HLAWW16, Theorem 4.3], but we repeat an abridged version for convenience. The significant difference is for showing $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$, which now relies on the quantum-proof properties of Ext (Definition 13) and the spooky classical leakage lemma (Lemma 14).

Define the following hybrids. In all of the hybrids, we will consider sampling a uniform (x, s) as sampling x and s independently, which is equivalent.

Hyb_0 The original leakage resilience game with the challenge bit b fixed to 0.

Hyb_1 Modify Hyb_0 as follows. Let the challenger also choose a sampling key $samK \leftarrow \mathsf{SamGen}(k)$. Change the sampling of the (left parts of) query inputs from

$$x_1, \ldots, x_{p(\lambda)} \leftarrow \{0,1\}^{p_1(\lambda)}$$

to

$$x_1, \ldots, x_{p(\lambda)} \leftarrow \mathsf{Dist}_1(samK).$$

Also change the sampling of the (left part of) challenge input from $x^* \leftarrow \{0,1\}^{p_1(\lambda)}$ to

$$x^* \leftarrow \mathsf{Dist}_2(samK).$$

By the input indistinguishability of $\mathcal{F}'$, we have that for any polynomial number of samples, adversaries cannot distinguish between sampling from

$$\mathsf{Dist}_1(samK), \mathsf{Dist}_2(samK)$$

or $\{0,1\}^{p_1(\lambda)}$, even given the key k in the plain. Hence, the same indistinguishability holds given the (spooky) leakage from the key, therefore we have $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$.

$\underline{\mathsf{Hyb}_2}$ Modify Hyb_1 by changing the sampling of the challenge from

$$y^* \leftarrow f_k(x^*)$$

to

$$y^* \leftarrow \{0,1\}^{p_4(\lambda)}.$$

We argue $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$ as follows. Let P denote the query input-output pairs $((x_i, s_i), \mathsf{Ext}(f'_k(x_i), s_i))_{i\in[p(\lambda)]}$ and L denote the ℓ-bit classical leakage string that $\mathcal{A}_{\mathsf{Leak}}$ outputs. By smoothness of $\mathcal{F}'$, we can replace $f'_k(x^*)$ with $U_{p_2(\lambda)}$ even given P (but not given the leakage). Therefore, we have $\mathbf{H}_\infty(f'_k(x^*)|P) = p_2(\lambda)$. Then, by Lemma 14, we have $\mathbf{H}_\infty(f'_k(x^*)|R_2, P, L) = p_2(\lambda) - \ell$. Finally, by Definition 13 and Theorem 13, we can replace $\mathsf{Ext}(f'_k(x^*), s^*)$ with $U_{p_4(\lambda)}$ even given R_2, P and L, which implies $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$.

$\underline{\mathsf{Hyb}_3}$ Modify Hyb_2 by undoing the changes we made from Hyb_0 to Hyb_1. By the same argument that shows $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_1$, we have $\mathsf{Hyb}_2 \approx \mathsf{Hyb}_3$. Also observe that Hyb_3 is the same as the original leakage-resilience game with the challenge bit b fixed to 1.

Finally, we get $\mathsf{Hyb}_0 \approx \mathsf{Hyb}_3$, which proves that $\mathcal{F}$ is $\ell(\lambda)$-spooky-classic-LR. □

5.1.2 Public-key encryption

Definition 51 (Spooky leakage-resilient public-key encryption)**.** *We say that a public-key encryption scheme* PKE *is an* $\ell(\lambda)$*-spooky-classic-leakage-resilient PKE if, for all states* $\rho \in \mathcal{H}^{\mathrm{poly}(\lambda)}$ *and pairs of QPT adversaries* $\mathcal{A} = (\mathcal{A}_{\mathsf{Main}}, \mathcal{A}_{\mathsf{Leak}})$ *where the output of* $\mathcal{A}_{\mathsf{Leak}}$ *consists of* $\ell(\lambda)$ *classical bits, advantage of* $\mathcal{A}$ *is negligible in the following game.*

1. *The challenger samples a public key - secret key pair*

$$pk, sk \leftarrow \mathsf{PKE.KeyGen}(1^\lambda).$$

2. *Initialize* $R_1, R_2 \leftarrow \rho$.

3. *The leakage adversary gets access to the keys, and produces an* ℓ*-bit classical leakage string* L *as*

$$L \leftarrow \mathcal{A}_{\mathsf{Leak}}(R_1, pk, sk).$$

4. *The adversary gets the leakage, and outputs two messages and a state* R

$$m_0, m_1, R \leftarrow \mathcal{A}_{\mathsf{Main}}(R_2, L).$$

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

6. *The adversary gets the challenge ciphertext and outputs a guess*

$$b' \leftarrow \mathcal{A}_{\mathsf{Main}}(R], L, ct).$$

7. *Output* 1 *iff* $b' = b$.

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

Theorem 37. *Assuming the existence of (post-quantum) public-key encryption schemes, for any polynomial $\ell(\cdot)$, there exists a $\ell(\lambda)$-spooky-classic-LR public-key encryption scheme.*

Proof. Let wHPS (Definition 21) be a public-key weak hash-proof system with output space $\{0,1\}^{p_1(\lambda)}$. See [HLAWW16, Section 3.2] for constructions of such schemes from any PKE. It can verified that the construction is post-quantum secure when the underlying PKE is. Let Ext be the extractor from Theorem 13 instantiated with $k = p_1(\lambda) - \ell(\lambda)$ and some $\varepsilon = \mathsf{negl}(\lambda)$. Let $p_2(\lambda), p_3(\lambda)$ denote the seed length and the output length of Ext, respectively. Then, we claim that the following construction from [HLAWW16, Theorem 3.3] is $\ell(\lambda)$-spooky-classic-LR.

PKE

- $\mathsf{PKE.KeyGen}(1^\lambda)$
  1. Output $\mathsf{wHPS.KeyGen}(1^\lambda)$.
- $\mathsf{PKE.Enc}(pk, m)$
  1. $s \leftarrow \{0,1\}^{p_2(\lambda)}$.
  2. $c_0, k \leftarrow \mathsf{wHPS.Encap}(pk)$.
  3. $c_1 = m \oplus \mathsf{Ext}(k, s)$.
  4. Output (c_0, c_1, s).
- $\mathsf{PKE.Dec}(sk, (c_0, c_1, s))$
  1. $k \leftarrow \mathsf{wHPS.Decap}(sk, c_0)$.
  2. Output $c_1 \oplus \mathsf{Ext}(k, s)$

Proof is mostly the same as that of [HLAWW16, Theorem 3.2], but we repeat an abridged version for convenience. The significant difference is for showing $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$, which now relies on the quantum-proof properties of Ext (Definition 13) and the spooky classical leakage lemma (Lemma 14).

Hyb_0 The original leakage resilience game (Definition 51).

Hyb_1 Modify Hyb_0 as follows. For the computation of the challenge ciphertext, change

$$c_1 = m_b \oplus \mathsf{Ext}(k, s)$$

to

$$k' \leftarrow \mathsf{wHPS.Decap}(\mathsf{sk}, \mathsf{c}_0)$$
$$c_1 = m_b \oplus \mathsf{Ext}(k', s).$$

By the correctness property of wHPS, we have $k = k'$ with probability 1 and hence $\mathsf{Hyb}_0 \equiv \mathsf{Hyb}_1$.

$\underline{\mathsf{Hyb}_2}$ Modify Hyb_1 by, for the computation of the challenge ciphertext, changing

$$c_0, k \leftarrow \mathsf{wHPS.Encap}(pk) \tag{13}$$

to

$$c_0 \leftarrow \mathsf{wHPS.Encap}^*(pk) \tag{14}$$

By the ciphertext indistinguishability property of wHPS, c_0 sampled as (13) and as (14) are indistinguishable, even given the public key and the private key in the plain. Therefore, they are indistinguishable given the (spooky) leakage, hence $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$.

$\underline{\mathsf{Hyb}_2}$ Modify Hyb_2 by, for the computation of the challenge ciphertext, changing

$$c_1 = m_b \oplus \mathsf{Ext}(k', s)$$

to

$$r \leftarrow \{0,1\}^{p_3(\lambda)}$$
$$c_1 = m_b \oplus r.$$

Observe the side information we have on k' consists only of R_2, pk, c_0 and the leakage L. By the smoothness property of wHPS, we have that k' is (exactly) uniform even given c_0 and pk. Therefore, we have $\mathbf{H}_\infty(k'|pk, c_0) = p_1(\lambda)$ and hence by Lemma 14 we get $\mathbf{H}_\infty(k'|R_2, L, pk, c_0) \geq p_1(\lambda) - \ell(\lambda)$. Finally, by Definition 13 and Theorem 13, we can replace $\mathsf{Ext}(k', s)$ with r even given R_2, L, pk, c_0, which implies $\mathsf{Hyb}_1 \approx \mathsf{Hyb}_2$.

Observe that in Hyb_2, the message is encrypted with a one-time pad key that is independent of everything else, which proves the security of PKE. □

5.1.3 Digital Signatures

In this section we show classical digital signature schemes resilient to spooky-classical leakage. The main adversary $\mathcal{A}_{\mathsf{Main}}$ attempts to succeed in the usual EUF-CMA forgery game with access to a signing oracle. But in addition there is a leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ who has knowledge of the secret key, and the main adversary is allowed to query once this adversary and obtain a bounded amount of classical leakage. Both adversaries share arbitrary entanglement. With this model in mind, we show constructions for one-time and t-time signature schemes tolerating different amounts of classical leakage.

Definition 52. *Let λ be a security parameter. A signature scheme is a tuple of PPT algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, where:*

- Gen *is a randomized algorithm that takes as input 1^λ and outputs a public key pk and a secret key sk.*
- Sign *is a possibly randomized algorithm that takes as input the secret key sk, a message m and output a signature $\sigma = \mathsf{Sign}_{\mathsf{sk}}(m)$.*
- Vrfy *is a deterministic algorithm that takes as input the public key pk, a message and a signature σ. It outputs a bit $b = \mathsf{Vrfy}_{\mathsf{pk}}(m, \sigma)$.*

We require that for all λ, $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\lambda)$, we have that

$$\mathsf{Vrfy}_{\mathsf{pk}}(m, \mathsf{Sign}_{\mathsf{sk}}(m)) = 1.$$

Security is defined via the usual notion of existential unforgeability under chosen-message attacks, except that the adversary attempting to win the game is allowed to obtain a bounded string of classical leakage from a separate local adversary who receives the secret key. Both adversaries share arbitrary quantum entanglement.

Definition 53 (Spooky leakage-resilient digital signature)**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *be a signature scheme. Given an adversary* $\mathcal{A} = (\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$*, we define the following experiment, with security parameter* λ*.*

- *Let* $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\lambda)$*.*
- *The local adversary* $\mathcal{A}_{\mathsf{Leak}}$ *has as input* 1^λ*, the secret key* sk *and a quantum register* R_1*. Upon being queried, it outputs classical leakage* $\rho = \mathcal{A}_{\mathsf{Leak}}(1^\lambda, \mathsf{sk}, R_1)$*.*
- *The second adversary* $\mathcal{A}_{\mathsf{Main}}$ *has as input* 1^λ*, the public key* pk *and a quantum register* R_2*, and has access to a signing oracle* $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ *and can query once to* $\mathcal{A}_{\mathsf{Leak}}$*. It outputs a pair* $(m^*, \sigma^*) = \mathcal{A}_{\mathsf{Main}}(1^\lambda, \mathsf{pk}, \rho, R_2)$*.*

We say that $\mathcal{A}$ *succeeds if* $\mathsf{Vrfy}_{\mathsf{pk}}(m^*, \sigma^*) = 1$ *and* m^* *was not queried to the signing oracle* $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ *by* $\mathcal{A}_{\mathsf{Main}}$*. Let* $L \geq |\rho|$ *be a bound on the classical leakage from* $\mathcal{A}_{\mathsf{Leak}}$*,* q *be a bound on the number of queries to the signing oracle* $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ *by* $\mathcal{A}_{\mathsf{Main}}$*, and* R_1 *and* R_2 *be a pair of quantum registers composing an arbitrary (possibly entangled) quantum state. We denote the probability that the adversary* $\mathcal{A}$ *wins by* $\Pr\left[\mathsf{Succ}^{L,q}_{\mathcal{A},\Pi}(\lambda)\right]$*.*

We say that Π *is a* q*-time* L*-spooky-classic-leakage-resilient signature if the quantity* $\Pr\left[\mathsf{Succ}^{L,q}_{\mathcal{A},\Pi}(\lambda)\right]$ *is negligible in* λ *for every PPT adversary* $\mathcal{A}$*.*

One-time signature scheme. The following scheme was originally proposed by Katz and Vaikuntanathan [KV09], but its security was only proven against classical adversaries. The scheme is essentially a variant of Lamport's signature scheme [Lam79], where the message is encoded using a linear error correcting code.

Consider a universal one-way hash function (Gen_H, H), mapping ℓ_{in}-bit inputs to λ-bit outputs. Let $\ell = R\lambda$ and $\ell_{in} = 2\lambda/\varepsilon$, where R is chosen such that a random binary matrix $A \in \{0,1\}^{\lambda \times \ell}$ defines a code with relative minimum distance $1/2 - \varepsilon$, except with probability negligible in λ.

Key generation: Choose a random binary matrix $A \in \{0,1\}^{\lambda \times \ell}$, and $x_{i,0}, x_{i,1} \in \{0,1\}^{\ell_{in}}$ random strings. Compute $s = \mathsf{Gen}_H(1^\lambda)$. Compute $y_{i,b} = H_s(x_i, b)$ for $i \in \{1, \ldots, \ell\}$ and $b \in \{0,1\}$. Let $\mathsf{pk} = (A, s, \{y_{i,b}\})$ and $\mathsf{sk} = \{x_{i,b}\}$.

Signing: To sign a message $m \in \{0,1\}^\lambda$, compute $\overline{m} = m \cdot A \in \{0,1\}^\ell$. The signature is $x_{1,\overline{m}_1}, \ldots, x_{\ell,\overline{m}_\ell}$.

Verification: Given a signature $x_1, \ldots, x_\ell$ on message m and public key $\mathsf{pk} = (A, s, \{y_{i,b}\})$, compute $\overline{m} = m \cdot A$ and output 1 if and only if $y_{i,\overline{m}_i} = H_s(x_i)$ for all i.

Theorem 38. *If* (Gen_H, H) *is a post-quantum secure universal one-way hash function, then the scheme above is a one-time* $(1/4 - \varepsilon)n$ *spooky-classical-leakage-resilient signature scheme, where* $n = 2\ell \cdot \ell_{in}$ *is the size of the secret key.*

Proof. Let δ be the success probability of an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$ in attacking the scheme Π. We construct an adversary $\mathcal{B}$ that succeeds with probability $(\delta - \mathsf{negl}(\lambda))/(4\ell)$ in breaking the security of the universal one-way hash function (Gen_H, H).

The adversary $\mathcal{B}$ locally randomly generates the matrix $A \in \{0,1\}^{\lambda \times \ell}$, and $x_{i,0}, x_{i,1} \in \{0,1\}^{\ell_{in}}$.

Then, $\mathcal{B}$ chooses a random $b^* \in \{0,1\}$ and $i^* \in \{1, \ldots, \ell\}$ and outputs x_{i^*,b^*} as the initial input for the hash function experiment. In turn, $\mathcal{B}$ receives $s = \mathsf{Gen}_H(1^\lambda)$. (The goal for $\mathcal{B}$ to win is to compute an output y such that $H_s(x_{i^*,b^*}) = H_s(y)$.)

After that, $\mathcal{B}$ computes $y_{i,b} = H_s(x_{i,b})$ and sets $\mathsf{pk} = (A, s, \{y_{i,b}\})$ and $\mathsf{sk} = \{x_{i,b}\}$. The secret key sk is given to $\mathcal{A}_{\mathsf{Leak}}$, who computes the leakage ρ, with $|\rho| \leq (1/4 - \varepsilon)n$.

The public key pk and ρ are given to $\mathcal{A}_{\mathsf{Main}}$. $\mathcal{B}$ answers the signing query from $\mathcal{A}_{\mathsf{Main}}$ using the secret key sk. Given that the simulation for $\mathcal{A}$ is perfect (since $\mathcal{B}$ generates the secret key identically as in the signature forgery game), we have that $\mathcal{A}_{\mathsf{Main}}$ outputs a forgery (m^*, σ^*) with probability δ.

Let $\overline{m} = m \cdot A$, where m is the message whose signature was requested by $\mathcal{A}_{\mathsf{Main}}$. The view of $\mathcal{A}_{\mathsf{Main}}$ about the secret key sk contains the signature $(x_{1,\overline{m}_1}, \ldots, x_{\ell,\overline{m}_\ell})$, the values $\{y_{i,1-\overline{m}_i}\}_{i=1}^{\ell}$ from the public key and the leakage ρ, as well as the quantum register R_2.

Parse the forgery output of $\mathcal{A}_{\mathsf{Main}}$ as $(m^*, \sigma^*) = (m^*, (x_1^*, \ldots, x_\ell^*))$, and let $\overline{m}^* = m^* \cdot A$. Let I be the set of indices where $\overline{m}$ and $\overline{m}^*$ differ. Note that it must hold that $\overline{m}$ and $\overline{m}^*$ differ in at least $|I| \geq (1/2 - \varepsilon) \cdot \ell$ indices (due to the fact that $m \neq m^*$ and A defines a code with relative minimum distance $1/2 - \varepsilon$ with overwhelming probability).

Let E be the event that the view of $\mathcal{A}_{\mathsf{Main}}$ fixes all values $\{x_{i,1-\overline{m}_i}\}$. We will show that the probability that the event E occurs is bounded by $2^{L-\Delta} = \mathsf{negl}(\lambda)$, where $L = \ell \cdot \ell_{in} + \ell\lambda + (1/4 - \varepsilon)2\ell \cdot \ell_{in}$ and $\Delta = \mathbf{H}_\infty(\mathsf{sk}) - (1/2 + \varepsilon) \cdot \ell \cdot \ell_{in}$.

To see this, consider a hybrid scenario where the leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ computes ρ from an independent random string instead of sk. By Lemma 14, it is enough to prove that in this case the probability that the event occurs in this hybrid scenario is bounded by $2^{L'-\Delta}$, where $L' = L - (1/4 - \varepsilon)2\ell \cdot \ell_{in}$.

In this hybrid scenario, the adversary receives in total L' bits related to the secret key (here the leakage from $\mathcal{A}_{\mathsf{Leak}}$ is independent of the secret key), and the secret key has length $2\ell \cdot \ell_{in}$. Using standard entropy arguments, we know that the probability that the L' bits decrease the entropy of the secret key by more than $\Delta = \mathbf{H}_\infty(\mathsf{sk}) - (1/2+\varepsilon) \cdot \ell \cdot \ell_{in}$ (meaning there is less than $(1/2+\varepsilon) \cdot \ell \cdot \ell_{in}$ entropy left), is at most $2^{L'-\Delta}$.

Now, observe that whenever the entropy left on the secret key is at least $(1/2+\varepsilon) \cdot \ell \cdot \ell_{in}$, then the event E does not hold. To see this, assume the contrary (that the adversary $\mathcal{A}_{\mathsf{Main}}$ fixes all values for indices in I). This would mean that $\mathbf{H}_\infty(\mathsf{sk} \mid \mathcal{A}_{\mathsf{Main}}\text{'s view}) \leq \sum_{i \notin I} \mathbf{H}_\infty(x_{i,1-\overline{m}_i} \mid \mathcal{A}_{\mathsf{Main}}\text{'s view}) \leq (1/2 + \varepsilon)\ell \cdot \ell_{in}$, in contradiction to the assumption on the conditional entropy of sk.

Then, by the argument above, we have that $x_i^* \neq x_{i,\overline{m}_i^*}$ for at least an index i^* and bit b^*. Therefore, with probability at least $1/(2\ell)$ this difference occurs at (i^*, b^*), the initial guess by $\mathcal{B}$, and in this case $\mathcal{B}$ finds a valid collision. Putting all together, $\mathcal{B}$ finds a valid collision with probability at least $(\delta - \Pr[E] - \mathsf{negl}(\lambda))/(4\ell)$. Since $\Pr[E] = \mathsf{negl}(\lambda)$, the result follows. □

Extension to t-time signature scheme with $\theta(n/t^2)$ leakage. The previous scheme can be modified to achieve a t-time L-spooky-classical-leakage signature scheme with $L = \theta(n/t^2)$, n being the size of the secret key, as also shown in [KV09], using so-called *cover-free families*.

Definition 54. *A family of non-empty sets $\mathcal{S} = \{S_1, \ldots, S_n\}$, where $S_i \subset U$ is a $(t, 1/2)$-cover free family if for all $S, S_1, \ldots, S_t \in \mathcal{S}$ it holds that $|S \setminus \cup_{i=1}^t S_i| \geq |S|/2$.*

Kumar, Rajagopalan, and Sahai [KRS99] construct a $(t, 1/2)$-cover-free family with $n = \Omega(2^\lambda)$ sets, and where each set has size $|S_i| = O(\lambda t)$ from a universe set of size $|U| = O(\lambda t^3)$. Let $f : \{0,1\}^\lambda \to \mathcal{S}$ be an injective function, we define the following scheme:

Key generation: Let $\ell = O(\lambda t^3)$ and $\ell_{in} = 8t^2\lambda$. Choose $x_i \in \{0,1\}^{\ell_{in}}$ for $i = 1, \ldots, \ell$. Compute $s = \mathsf{Gen}_H(1^\lambda)$ and $y_i = H_s(x_i)$ for $i \in \{1, \ldots, \ell\}$. Let $\mathsf{pk} = (s, \{y_i\})$ and $\mathsf{sk} = \{x_i\}$.

Signing: To sign a message $m \in \{0,1\}^\lambda$, compute $f(m) = S_m \in \mathcal{S}$. The signature is $\{x_i\}_{i \in S_m}$.

Verification: Given a signature $\{x_i\}$ on message m, with respect to public key $\mathsf{pk} = (s, \{y_i\})$, compute $S_m = f(m)$ and output 1 if and only if $y_i = H_s(x_i)$ for all $i \in S_m$.

The proof of the following theorem is similar to the proof of Theorem 38.

Theorem 39. *If (Gen_H, H) is a post-quantum secure universal one-way hash function, then the described scheme is a t-time $\theta(n/t^2)$ spooky-classical-leakage-resilient scheme, where $n = \ell \cdot \ell_{in}$ is the size of the secret key.*

Proof. Let δ be the success probability of an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$ in attacking the scheme Π. We construct an adversary $\mathcal{B}$ that succeeds with probability $(\delta - \mathsf{negl}(\lambda))/(2\ell)$ in breaking the security of the universal one-way hash function (Gen_H, H).

The adversary $\mathcal{B}$ locally randomly generates $x_i \in \{0,1\}^{\ell_{in}}$. Then, $\mathcal{B}$ chooses a random $i^* \in \{1, \ldots, \ell\}$ and outputs x_i as the initial input for the hash function experiment. In turn, $\mathcal{B}$ receives $s = \mathsf{Gen}_H(1^\lambda)$. (The goal for $\mathcal{B}$ to win is to compute an output y such that $H_s(x_{i^*}) = H_s(y)$.)

After that, $\mathcal{B}$ computes $y_i = H_s(x_i)$ and sets $\mathsf{pk} = (s, \{y_i\})$ and $\mathsf{sk} = \{x_i\}$. The secret key sk is given to $\mathcal{A}_{\mathsf{Leak}}$. Then, whenever $\mathcal{A}_{\mathsf{Leak}}$ is queried, he computes the leakage ρ, with $|\rho| = \theta(n/t^2)$.

The public key pk is given to $\mathcal{A}_{\mathsf{Main}}$. When $\mathcal{A}_{\mathsf{Main}}$ queries $\mathcal{A}_{\mathsf{Leak}}$, he gets ρ. Moreover, $\mathcal{B}$ answers the signing query from $\mathcal{A}_{\mathsf{Main}}$ using the secret key sk. Given that the simulation for $\mathcal{A}$ is perfect (since $\mathcal{B}$ generates the secret key identically as in the signature forgery game), we have that $\mathcal{A}_{\mathsf{Main}}$ outputs a forgery (m^*, σ^*) with probability δ.

Let $\overline{m}_1, \ldots, \overline{m}_t$, where $\overline{m}_i = f(m_i)$ and m_i is the i-th message whose signature was requested by $\mathcal{A}_{\mathsf{Main}}$. Let $T = \cup_{j=1}^t S_{m_j}$. The view of $\mathcal{A}_{\mathsf{Main}}$ about the secret key sk contains the values $\{x_i\}_{i \in T}$, the public-key values $\{y_i\}_{i \notin T}$ and the leakage ρ, as well as the quantum register R_2.

Now parse the forgery output of $\mathcal{A}_{\mathsf{Main}}$ as $(m^*, \sigma^*) = (m^*, \{x_i\}_{i \in S_{m^*}})$. Let I be the set of indices where T and S_{m^*} differ. Because $\mathcal{S}$ is a $(t, 1/2)$-cover-free family, we have that these sets differ in at least $|I| \geq |S|/2 = O(\lambda t)$ indices.

Let E be the event that the view of $\mathcal{A}_{\mathsf{Main}}$ fixes all values $\{x_i\}_{i \in S_{m^*} \setminus T}$. We will show that the probability that the event E occurs is bounded by $2^{L-\Delta} = \mathsf{negl}(\lambda)$, where $L = \lambda t^2 + \ell\lambda + \theta(n/t^2)$ and $\Delta = \mathbf{H}_\infty(\mathsf{sk}) - (\ell - O(\lambda t))\ell_{in}$.

To see this, consider a hybrid scenario where the leakage adversary $\mathcal{A}_{\mathsf{Leak}}$ computes ρ from an independent random string instead of sk. By Lemma 14, it is enough to prove that in this case the probability that the event occurs in this hybrid scenario is bounded by $2^{L'-\Delta}$, where $L' = L - \theta(n/t^2)$.

In this hybrid scenario, the adversary receives in total L' bits related to the secret key (here the leakage from $\mathcal{A}_{\mathsf{Leak}}$ is independent of the secret key), and the secret key has length $\ell \cdot \ell_{in}$.

Using standard entropy arguments, we know that the probability that the L' bits decrease the entropy of the secret key by more than $\Delta = \mathbf{H}_\infty(\mathsf{sk}) - (\ell - O(\lambda t))\ell_{in}$ (meaning there is less than $(\ell - O(\lambda t))\ell_{in}$ entropy left), is at most $2^{L'-\Delta}$.

Now, observe that whenever the entropy left on the secret key is at least $(\ell - O(\lambda t))\ell_{in}$, then the event E does not hold. To see this, assume the contrary (that the adversary $\mathcal{A}_{\mathsf{Main}}$ fixes all values for indices in I). This would mean that $\mathbf{H}_\infty(\mathsf{sk} \mid \mathcal{A}_{\mathsf{Main}}\text{'s view}) \leq \sum_{i \notin I} \mathbf{H}_\infty(x_i \mid \mathcal{A}_{\mathsf{Main}}\text{'s view}) \leq |U \setminus I| \cdot \ell_{in} = (\ell - O(\lambda t))\ell_{in}$, in contradiction to the assumption on the conditional entropy of sk.

Then, by the argument above, we have that $x_i^* \neq x_j$ for $j \in T$ for at least an index i^*. Therefore, with probability at least $1/\ell$ this difference occurs at i^*, the initial guess by $\mathcal{B}$, and in this case $\mathcal{B}$ finds a valid collision. Putting all together, $\mathcal{B}$ finds a valid collision with probability at least $(\delta - \Pr[E] - \mathsf{negl}(\lambda))/(2\ell)$. Since $\Pr[E] = \mathsf{negl}(\lambda)$, the result follows. □

Extension to t-time stateful signature scheme with leakage independent of t. The above scheme tolerates a leakage that decreases with the number of times the signature can be used. In this section, we show how to improve the scheme from above to a many-times *stateful* signature scheme, with leakage that is linear in the size of the secret, by providing a leakage-preserving compiler that transforms any 3-time spooky leakage signature scheme into a many-times spooky leakage signature scheme. By instantiating the 3-time spooky leakage resilient scheme with $\theta(n/9)$ leakage from above, we obtain a t-time spooky leakage resilient scheme with the same leakage tolerance.

The compiler was originally introduced by Faust, Kiltz, Pietrzak, and Rothblum [FKPR10] in the context of classical leakage-resilience. We show that the scheme also achieves $\theta(n/9)$ spooky leakage resilience.

We first define a stateful signature scheme. The difference with respect to a stateless scheme is that the signing algorithm outputs in addition an updated secret key sk', which replaces the previous secret key.

Definition 55. *Let λ be a security parameter. A stateful signature scheme is a tuple of PPT algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, where:*

- Gen *is a randomized algorithm that takes as input 1^λ and outputs a public key pk and a secret key sk.*
- Sign *is a possibly randomized algorithm that takes as input the secret key sk, a message m and outputs a signature and the updated secret key $(\sigma, \mathsf{sk}') = \mathsf{Sign}_{\mathsf{sk}}(m)$.*
- Vrfy *is a deterministic algorithm that takes as input the public key pk, a message and a signature σ. It outputs a bit $b = \mathsf{Vrfy}_{\mathsf{pk}}(m, \sigma)$.*

We require the usual correctness properties, that for all λ, any pair of keys $(\mathsf{pk}, \mathsf{sk}')$ initially generated by the Gen algorithm, but where sk' might have been updated according to the signing algorithm, we have that $\mathsf{Vrfy}_{\mathsf{pk}}(m, \mathsf{Sign}_{\mathsf{sk}'}(m)) = 1$.

Security is defined similarly as with stateless signatures, except that the adversary $\mathcal{A}_{\mathsf{Leak}}$ computes the leakage according to the current updated secret key sk'.

Definition 56 (Spooky leakage-resilient stateful digital signature)**.** *Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a stateful signature scheme. Given an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$, we define the following experiment, with security parameter λ.*

- *Let $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\lambda)$.*
- *The local adversary $\mathcal{A}_{\mathsf{Leak}}$ has as input 1^λ, the current secret key sk' and a quantum register R_1. Upon being queried, it outputs classical leakage $\rho = \mathcal{A}_{\mathsf{Leak}}(1^\lambda, \mathsf{sk}', R_1)$.*

- *The second adversary* $\mathcal{A}_{\mathsf{Main}}$ *has as input* 1^λ*, the public key* pk *and a quantum register* R_2*, and has access to a signing oracle* $\mathsf{Sign}_{\mathsf{sk}'}(\cdot)$ *(which updates the secret key every time it sends a message) and can query once to* $\mathcal{A}_{\mathsf{Leak}}$*. It outputs a pair* $(m^*, \sigma^*) = \mathcal{A}_{\mathsf{Main}}(1^\lambda, \mathsf{pk}, \rho, R_2)$*.*

We say that $\mathcal{A}$ *succeeds if* $\mathsf{Vrfy}_{\mathsf{pk}}(m^*, \sigma^*) = 1$ *and* m^* *was not queried to the signing oracle by* $\mathcal{A}_{\mathsf{Main}}$*. Let* $L \geq |\rho|$ *be a bound on the classical leakage from* $\mathcal{A}_{\mathsf{Leak}}$*,* q *be a bound on the number of queries to the signing oracle by* $\mathcal{A}_{\mathsf{Main}}$*, and* R_1 *and* R_2 *be a pair of quantum registers composing an arbitrary (possibly entangled) quantum state. We denote the probability that the adversary* $\mathcal{A}$ *wins by* $\Pr\left[\mathsf{Succ}^{L,q}_{\mathcal{A},\Pi}(\lambda)\right]$*.*

We say that Π *is a* q*-time* L*-spooky-classical-leakage-resilient stateful signature if* $\Pr\left[\mathsf{Succ}^{L,q}_{\mathcal{A},\Pi}(\lambda)\right]$ *is negligible in* λ *for every PPT adversary* $\mathcal{A}$*.*

Leakage-preserving compiler from 3*-time to* t*-times stateful signature scheme.* Given a leakage-resilient 3-time signature scheme Π, we recall the tree-based leakage-resilient signature scheme $\mathsf{Comp}(\Pi)$ presented in [FKPR10]. Given any fixed d, the construction can sign up to $t = 2^{d+1} - 2$ messages.

At a high level, the stateful signing algorithm traverses the $2^{d+1} - 1$ nodes of a binary tree of depth d in a depth-first order. Suppose the algorithm is signing the i-th message m, and the internal state points to the i-th node w (according to the depth-first order). The algorithm first computes a fresh pair $(\mathsf{pk}_w, \mathsf{sk}_w)$ for this node. Then, the signature consists of a pair (σ, Γ), where σ is a signature on m according to the 3-time signature scheme, and Γ contains a signature path from the root to the node w (where for each node v on the path, its corresponding public key pk_v is signed under the secret key associated to its parent node $\mathsf{sk}_{\mathsf{par}(v)}$). The public key of $\mathsf{Comp}(\Pi)$ is the public key associated to the root node, and verification is done by verifying that all 3-time signatures on the path are correct.

Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a leakage-resilient 3-time signature scheme. We describe the compiled stateful signature scheme $\mathsf{Comp}(\Pi) = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy}')$.

Key generation: Compute $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\lambda)$. Let $S = \{\mathsf{sk}\}$, $\Gamma = \emptyset$ and w_0 be the root node. Further let $\mathsf{sk}' = (w_0, S, \Gamma)$ and $\mathsf{pk}' = \mathsf{pk}$. Return the pair of keys $(\mathsf{pk}', \mathsf{sk}')$.

Signing: Let $m \in \{0,1\}^\lambda$ be the message, and $\mathsf{sk}' = (w, S, \Gamma)$ be the current secret key. The signing algorithm is performed as follows.

- Compute the next node w' (after w) in depth-first order.
- Generate a new key pair $(\mathsf{pk}_{w'}, \mathsf{sk}_{w'}) = \mathsf{Gen}(1^\lambda)$.
- Compute $\sigma = \mathsf{Sign}_{\mathsf{sk}_{w'}}(m)$.
- Obtain $\mathsf{sk}_{\mathsf{par}(w')}$ from S and compute $\phi_{w'} = \mathsf{Sign}_{\mathsf{sk}_{\mathsf{par}(w')}}(\mathsf{pk}_{w'})$.
- Remove $\mathsf{sk}_{\mathsf{par}(w')}$ from S if w' is not a left child and add $\mathsf{sk}_{w'}$ to S if w' is not a leaf.
- Given Γ and $(\mathsf{pk}_{w'}, \phi_{w'})$, update Γ to $[(\mathsf{pk}_{w_1}, \phi_{w_1}), \ldots, (\mathsf{pk}_{w'}, \phi_{w'})]$ the signature path from the root to w'.
- Let $\Sigma = (\sigma, \Gamma)$ and $\mathsf{sk}' = (w', S, \Gamma)$.
- Return (Σ, sk') as the signature.

Verification: Given a signature (Σ, sk') on message m, with respect to public key pk, parse $\Sigma = (\sigma, \Gamma)$. Then, verify all signatures in the signature chain Γ and the signature σ on the message. If all signatures are correct, output 1. Otherwise, output 0.

Theorem 40. *Let Π be a 3-time L-spooky-classical-leakage resilient signature scheme. Then, $\mathsf{Comp}(\Pi)$ is a t-time L-spooky-classical-leakage resilient stateful signature scheme.*

Proof. Let $\mathcal{A}^* = (\mathcal{A}^*_{\mathsf{Leak}}, \mathcal{A}^*_{\mathsf{Main}})$ be an adversary that wins in the experiment for the security of the stateful signature scheme described in Definition 56, with probability δ.

We will show how to construct an adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{Leak}}, \mathcal{A}_{\mathsf{Main}})$ that wins with probability δ/t the experiment for the security of the signature scheme described in Definition 53.

We initialize the quantum register for $\mathcal{A}_{\mathsf{Leak}}$ to be the quantum register of $\mathcal{A}^*_{\mathsf{Leak}}$, i.e., $R_1 = R^*_1$. Similarly, we initialize the quantum register of $\mathcal{A}_{\mathsf{Main}}$ to be the quantum register of $\mathcal{A}^*_{\mathsf{Main}}$, i.e. $R_2 = R^*_2$.

Simulation of public key. On input pk from the challenger of the 3-time signature security experiment, sample a node w' at random from the first t nodes. The key $(\mathsf{pk}_{w'}, \mathsf{sk}_{w'})$, where $\mathsf{pk}_{w'} = pk$, will be the challenge key. [Note that $\mathsf{sk}_{w'}$ is not known to $\mathcal{A}_{\mathsf{Main}}$.] The other keys $(\mathsf{pk}_w, \mathsf{sk}_w)$, for $w \neq w'$ are generated (as needed during the simulation of the signing oracle) using the key generation algorithm Gen of the 3-time signature scheme with fresh randomness. Define $\mathsf{pk}' = \mathsf{pk}_\epsilon$ to be the key corresponding to the root node. Send pk' to $\mathcal{A}^*_{\mathsf{Main}}$.

Simulation of signing oracle. Upon receiving a query to the signing oracle on message m, we distinguish two cases. First, the computation of the signature (Σ, sk'') using the current secret key sk' does not require the usage of the challenge secret key $\mathsf{sk}_{w'}$. In this case, simply compute the signature (Σ, sk'') and return it to $\mathcal{A}^*_{\mathsf{Main}}$. Second, computing the signature (Σ, sk'') requires access to the challenge secret key $\mathsf{sk}_{w'}$. In this case, compute the signature (Σ, sk'') using the available signing oracle of the 3-time signature game.

Simulation of leakage query. Upon receiving a leakage query, compute the leakage by querying the adversary $\mathcal{A}^*_{\mathsf{Leak}}$ (who has knowledge on the current secret keys).

The simulation is perfect and has the right distribution. Therefore, the adversary $\mathcal{A}^* = (\mathcal{A}^*_{\mathsf{Leak}}, \mathcal{A}^*_{\mathsf{Main}})$ outputs a forgery (m, Σ) with probability δ. Now we argue that we can extract a forgery with respect to at least one of the keys $(\mathsf{pk}_v, \mathsf{sk}_v)$, $v \in W$, where W is the set of nodes that have been visited during the signature query phase. Parse $\Sigma = (\sigma, \Gamma)$. Let $U = \{\Gamma_v\}_{v \in W}$ be the set of all signature chains that have been generated in the experiment. If $\Gamma \in U$, then $\Gamma = \Gamma_v$ for a node $v \in W$. If Σ is a valid forgery, then $\sigma = \mathsf{Sign}_{\mathsf{sk}_u}(m)$, where m was not queried. Otherwise, if $\Gamma \notin U$, there is a node $v \in W$ such that $\phi \in \Gamma$, with $\phi \in \mathsf{Sign}_{\mathsf{sk}_u}(\mathsf{pk}'')$, where pk'' has not been queried before.

As a consequence, we can extract a forgery with probability δ/t (namely, when the node v from which we can extract the forgery happens to be the challenge node w). □

From Theorems 40 and 39, we obtain the following corollary.

Corollary 9. *Assuming a post-quantum secure universal one-way hash function, there exists a t-time $\theta(n/9)$-spooky-classical-leakage resilient stateful signature scheme, where n is the size of the secret key.*

5.2 Spooky leakage-resilient secret sharing

In this section we study classical secret sharing schemes which are resilient to side-channel attacks mounted by adversaries with quantum capabilities and shared entanglement. We start by introducing the leakage model we will be focusing on.

5.2.1 Leakage model

We consider a quantum analogue of the local leakage model in classical leakage-resilient secret sharing [BDIR18, ADN+19, SV19, CKOS22], which we term *spooky local leakage*.

Suppose that a secret x is distributed into a tuple of n shares $S = (S_1, \ldots, S_n)$ according to some secret sharing scheme. Consider an unauthorized set $T \notin \Gamma$. To each share S_i for $i \notin T$, we associate a local leakage adversary $\mathcal{A}_i$ which has access to S_i and to a quantum register R_i. We allow an arbitrary quantum state $\sigma = \sigma^{R_1,\ldots,R_n}$ to be stored across the registers $R_1, \ldots, R_n$. Note that the contents of different registers may be arbitrarily entangled with each other. Let $\mathcal{A} = (\mathcal{A}_i)_{i \notin T}$. For a given leakage bound ℓ, we say that $(\mathcal{A}, \sigma)$ is an *ℓ-spooky local leakage adversary* if each adversary $\mathcal{A}_i$ produces quantum leakage by applying a quantum circuit to S_i and the contents of R_i which outputs a ℓ-qubit quantum state in $(\mathcal{H})^{\otimes \ell}$ into a leakage register L_i. We denote the contents of L_i by $\mathcal{A}_i(S_i, R_i)$ and the joint adversarial state across all registers $((R_i)_{i \in T}, (L_i)_{i \notin T})$ by $\mathcal{A}(S, \sigma)$.

We are now ready to define secret sharing schemes resilient against spooky local leakage.

Definition 57 (Spooky locally leakage-resilient secret sharing)**.** *We say that a tuple of quantum operations* $(\mathsf{Share}, \mathsf{Rec})$ *is an* (ℓ, ε)-spooky locally leakage-resilient secret sharing scheme *with message space* $\mathcal{X}$ *and share space* $\mathcal{S}$ *for an access structure* Γ *if* $(\mathsf{Share}, \mathsf{Rec})$ *is an* ε*-secret sharing scheme with message space* $\mathcal{X}$ *and share space* $\mathcal{S}$ *for* Γ *which additionally satisfies the following property:*

- ***Spooky local leakage-resilience:*** *Fix any two messages* $x^{(0)}, x^{(1)} \in \mathcal{X}$ *and define* $S^b = \mathsf{Share}(x^{(b)})$ *for* $b \in \{0, 1\}$*. Then, for any* ℓ*-spooky local leakage adversary* $(\mathcal{A}, \sigma)$ *and unauthorized set* $T \notin \Gamma$ *it holds that*

$$S^0_T, \mathcal{A}(S^0, \sigma) \approx_\varepsilon S^1_T, \mathcal{A}(S^1, \sigma).$$

5.2.2 A simple upper bound on the tolerable spooky local leakage rate via superdense coding

In the classical setting, we know several schemes with shares of size N which resist local leakage of $\ell = (1 - \delta)N$ bits from each share for any arbitrary constant $\delta > 0$ [ADN+19, SV19, CKOS22]. It is natural to wonder whether this is achievable in the setting of spooky local leakage. Superdense coding provides a negative answer – spooky leakage-resilience is impossible if the leakage rate is at least $1/2$.

Theorem 41 (Theorem 9, restated)**.** *If there exists an* (ℓ, ε)*-spooky locally leakage-resilient secret sharing scheme with share space* $\{0, 1\}^N$ *and error* $\varepsilon < 1$*, we must have*

$$\ell < N/2.$$

Moreover, the adversary can guess the secret with probability δ *whenever* $\ell \geq \frac{1}{2}(N - \log(1/\delta))$*.*

Proof. We first show that access to $\ell \geq N/2$ qubits of leakage from each share allows an adversary to perfectly recover the secret via superdense coding [NC10, Section 2.3]. This is a method through

which Alice can communicate an arbitrary N-bit string to Bob by operating on her half of $N/2$ shared EPR pairs and then sending them to Bob. Suppose that the secret sharing scheme realizing access structure Γ shares a secret b into n shares $S_1, \ldots, S_n$. Let $T \notin \Gamma$ be an unauthorized set with $T \cup \{i\} \in \Gamma$ for some i. Note that such a set T always exists, by choosing a minimal authorized set and dropping one of the elements. Assume also that the leakage adversaries $\mathcal{A}_i$ and $\mathcal{A}_j$ for some $j \in T$ share $N/2$ EPR pairs. If $\ell \geq N/2$, then we can have $\mathcal{A}_i$ prepare its halves of the $N/2$ EPR pairs according to superdense coding of S_i and leak the resulting $N/2$ qubits. The leakage adversary $\mathcal{A}_j$ simply leaks its half of the $N/2$ EPR pairs. Therefore, an adversary that has access to S_T along with the leakage from $\mathcal{A}_i$ and $\mathcal{A}_j$ can recover S_i and perfectly reconstruct the secret b from (S_T, S_i) since $T \cup \{i\} \in \Gamma$, thus breaking spooky local leakage-resilience.

To see the second part of the theorem statement, consider the modified leakage attack where superdense coding is used to transmit only the first $N - \log(1/\delta)$ bits of S_i, which requires leaking $\frac{1}{2}(N - \log(1/\delta))$ qubits. The adversary can guess the remaining $\log(1/\delta)$ bits of S_i, and hence recover the secret, with probability δ. □

5.2.3 Nearly optimal 2-out-of-2 spooky locally leakage-resilient secret sharing.

We begin by constructing a simple and nearly optimal 2-out-of-2 spooky leakage-resilient secret sharing scheme via quantum-proof two-source extractors.

Observe that the inner product extractor IP (Lemma 7) supports *efficient preimage sampling*, meaning that we can sample uniformly from the preimages $\mathsf{IP}^{-1}(b)$ in an efficient manner. We will exploit this observation in conjunction with Lemma 7 to prove the following result.

Theorem 42 (Theorem 10, restated)**.** *For every leakage bound $\ell \geq 0$ and error $\varepsilon > 0$ there exists an efficient 2-out-of-2 (ℓ, ε)-spooky leakage-resilient secret sharing scheme which shares one bit into two shares of size $N = 2(\ell + \log(1/\varepsilon) - 1)$. In other words, this scheme withstands local leakage of $\ell = N/2 + 1 - \log(1/\varepsilon)$ qubits from each share.*

Proof. Consider the secret sharing scheme which given a bit $b \in \{0,1\}$ samples $(X, Y) \leftarrow \mathsf{IP}^{-1}(b)$ and sets $S_1^b = X$ and $S_2^b = Y$, where $\mathsf{IP} : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}$ is the inner product extractor with N as in the theorem statement. Note that both sharing and reconstruction can be performed efficiently (i.e., in time polynomial in ℓ and $\log(1/\varepsilon)$). The correctness of this scheme is trivial.

We claim that the scheme above is (ℓ, ε)-spooky locally leakage-resilient. Suppose not, for a contradiction. Then, without loss of generality there is a spooky local leakage adversary $(\mathcal{A}, \sigma)$ with $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and a distinguisher $\mathcal{D}$ with 1-bit output such that

$$\Pr\left[\mathcal{D}(S_1^1, \rho_{\mathsf{Leak}}^{\mathcal{A},1}) = 1\right] - \Pr\left[\mathcal{D}(S_1^0, \rho_{\mathsf{Leak}}^{\mathcal{A},0}) = 1\right] > \varepsilon. \tag{15}$$

We now use $\mathcal{D}$ to construct another distinguisher $\mathcal{D}'$ which breaks the extractor property of IP, contradicting Lemma 8.

Let X and Y be independent and uniformly distributed over $\{0,1\}^N$. Consider the game where a challenger computes the local leakages $\mathcal{A}((X,Y),\sigma)$ using the leakage adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ above on input $(X, \sigma_{\mathsf{R}_1})$ and $(Y, \sigma_{\mathsf{R}_2})$, respectively, and provides $\mathcal{A}((X,Y),\sigma)$, X, and a bit Z, where Z is either $\mathsf{IP}(X,Y)$ or independent and uniformly random. By Lemma 8 and the choice of N, it follows that for every distinguisher $\mathcal{D}'$ with 1-bit output we have

$$\left|\Pr\left[\mathcal{D}'(\mathcal{A}((X,Y),\sigma), X, Z = \mathsf{IP}(X,Y)) = 1\right] - \Pr[\mathcal{D}(\mathcal{A}((X,Y),\sigma), X, Z = U_1) = 1]\right| \leq \varepsilon.$$

However, consider the distinguisher $\mathcal{D}'$ which on input $(\mathcal{A}((X,Y),\sigma), X, Z)$ computes

$$\widetilde{b} = \mathcal{D}(X, \mathcal{A}((X,Y),\sigma))$$

and outputs 1 if and only if $\widetilde{b} = Z$. Then, we have

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}'(\mathcal{A}((X,Y),\sigma),X,Z=\mathsf{IP}(X,Y))=1\right]-\Pr[\mathcal{D}(\mathcal{A}((X,Y),\sigma),X,Z=U_1)=1]\right| \\
&=\left|\Pr\left[\mathcal{D}'(\mathcal{A}((X,Y),\sigma),X,Z=\mathsf{IP}(X,Y))=1\right]-\frac{1}{2}\right| \\
&=\left|\Pr\left[\widetilde{b}=\mathsf{IP}(X,Y)\right]-\frac{1}{2}\right| \\
&=\left|\frac{1}{2}\Pr\left[\widetilde{b}=\mathsf{IP}(X,Y)|\mathsf{IP}(X,Y)=0\right]+\frac{1}{2}\Pr\left[\widetilde{b}=\mathsf{IP}(X,Y)|\mathsf{IP}(X,Y)=1\right]-\frac{1}{2}\right| \\
&=\left|\frac{1}{2}\Pr\left[\mathcal{D}(X^0,\mathcal{A}((X^0,Y^0),\sigma))=0\right]+\frac{1}{2}\Pr\left[\mathcal{D}(X^1,\mathcal{A}((X^1,Y^1),\sigma))=1\right]-\frac{1}{2}\right| \\
&>\varepsilon,
\end{aligned}$$

where the last inequality follows from Equation (15). This contradicts Lemma 8. □

Note that, because of Theorem 9, the local leakage bound ℓ withstood by the scheme from Theorem 10 is nearly optimal with respect to the share size and the error ε.

5.2.4 Spooky locally leakage-resilient secret sharing for all 3-monotone access structures

We now show how we can leverage the 2-out-of-2 scheme from Section 5.2.3 to obtain a compiler for spooky leakage-resilient secret sharing over any 3-monotone[13] access structure. To this end, we show that the approach from [ADN+19] carries over to a setting with shared entanglement if we use a quantum-proof two-source extractor with the appropriate output length and supporting efficient preimage sampling.

Let $(\mathsf{Share},\mathsf{Rec})$ be an arbitrary (not necessarily leakage-resilient) secret sharing scheme realizing a given access structure Γ sharing one bit into n shares of size M. Suppose we have access to an explicit quantum-proof two-source extractor $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}^M$ that supports efficient preimage sampling. Then, we consider a modified sharing procedure $\overline{\mathsf{Share}}$ defined as follows on input $b \in \{0,1\}$:

1. Compute $(S'_1,\dots,S'_n) = \mathsf{Share}(b)$;
2. For $i \in [n]$, sample $(X_i,Y_i) \leftarrow \mathsf{Ext}^{-1}(S'_i)$;
3. Set the i-th share as $S_i = (X_i, Y_{-i})$, where $Y_{-i} = (Y_1,\dots,Y_{i-1},Y_{i+1},\dots,Y_n)$.

Correctness of this compiled scheme is straightforward: We can efficiently recover b given any set of shares $T \in \Gamma$. Note also that the total share size is $n^2 \cdot N$, and that sharing is efficient. It remains to argue that the compiled scheme is spooky leakage-resilient when we instantiate Ext appropriately. To this end, we use the extractor from Lemma 8 and obtain the following result.

Theorem 43 (Theorem 40, restated)**.** *Given an access structure Γ over n parties and a δ-statistically private secret sharing scheme realizing Γ with shares of size M, there exists an $(\ell, \delta + \varepsilon \cdot n \cdot 2^M)$-spooky leakage-resilient secret sharing scheme realizing Γ with shares of size $n \cdot N$ with*

$$N = (M + \ell + \log(1/\varepsilon) - 1).$$

Moreover, the compiled scheme is efficient whenever the underlying one is too.

[13] We say that an access structure Γ is *t-monotone* if $|T| \geq t$ for every authorized set $T \in \Gamma$.

Before we prove Theorem 40, we introduce the following useful lemma generalizing [CG17, Proposition B.3].

Lemma 16. *Let (Z,ρ) and (Z',ρ') be mixed quantum states where Z and Z' are classical over a common set $\mathcal{Z}$. Suppose that $(Z,\rho) \approx_\varepsilon (Z',\rho')$ and let $E \subseteq \mathcal{Z}$ be a set such that $\Pr[Z \in E] = p$. Then, it holds that*

$$(Z,\rho|Z \in E) \approx_{\varepsilon/p} (Z',\rho'|Z' \in E).$$

Proof. Suppose that

$$(Z,\rho|Z \in E) \not\approx_{\varepsilon/p} (Z',\rho'|Z' \in E)$$

for some event $E \subseteq \mathcal{Z}$ with $\Pr[Z \in E] = p$. By the operational interpretation of trace distance, this means that there is a distinguisher $\mathcal{D}$ which correctly guesses whether it is interacting with the mixed state $(Z,\rho|Z \in E)$ or $(Z',\rho'|Z' \in E)$ with probability larger than $1/2 + \varepsilon/p$.

Consider the following distinguisher $\mathcal{D}'$ with 1-bit output which aims to distinguish between (Z,ρ) and (Z',ρ'): Given some mixed classical-quantum state $\sigma_{\mathsf{R}_1,\mathsf{R}_2}$ where the contents of register R_1, denote them by W, are classical, $\mathcal{D}'$ first checks whether $W \in E$. Note that this can be done without disturbing the overall quantum mixed state since W is classical. If this is not the case, then $\mathcal{D}'$ outputs a random bit. Otherwise, $\mathcal{D}'$ invokes $\mathcal{D}$ on $\sigma_{\mathsf{R}_1,\mathsf{R}_2}$ and simply outputs whatever $\mathcal{D}$ guesses. Then, the probability that $\mathcal{D}'$ guesses (Z',ρ') when $\sigma_{\mathsf{R}_1,\mathsf{R}_2} = (Z,\rho)$ is smaller than

$$\Pr[Z \notin E] \cdot \frac{1}{2} + \Pr[Z \in E](1/2 - \varepsilon/p) = 1/2 - \varepsilon,$$

which contradicts the fact that $(Z,\rho) \approx_\varepsilon (Z',\rho')$. □

We now proceed to the proof of Theorem 40.

Proof of Theorem 40. The claims about share size and efficiency are straightforward. We prove spooky local leakage-resilience via a hybrid argument.

Fix shares $(S'^b_1,\dots,S'^b_n)$ of $b \in \{0,1\}$ from the underlying secret sharing scheme $(\mathsf{Share},\mathsf{Rec})$, an unauthorized set $T \notin \Gamma$ and a tuple of local leakage adversaries $\mathcal{A} = (\mathcal{A}_1,\dots,\mathcal{A}_n)$ with arbitrary shared entanglement and ℓ-qubit output. Without loss of generality, rename parties so that $T = \{1,\dots,t\}$. Let $(S^b_1,\dots,S^b_n)$ denote the shares obtained by sharing a bit $b \in \{0,1\}$ using $\overline{\mathsf{Share}}$, and $\rho^{\mathcal{A},b}_{\mathsf{Leak}}$ the corresponding spooky quantum leakage. Consider the modified experiment which proceeds exactly like $\overline{\mathsf{Share}}(b)$ but replaces (X_{t+1},Y_{t+1}) by (X^*_{t+1},Y^*_{t+1}) both sampled independently and uniformly at random from $\{0,1\}^N$. Denote the shares and leakage resulting from this modified experiment by $(S^{b,t+1}_1,\dots,S^{b,t+1}_n)$ and $\rho^{\mathcal{A},b,t+1}_{\mathsf{Leak}}$. Then, by the choice of N in the theorem statement and Lemma 8 it holds that

$$\mathsf{Ext}(X^*_{t+1},Y^*_{t+1}),S^{b,t+1}_T,\rho^{\mathcal{A},b,t+1}_{\mathsf{Leak}} \approx_\varepsilon U_M,S^{b,t+1}_T,\rho^{\mathcal{A},b,t+1}_{\mathsf{Leak}}. \tag{16}$$

This is so because we can see $\rho^{\mathcal{A},b,t+1}_{\mathsf{Leak}}$ as spooky local ℓ-qubit leakage on X^*_{t+1} and unbounded leakage on Y^*_{t+1}, since only $S^{b,t+1}_{t+1}$ depends on X^*_{t+1}. Consider now the event that $U_M = S'^b_{t+1}$. Since this event holds with probability 2^{-M}, it follows from Equation (16) and Lemma 16 that

$$S^b_T,\rho^{\mathcal{A},b}_{\mathsf{Leak}} \approx_{\varepsilon\cdot 2^M} S^{b,t+1}_T,\rho^{\mathcal{A},b,t+1}_{\mathsf{Leak}},$$

since $S^{b,t+1}_T,\rho^{\mathcal{A},b,t+1}_{\mathsf{Leak}}$ correspond exactly to $S^b_T,\rho^{\mathcal{A},b}_{\mathsf{Leak}}$ when conditioned on the event $\mathsf{Ext}(X^*_{t+1},Y^*_{t+1}) = S'^b_{t+1}$.

More generally, let $(S_1^{b,t+1}, \ldots, S_n^{b,j})$ and $\rho_{\mathsf{Leak}}^{\mathcal{A},b,j}$ be the modified shares and leakage obtained via the modified $\overline{\mathsf{Share}}(b)$ experiment where $(X_i, Y_i)_{i=t+1,\ldots,j}$ are all replaced by (X_i^*, Y_i^*) independent and uniformly distributed over $\{0,1\}^N$. Repeating the argument from the previous paragraph $n-t$ times and repeatedly invoking the triangle inequality, we conclude that

$$S_T^b, \rho_{\mathsf{Leak}}^{\mathcal{A},b} \approx_{n\cdot\varepsilon\cdot 2^M} S_T^{b,n}, \rho_{\mathsf{Leak}}^{\mathcal{A},b,n}.$$

Observe that $\rho_{\mathsf{Leak}}^{\mathcal{A},b,n}$ is a function of $S_T'^b$ only. Therefore, since $S_T'^0 \approx_\delta S_T'^1$ by the δ-statistical privacy of $(\mathsf{Share}, \mathsf{Rec})$, it follows by the triangle inequality that

$$S_T^0, \rho_{\mathsf{Leak}}^{\mathcal{A},0} \approx_{\delta+n\cdot\varepsilon\cdot 2^M} S_T^1, \rho_{\mathsf{Leak}}^{\mathcal{A},1},$$

as desired. □

5.3 Spooky leakage-resilient computation

In this section we present a compiler that transforms general computations so that they can be run securely even in the presence of spooky leakage attacks. The compiler was introduced by Goldwasser and Rothblum [GR12], and we show that it tolerates a *spooky-leakage* adversary, who learns a bounded local leakage on each computation instruction, where the leakage functions are computed using quantum circuits with classical output, and are allowed to share arbitrary entanglement.

The computation is represented by a circuit C taking two inputs, a secret input y and a public input x, and an output. For example, C can be a public encryption algorithm, y a secret key, and x a message (known to the encryptor) to encrypt. The compiler takes as input the circuit C and transforms it into an algorithm that is resilient to spooky-leakage.

More concretely, the leakage model considered is the so-called *only computation leaks* (alternatively, the leaky CPU) model, where the algorithm is composed of a sequence of instructions, which are basic subcomputations coming from a fixed universal set of instructions. The adversary learns a bounded local classical leakage on each operand to an instruction when it is executed, and the leakage functions have access to arbitrary entanglement. The leakage on each instruction can be the result of a computationally unbounded function of the internal state of the instruction.

Universal set of instructions. The set of instructions that we need for our purposes are: generating a random matrix/vector of bits, addition and multiplications of matrices, and permuting a sequence of vectors. All these instructions can be computed by circuits with small polynomial size.

Security definition. The input to the compiler is a circuit C that is known to all parties, and takes two inputs. The input y is fixed and secret, whereas the input x is chosen by the user for evaluation. Security requires that for any unbounded adversary choosing the input x, and with access to spooky leakage on the transformed computation, the adversary learns nothing more than the circuit's outputs.

The compiler consists of two parts, the initialization and the evaluation. The initialization phase occurs once at the beginning and only depends on the circuit C and the secret input y. This phase occurs with no leakage. The evaluation phase then occurs whenever the user wants to evaluate the circuit $C(y, \cdot)$ on input x, and is computed under leakage.

We consider the notion of L-spooky leakage, which is similar to the one introduced in Section 5.2.1 introduced for secret sharing, but adjusted to this setting. More concretely, the adversary has access to outputs of leakage functions on each instruction step I, where the leakage function

takes as input the input to the instruction and any randomness used in the instruction, as well as a quantum register R_I, and outputs at most L bits. The leakage functions are decided in advance, and consist of a quantum circuit with classical output, and the registers $\{R_I\}_I$ may be arbitrarily entangled with each other.

Definition 58 (L- Spooky Leakage Secure Compiler)**.** *Let λ be a security parameter and n be a natural number. We say that a compiler $(\mathsf{Init}, \mathsf{Eval})$ for a circuit C on two n-bit inputs is L-spooky-classical-leakage-resilient secure if for every $y \in \{0,1\}^n$ the following holds:*

- *Initialization: $\mathsf{Init}(1^\lambda, C, y)$ runs in time $\mathrm{poly}(\lambda, n)$ and outputs an initial state st.*
- *Evaluation: the evaluation procedure Eval is run on state st and input $x \in \{0,1\}^n$. We require that with overwhelming probability (over the randomness of Init and invocation of Eval), $C(y,x) = \mathsf{Eval}(\mathsf{st}, x)$.*
- *L-Spooky-Classical-Leakage Security: Consider an execution of $\mathsf{Init}(1^\lambda, C, y)$ returning st, followed by an execution of $\mathsf{Eval}(\mathsf{st}, x)$. Let $\mathsf{Real}_{\mathcal{A}}$ be the view of the adversary in the whole execution, including input x, output of $\mathsf{Eval}(\mathsf{st}, x)$ and L-spooky-classical-leakage on $\mathsf{Eval}(\mathsf{st}, x)$ (meaning, local $L(\lambda)$-bit bounded leakage from each basic instruction step, with access to arbitrary entanglement). Further let $\mathsf{Ideal}_{\mathcal{A}}$ be the output of the simulator $\mathcal{S}$. The simulator includes $\mathcal{S}_{\mathsf{Init}}$ and $\mathcal{S}_{\mathsf{Eval}}$ procedures, where $\mathcal{S}_{\mathsf{Init}}$ generates the initial state st from the description of the circuit C, and $\mathcal{S}_{\mathsf{Eval}}$ generates the simulated leakage from the input x chosen by the adversary and the circuit output $C(y,x)$.*

  We require that there exists a simulator $\mathcal{S}$ such that the view $\mathsf{Real}_{\mathcal{A}}$ of every (potentially computationally unbounded) adversary $\mathcal{A}$ choosing input x, and with access to L-spooky leakage on $\mathsf{Eval}(\mathsf{st}, x)$, is overwhelming statistically close to the view $\mathsf{Ideal}_{\mathcal{A}}$ generated by $\mathcal{S}$, which only gets the description of the adversary and the pair $(x, C(y,x))$.

5.3.1 Spooky leakage-resilient encryption

A main ingredient of the compiler construction is a spooky leakage resilient one-time pad encryption scheme LROTP. We use the scheme from [GR12] (originally proven secure against classical adversaries) and show that it is actually resilient under spooky bounded leakage attacks on the key and the ciphertext. The encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ works as follows:

- Key generation: $\mathsf{KeyGen}(1^\lambda)$ outputs a uniform random key $k \in \{0,1\}^\lambda$ such that $k[0] = 1$.
- Encryption: Given a message $m \in \{0,1\}$ and a key $k \in \{0,1\}^\lambda$, the encryption algorithm $\mathsf{Enc}(k, m)$ outputs a uniform λ-bit ciphertext $c \in \{0,1\}^\lambda$ such that $c[1] = 1$ and $\langle k, c\rangle = m$.
- Decryption: Given key $k \in \{0,1\}^\lambda$ and ciphertext $c \in \{0,1\}^\lambda$, the decryption algorithm $\mathsf{Dec}(k, c)$ outputs the message bit $\langle k, c\rangle$.

Semantic security under L-spooky-classical-leakage. We need that statistical security holds against a spooky adversary who launches attacks on both key and ciphertext. The attack applies bounded leakage functions to each part, where the leakage functions have access to arbitrary entanglement. Intuitively, because the leakage is of bounded length and operates separately on key and ciphertext, these remain high entropy sources and are independent (up to their inner product being the plaintext). But since the inner product is a strong quantum-proof two-source extractor (see Lemma 7), the plaintext will be statistically close to uniform even with leakage.

Definition 59. *An encryption scheme* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is semantically secure under L-spooky-classical-leakage attacks if for every (potentially unbounded) adversary $\mathcal{A}$, the adversary's winning probability in the following game described below is at most* $1/2 + \mathsf{negl}(\lambda)$*:*

- *The game chooses key $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, chooses uniformly at random a bit $b \in \{0,1\}$, and generates a ciphertext $c = \mathsf{Enc}(k, b)$.*
- *The adversary launches a L-bounded local spooky leakage attack on k and c, and outputs a guess b':*
$$b' \leftarrow \mathcal{A}^{L(\lambda)}(1^\lambda)[k, c].$$
The adversary wins if $b' = b$.

Lemma 17. *The one-time pad spooky leakage-resilient* LROTP *scheme is semantically secure under L-spooky leakage, for $L = \lambda/3$.*

Proof. The key and ciphertext are both high entropy sources, satisfying $\langle k, c\rangle = b$, and from Lemma 7 we know that IP is a strong quantum-proof two-source extractor for $L = \lambda/3 \leq \lambda - \lambda/2 - \log(1/\varepsilon) + 1$, for $\varepsilon = 2^{-\lambda/6}$. This means that the plaintext b is statistically close to uniform even given L-bit spooky leakage. □

Key and ciphertext refreshing. The LROTP scheme allows to refresh both the key and ciphertext, injecting new entropy while maintaining the corresponding plaintext, using the methods described below. One can generate new entropy key σ and ciphertext τ, using the methods KeyEntGen and CipherEntGen, and refresh a known key or ciphertext using KeyRefresh and CipherRefresh. Moreover, using CipherCorrelate, and without knowledge of a key k, one can correlate a known ciphertext c into a ciphertext c^*, so that the plaintext of c^* under key $k' \leftarrow \mathsf{KeyRefresh}(k, \sigma)$ is the same as the plaintext of c under k. And similarly one can correlate keys using ciphertext entropy with the method KeyCorrelate.

- $\mathsf{KeyEntGen}(1^\lambda)$: output a uniformly random $\sigma \in \{0,1\}^\lambda$ such that $\sigma[0] = 0$.
- $\mathsf{KeyRefresh}(k, \sigma)$: output $k \oplus \sigma$.
- $\mathsf{CipherCorrelate}(c, \sigma)$: modify $c[0] \leftarrow c[0] \oplus \langle c, \sigma\rangle$, and output c.
- $\mathsf{CipherEntGen}(1^\lambda)$: output a uniformly random $\tau \in \{0,1\}^\lambda$ such that $\tau[1] = 0$.
- $\mathsf{CipherRefresh}(c, \tau)$: output $c \oplus \tau$.
- $\mathsf{KeyCorrelate}(k, \tau)$: modify $k[1] \leftarrow k[1] \oplus \langle k, \tau\rangle$, and then output k.

Moreover, we note that one can refresh a key k_i and a ciphertext c_i separately into an updated ciphertext c_i^* encrypted under a newly generated key k such that c_i^* (under key k) contains the same plaintext as c_i (under key k_i). This is performed as follows:

1. Generate a new key $k \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. Let $\sigma_i \leftarrow k_i \oplus k$.
3. Let $c_i^*[0] \leftarrow c_i[0] \oplus \langle c_i, \sigma_i\rangle$, and $c_i^*[j] = c_i[j]$ for $j > 0$.

Homomorphic operations. The LROTP scheme also allows for homomorphic addition of ciphertexts c_1, c_2 under the same k, by computing $c \leftarrow c_1 \oplus c_2$ (bit-wise). By linearity, the plaintext underlying c is the XOR of plaintexts underlying c_1 and c_2. Moreover, one can add a plaintext a to a ciphertext c (encrypting x under k) to generate a new ciphertext $c' = c \oplus (a, 0, \ldots, 0)$ encrypting $x \oplus a$ under k.

5.3.2 Compiler overview

Now we proceed to give an overview of the compiler Π_{Comp}. The compiler takes as input a secret input y and a public circuit C. The circuit C takes as input the secret y and a public input x, and outputs a single bit. The compiler then outputs a functionally equivalent algorithm Eval, i.e. $C(y, x) = \mathsf{Eval}_y(x)$ for all x, consisting of a sequence of instructions, where the adversary can learn bounded-length leakage on each instruction. Importantly, we assume that the compiler is run once at the beginning and is not subject to leakage. In particular, there is no leakage on the secret y. The leakage only occurs during the execution of $\mathsf{Eval}_y(x)$.

Let us assume that the circuit C is described of NAND gates. The high-level idea is to keep track of each intermediate wire value v_i of the circuit $C(y, x)$ in an encrypted form via a pair key-ciphertext (k_i, c_i). The value v_i will be protected because no instruction makes use of the key and ciphertext at once, and the encryption scheme tolerates spooky bounded local leakage on key and ciphertext (see Section 5.3.1). Moreover, we will have a procedure that allows to compute the NAND gate in order to evaluate the circuit gate by gate. Details follow below.

Initialization. For each y-input wire i with bit value $y[j]$, generate a leakage-resilient one-time pad encryption of $y[j]$, (k_i, c_i). For each x-input wire i, generate an encryption of 0, (k_i, c_i). For each internal wire i, choose a random bit r_i and generate two encryptions of r_i, denoted (l_i, d_i) and (l'_i, d'_i). Finally, for each internal wire i (including the output wire), generate an encryption of 1, (o_i, e_i).

Evaluation. Given input x, the algorithm Eval transforms the ciphertexts c_i (initially encoding 0 for each x-input wire) to ciphertexts encoding each bit $x[j]$, simply by homomorphically adding the 0-ciphertext with the plaintext $x[j]$: $c_i = c_i + (x[j], 0, \ldots, 0)$.

The algorithm Eval then simply computes a pair (k_i, c_i) for each wire v_i following a fixed topological order gate-by-gate from the inputs to the output. For each NAND gate, given encryptions of the two bits on the input wires, it computes an encryption of the output wire. The final output ciphertext is then decrypted.

Computation of NAND. Given two ciphertext pairs (k_i, c_i) and (k_j, c_j) encrypting wire values v_i and v_j, the procedure first computes the bit $a_k = (v_i \ \mathsf{NAND} \ v_j) \oplus r_k$, for random r_k. This is performed in four steps:

1. Key unification: Choose a single key k, and compute ciphertexts $(c^*_i, c^*_j, d^*_k, e^*_k)$ encrypting $(v_i, v_j, r_k, 1)$ under the same key k, using the ciphertext pairs (k_i, c_i), (k_j, c_j), (l_k, d_k) and (o_k, e_k), encrypting the wire values v_i and v_j, and random value r_k and 1 (corresponding to the output wire k, generated in the initialization phase), respectively. See Section 5.3.1.

2. Tuple generation via homomorphism: Given the above ciphertexts under the same key, use the homomorphic property of LROTP (see Section 5.3.1) to compute the tuple

$$C \leftarrow (d^*_k, (c^*_i \oplus d^*_k), (c^*_j \oplus d^*_k), (c^*_i \oplus c^*_j \oplus d^*_k \oplus e^*_k)),$$

which are ciphertexts encrypting $(r_k, (v_i \oplus r_k), (v_j \oplus r_k), (v_i \oplus v_j \oplus r_k \oplus 1))$.

3. Tuple permutation: This procedure takes as input a key k and the tuple of four ciphertexts, and outputs four randomly permuted fresh pairs of key-ciphertext, encrypting the same plaintexts. Details follow below.

4. Output determination: Decrypt the four ciphertexts. If there is one 0, let $a_k = 0$. Otherwise, let $a_k = 1$. Compute the output value as $k_k = l'_k$ and $c_k = d'_k \oplus (a_k, 0, \ldots, 0)$, where d'_k is the second ciphertext encrypting r_k under key l'_k from the initialization phase.

Correctness of the NAND operation follows from the fact that the tuple of ciphertexts C contains a 0 if and only if $(v_i \ \mathsf{NAND} \ v_j) \oplus r_k = 0$, and correctness of the initialization phase. At a very high level, security will hold from three main aspects: first, the bit a_k can be made public since r_k is random; second, the one-time pad encryption is spooky classical-leakage-resilient and supports homomorphic plaintext addition; finally, the permutation procedure is made statistically close to random even under leakage.

Leakage-resilient tuple permutation. This procedure takes as input a key k and the tuple of four ciphertexts (under the same key k), and outputs four randomly permuted fresh pairs of key-ciphertext. The procedure guarantees 1) correctness, meaning that the underlying plaintexts are a random permutation of the plaintexts corresponding to the input ciphertexts; and 2) security, meaning that the permutation is random in the view of a (possibly) unbounded adversary. More precisely, there is a simulator that generates the leakage and the pairs of output key-ciphertext, with access only to the marginal distribution from which the input key and ciphertexts are drawn, and a random permutation of the plaintexts underlying the input ciphertexts. The joint distribution of the leakage and the outputs is independent of the used permutation.

The procedure proceeds in iterations. At each iteration, the input and output is a tuple of four pairs of key-ciphertext, where the output ciphertexts have as underlying plaintexts some permutation of the input ciphertexts. At each iteration, the permutation will look fairly random to a leakage adversary, and the composition of permutations will be statistically close to uniform. A description of $\mathsf{Permute}(k, C)$, where k is the input key and C is the tuple of four ciphertexts follows:

- Take $K_0 \leftarrow (k, k, k, k)$, $C_0 \leftarrow C$, and $\ell = \mathrm{poly}\log(\lambda)$.
- For $i \in [\ell]$:
  1. for $j \in [\lambda]$, $k \in [4]$: $\sigma_i[j][k] \leftarrow \mathsf{KeyEntGen}(1^\lambda)$, $L_i[j][k] \leftarrow \mathsf{KeyRefresh}(K_i[k], \sigma_i[j][k])$.
  2. for $j \in [\lambda]$, $k \in [4]$: $D_i[j][k] \leftarrow \mathsf{CipherCorrelate}(C_i[k], \sigma_i[j][k])$.
  3. for $j \in [\lambda]$, $k \in [4]$: $\tau_i[j][k] \leftarrow \mathsf{CipherEntGen}(1^\lambda)$, $D'_i[j][k] \leftarrow \mathsf{CipherRefresh}(D_i[j][k], \tau_i[j][k])$.
  4. for $j \in [\lambda]$, $k \in [4]$: $L'_i[j][k] \leftarrow \mathsf{KeyCorrelate}(L_i[j][k], \tau_i[j][k])$.
  5. pick $\pi_i \in_R S_4^\lambda$, for $j \in [\lambda]$: $L''_i[j] \leftarrow \pi_i[j](L'_i[j])$, $D''_i[j] \leftarrow \pi_i[j](D'_i[j])$.
  6. pick $j_i^* \in_R [\lambda]$. Save $K_{i+1} \leftarrow L''_i[j_i^*]$, and $C_{i+1} \leftarrow D''_i[j_i^*]$.
- Output (K_ℓ, C_ℓ)

Correctness of the permutation procedure is immediate. Security is formalized by the existence of a simulator that generates the view of the leakage attack. The attack proceeds in two phases: first an adversary $\mathcal{A}_1$ mounts an attack operating separately on key k and tuple of four ciphertexts C. Then, a second adversary $\mathcal{A}_2$ performs an attack on the execution of Permute with those inputs.

The simulator only gets a random permutation of the plaintexts underlying the input k and C, and simulates the leakage generated by $\mathcal{A}_1$ and $\mathcal{A}_2$. The following lemma follows from the security of the one-time pad scheme, and its proof is presented in [GR12].

Lemma 18. *There is a simulator $\mathcal{S}_{\mathsf{Permute}}$ and a leakage bound $L(\lambda) = \tilde{\Omega}(\lambda)$ such that for any λ, leakage adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$, and any bit values $(b_1, b_2, b_3, b_1 + b_2 + b_3 + 1)$ the following views are statistically close:*

$$\mathit{REAL} = \left(\mathcal{A}_1^{L(\lambda)}[k, C], \mathcal{A}_2^{L(\lambda)}[(K', C') \leftarrow \mathsf{Permute}(k, C)], K', C'\right)_{(k,(c_1,c_2,c_3))\sim D}$$

$$\mathit{IDEAL} = \left(\mathcal{S}_{\mathsf{Permute}}(\vec{b}', \mathcal{K}, \mathcal{C})\right)_{\mu \in_R S_4, \vec{b}' \leftarrow \mu(\vec{b})},$$

where D is the distribution for uniform key k and ciphertexts c_1, c_2 and c_3 encrypting the bits b_1, b_2 and b_3 (respectively), and where $C = (c_1, c_2, c_3, (c_1 \oplus c_2 \oplus c_3 \oplus (1, 0, \ldots, 0)))$.

Proof. The simulator takes as input a uniform random permutation $\vec{b}'$ of the input bits. It outputs a leakage and the output (K', C') as follows. First sample a pair (k, C), where C contains encryptions of 0 (instead of $\vec{b}$ as in the real world).

Then compute the leakage $w = (\mathcal{A}_1^{L(\lambda)}[k, C], \mathcal{A}_2^{L(\lambda)}[(K', C') \leftarrow \mathsf{Permute}(k, C)])$. The adversary then samples K' according to the conditional distribution where the leakage is w, the permutation is π (the composed distribution used by the Permute procedure), and the pair key-ciphertext is (k, C). Similarly, sample C' from the conditional distribution where the leakage is w, the permutation is π, the pair key-ciphertext is (k, C), and the inner products of C' and K' are the bits in $\vec{b}'$. The simulator outputs (w, K', C').

Note that the pair (k, C) chosen in the real and ideal distributions only differ in the fact that the plaintexts of C are $\vec{b}$ in the real distribution, and 0 in the ideal distribution. Further note that the Permute procedure operates separately on k and on C. By the security of the one-time pad scheme, the distributions of the leakage generated in the real and ideal transcripts are statistically close. Now we need to argue that the joint distribution of (K', C') in both real and ideal are statistically close. For that, we first consider a hybrid distribution, which is generated exactly the same as $\mathcal{S}_{\mathsf{Permute}}$ does, except that C' is generated conditioning on the fact that the inner products of C' and K' is $\pi(\vec{b})$, rather than $\vec{b}'$.

To see that this hybrid distribution is statistically close to the real distribution, observe that one can sample the real distribution as follows: first sample (k, C), where k is uniform, and C contains encryptions of the real bits $\vec{b}$, and then follow exactly the steps of the hybrid distribution (where the considered key-ciphertexts tuple is (k, C)). By semantic security of the LROTP scheme, the distributions for real and hybrid are statistically close.

To see that the hybrid distribution is statistically close to the ideal distribution, observe that when the pair (k, C), where k is uniform, and C contains encryptions of 0, the composed permutation π is indistinguishable from uniform random. As such, the only difference between the hybrid and ideal distributions, is that in the hybrid distribution we condition (K', C') on the permutation being π and the leakage w, whereas in the ideal distribution this is conditioned on a uniformly random composed permutation $(\mu \circ \pi)$. But drawing from these two conditional distributions yields statistically close views. □

The following theorem proves that the compiler is secure. The proof is taken from the proof presented in [GR12].

Theorem 44. *There is a leakage bound function $L = \Theta(\lambda)$ such that for every λ the compiler Π_{Comp} is a $L(\lambda)$-spooky leakage secure compiler as defined in Definition 58.*

Proof. We provide a rough sketch of the theorem. The simulator is composed of two parts, $\mathcal{S}_{\mathsf{Init}}$ and $\mathcal{S}_{\mathsf{Eval}}$. The $\mathcal{S}_{\mathsf{Init}}$ starts by initializing all ciphertexts as in the initialization step of the real protocol. Then, the simulator $\mathcal{S}_{\mathsf{Eval}}$ computes the values on all the internal wires for the circuit computation $C(0, x)$, where x is the input. For each internal wire, it chooses a value $a_i \in_R \{0, 1\}$ and the key-ciphertext pair (k_i, c_i), and the output wire is set to the real value $a_{\mathsf{out}} = C(y, x)$.

The simulator then follows exactly the protocol steps, where the NAND computation is simulated as follows:

The simulator uses the wire value a_k and it chooses bits (v'_i, v'_j, r'_k) such that $a_k = (v'_i \mathsf{\ NAND\ } v'_j) \oplus r'_k$. The simulator runs the leakage attack on freshly generated keys and ciphertexts encrypting these bit values, and outputs its view. Given that the tuple permutation operation does not reveal the order of the ciphertexts, the NAND computation can be simulated as follows.

In the case of the real attack, we have the tuple $(r_k, (v_i \oplus r_k), (v_j \oplus r_k), (v_i \oplus v_j \oplus r_k \oplus 1))$, and in the ideal attack we have the tuple $(r'_k, (v'_i \oplus r'_k), (v'_j \oplus r'_k), (v'_i \oplus v'_j \oplus r'_k \oplus 1))$. Note that in both cases, the number of 0's and 1's is the same (three 0's if $a_k = 1$ and one 0 if $a_k = 0$). Therefore, both tuples are distributed identically. □

6 Acknowledgements

Alper Cakan 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.

[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.

[AKL+22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. On the feasibility of unclonable encryption, and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 212–241, Cham, 2022. Springer Nature Switzerland.

[AKN+23] Shweta Agrawal, Fuyuki Kitagawa, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Public key encryption with secure key leasing. In *Advances in Cryptology – EUROCRYPT 2023*, 2023. To appear.

[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.

[BK22] James Bartusek and Dakshita Khurana. Cryptography with certified deletion. Cryptology ePrint Archive, Paper 2022/1178, 2022. https://eprint.iacr.org/2022/1178.

[BL20] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via oracles. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:22, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CG17] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. *J. Cryptol.*, 30(1):191–241, 2017.

[CHSH69] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[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.

[CLMW10] Toby S Cubitt, Debbie Leung, William Matthews, and Andreas Winter. Improving zero-error classical communication with entanglement. *Physical Review Letters*, 104(23):230503, 2010.

[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.

[DEOR04] Yevgeniy Dodis, Ariel Elbaz, Roberto Oliveira, and Ran Raz. Improved randomness extraction from two independent sources. In Klaus Jansen, Sanjeev Khanna, José D. P. Rolim, and Dana Ron, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 334–344, Berlin, Heidelberg, 2004. 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.

[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.

[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.

[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.

[GKK+09] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM Journal on Computing*, 38(5):1695–1708, 2009.

[GR12] Shafi Goldwasser and Guy N Rothblum. How to compute in the presence of leakage. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 31–40. IEEE, 2012.

[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.

[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.

[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.

[KRS99] Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Coding constructions for blacklisting problems without computational assumptions. In *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*, pages 609–623. Springer, 1999.

[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.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.

[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.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography. In *Theory of Cryptography Conference*, pages 278–296. Springer, 2004.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[NR98] Moni Naor and Omer Reingold. From unpredictability to indistinguishability: A simple construction of pseudo-random functions from macs. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 267–282, Berlin, Heidelberg, 1998. 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.

[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.

[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.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, jan 1983.