Powerful Primitives in the Bounded Quantum Storage Model

Mohammed Barhoush
Université de Montréal (DIRO), Canada

Louis Salvail
Université de Montréal (DIRO), Canada

Abstract

The bounded quantum storage model aims to achieve security against computationally unbounded adversaries that are restricted only with respect to their quantum memories. In this work, we provide the following contributions in this model:

1. We build one-time programs and utilize them to construct CCA1-secure symmetric key encryption and message authentication codes. These schemes require no quantum memory from honest users, yet they provide information-theoretic security against adversaries with arbitrarily large quantum memories, as long as the transmission length is suitably large.
2. We introduce the notion of k-time *program broadcast* which is a form of program encryption that allows multiple users to each learn a single evaluation of the encrypted program, while preventing any one user from learning more than k evaluations of the program. We build this primitive unconditionally and employ it to construct CCA1-secure asymmetric key encryption, encryption tokens, signatures, and signature tokens. All these schemes are information-theoretically secure against adversaries with roughly $e^{\sqrt{m}}$ quantum memory where m is the quantum memory required for the honest user.

All of the constructions additionally satisfy disappearing security, essentially preventing an adversary from storing and using a transmission later on.

2012 ACM Subject Classification Security and privacy → Information-theoretic techniques

Keywords and phrases Quantum Cryptography, Bounded Quantum Storage Model, Information-Theoretic Security

Digital Object Identifier 10.4230/LIPIcs.ITC.2025.2

Related Version *Full Version*: https://arxiv.org/abs/2302.05724

1 Introduction

Most of the interesting cryptographic concepts of security are unattainable when dealing with completely unrestricted adversaries. The conventional approach to resolve this conundrum is to focus solely on adversaries operating within polynomial time. Nevertheless, given our current understanding of complexity theory, security in this setting can only be guaranteed under the assumed hardness of solving certain problems, such as factoring, computing the discrete log, or learning with errors. As a consequence, security in the *computational model* is usually conditional.

An alternative approach is to constrain the quantum memory (qmemory) available to the adversary. This model is called the *Bounded Quantum Storage Model* (BQSM) and was first introduced in 2005 by Damgård, Fehr, Salvail, and Schaffner [9, 30]. They demonstrated that non-interactive oblivious transfer and bit-commitment can be achieved information-theoretically in this model! Surprisingly, these schemes demand no qmemory from honest participants and can be made secure against adversaries with arbitrarily large qmemories by sufficiently extending the transmission length. Subsequently, this oblivious transfer scheme was later adapted to the more useful variant of non-interactive 1-2 oblivious transfer[1] [10]. Henceforth, we denote this 1-2 oblivious transfer scheme as *BQS-OT*.

Concretely, in the BQSM, an adversary $\mathcal{A}_{\mathtt{s}}$ has access to unlimited resources at all times except at certain points. At these points, we say *the memory bound applies*, and the adversary is forced to reduce its stored state to $\mathtt{s}$-qubits. We emphasize that the adversary is again unrestricted with respect to its qmemory after the bound applies and is never restricted with respect to its computational power or classical memory.

In practice, the point when the memory bound applies can be implemented by pausing and delaying further transmissions which enforces the bound given the technological difficulties of maintaining a quantum state. Even in the foreseeable future, qmemory is not only expected to be very expensive but also very limited in size to allow for good enough fidelity. Consequently, this model appears to provide an apt characterization of real-world adversaries.

1.1 Our Results

In this paper, we delve deeper into the BQSM. We first adapt conventional definitions, such as those pertaining to encryption and one-time programs, to the BQSM. We then proceed to construct a variety of cryptographic primitives as elaborated on below.

We first build information-theoretic secure one-time programs [2] through utilizing the BQS-OT scheme. We then leverage one-time programs to construct information-theoretic secure CCA1-symmetric encryption. All these schemes are secure against any computationally unbounded adversary with $\mathtt{s}$ qmemory where $\mathtt{s}$ can be any fixed polynomial (in the security parameter) whereas honest users do not need any qmemory.

Next, we address the challenges associated with information-theoretic *asymmetric* key cryptography, particularly the task of hiding the secret key from the public key without relying on computational assumptions. We propose the novel primitive termed program broadcast as a natural solution.

Roughly speaking, a k-time program broadcast of a function P allows arbitrarily many users to each (1) obtain a single evaluation of the program, and, at the same time, (2) cannot be used by any user to learn more than k evaluations of the program. We proceed to construct an information-theoretically secure program broadcast and employ it to build CCA1-secure asymmetric key encryption. In the full paper, we also use program broadcast to build information-theoretically secure signatures, signature tokens, and encryption tokens. A signature token, as first defined in [5], can be used to sign a single message (without the signing key) and then self-destructs. We similarly define decryption tokens – each such token allows its holder to decrypt a single ciphertext and then self-destructs.

The schemes built from program broadcast are secure against any computationally unbounded adversary with $\mathtt{s}$ qmemory where $\mathtt{s}$ can be any fixed polynomial in the security parameter but require $\lg^k(\lambda)$ qmemory for the receiver where $k \in \mathbb{R}$ can take any value larger than 2. This implies that the gap between the qmemory required of the honest user and the needed qmemory to break security approaches $\lg^2(\lambda)$ $\mathsf{vs.}$ $\mathsf{poly}(\lambda)$ which translates to a gap of m $\mathsf{vs.}$ $e^{\sqrt{m}}$ by setting $m = \lg^2(\lambda)$. While we cannot assert that the required qmemory for

[1] 1-2 *oblivious transfer* is a primitive allowing a sender to transmit two strings so that (1) a receiver can choose to receive anyone of the two strings without learning anything about the other one and (2) the sender is oblivious of the receiver's choice.

[2] A one-time program (as introduced in [15]) (1) can be used to obtain a single evaluation of the program on any input chosen by the user, and at the same time, (2) (black-box obfuscation) cannot be used to obtain any other information about the program (see Definition 11).

the honest receiver in the asymmetric setting is optimal, we show in the full paper that it is not possible to achieve asymmetric key cryptography without requiring any qmemory for the honest user. Given that our qmemory requirements are already quite minimal, there is little room for improvement. For simplicity, in the rest of the paper we work with $k = 3$, however, our results easily apply to any value $k > 2$.

Prior to this work, none of the aforementioned primitives have been achieved in the BQSM and all are impossible information-theoretically in the plain model.

Our constructions additionally satisfy disappearing security which shall be formally defined in Sec. 5 but we provide a brief overview here. A transmission, say a ciphertext or signature, is deemed disappearing if it can no longer be used after a certain point [21]. In the encryption setting, an adversary cannot decrypt a ciphertext even if the private key is revealed afterward. While in the authentication setting, an adversary cannot forge a signature of a message μ even if it has received a signature of μ earlier! Such disappearing properties are impossible in the plain model for obvious reasons, but in the BQSM an adversary cannot necessarily store a ciphertext or signature for later use.

1.2 Notes on Feasibility

The qmemory requirements for the receiver in our program broadcast, asymmetric key encryption, and signature schemes make these constructions difficult to actualize with current technology given the difficulty of storing quantum states. It is worth noting, however, that the users only need a small amount of qmemory and only need it at the start in order to process the public keys. Therefore, we hope that these schemes will become feasible with further advancements.

Meanwhile, our one-time programs, symmetric key encryption, and message authentication codes can be realized with current technology as they only require the ability to prepare, send, and measure a single qubit in the BB84 basis. Specifically, these constructions can be implemented on hardware designed to run quantum key distribution (QKD), albeit with a caveat. Indeed, popular *weak coherent pulse sources* require interaction as the receiver needs to tell the sender which pulses were received. However, other technologies can enable non-interactive quantum transmissions to allow for non-interactive primitives such as one-time programs, encryption, and signature schemes. For instance, *heralded photon sources* can, in principle, remove the need for interaction.

Note also that our constructions can be modified to allow for users with imperfect apparatus to perform non-interactive error correction without compromising the security of the scheme as we briefly discuss in Sec. 9. With all that being said, in this work, we focus on building the theory and leave it as an open research direction to properly prepare the theoretical constructions for real-world applications.

1.3 Related Work

We compare our results with similar contributions in alternative models.

1.3.1 Plain Model

One-time programs in the plain model are impossible for the simple reason that any software can be stored and then reused to obtain multiple evaluations. Broadbent, Gutoski, and Stebila [7] extended this idea to the quantum realm and showed that quantum one-time programs are also impossible. These results do not apply to the BQSM as one-time programs are indeed feasible.

That being said, the works [15, 18] constructed one-time programs by leveraging one-time memory devices. These hardware devices essentially implement 1-2 oblivious transfer – each device holds two secrets and allows the device holder to choose and learn one and only one secret. Essentially, a circuit is encrypted and sent along with appropriate one-time memory devices. A receiver can only obtain enough keys from the devices to learn one evaluation from the encrypted circuit. Our one-time program construction involves replacing the one-time memory devices with the BQS-OT scheme, which serves the same purpose, although our proof requires novel analysis.

1.3.2 Bounded Classical Storage Model (BCSM)

Memory limitations were first considered in the classical setting. Specifically, in 1992, Maurer [26] introduced the *bounded classical storage model* (BCSM) where adversaries are restricted only with respect to the size of their classical memory. A cipher was shown to guarantee privacy unconditionally, even against adversaries having access to a larger memory than what is needed for honest players. Subsequent works [20, 11, 28, 12] have achieved information-theoretic (interactive) oblivious transfer, key agreement, and symmetric key encryption and signatures in this model. Furthermore, Guan, Wichs, and Zhandry [19] recently provided constructions for disappearing ciphertext and signature schemes based on public-key cryptography and one-way functions, respectively.

Overall, the BQSM seems to provide stronger security guarantees when compared to the BCSM for several reasons. First of all, in the BCSM, the gap between the required memory to run many of these schemes securely and the size of the memory needed to break for various primitives, including asymmetric encryption, is typically on the order of m vs. m^2, which is optimal [12]. This memory gap is a notable vulnerability as if it is feasible for honest users to store m bits, it does not seem too difficult for a determined adversary to store m^2 bits. In contrast, the memory gap of m vs. $e^{\sqrt{m}}$ for primitives in the BQSM is far more significant. Moreover, certain primitives achievable in the BQSM are completely unattainable in the BCSM. For instance, non-interactive oblivious transfer was shown to be impossible in the BCSM [11] which implies the impossibility of the stronger one-time program primitive.

1.3.3 Noisy Quantum Storage Model

In the *Noisy Quantum Storage Model*, which was first introduced in [31], the qmemory of the adversary is not limited in size but is intrinsically noisy. The BQS-OT schemes proposed in [9, 10] can, essentially as such, be shown secure against adversaries with noisy qmemories. In this work, we only study the BQSM and an interesting research direction is to generalize our results to the Noisy Quantum Storage Model.

2 Technical Overview

We now discuss each of our results in more detail. We first describe how to construct one-time programs and then, how to use them to build symmetric key encryption. Next, we explain why we need a new tool to tackle asymmetric cryptography. Correspondingly, we introduce the notion of program broadcast and describe how to build it in the BQSM. We then show how program broadcast can be used to build asymmetric encryption. Finally, we discuss two impossibility results related to the qmemory requirements and disappearing properties of the schemes.

2.1 One-Time Programs

We now present a quantum algorithm $\mathcal{O}$ that converts any polynomial classical circuit to a one-time program in the BQSM (Theorem 17). Fortunately, the BQSM dodges all the impossibility proofs of one-time programs. The work showing the impossibility of quantum one-time programs [7] does not apply to our setting since our programs cannot be stored by the adversary to be reused. Moreover, the proof of the impossibility of quantum obfuscation [7] does not apply since it requires the adversary to run the quantum circuit homomorphically. Adversaries in our setting are forced to continuously perform measurements on quantum states (due to the memory bound) which interrupts a quantum homomorphic evaluation.

Before describing the construction, we discuss its intuition. A natural first attempt is to apply previous one-time program constructions [15, 18] based on one-time memory devices but replace the devices with BQS-OT transmissions. The problem is that simulating an adversary requires determining which 1-out-of-2 strings are received from the oblivious transfer transmissions. This information allows the simulator to deduce which evaluation the adversary has learned from the program. In earlier constructions, this is not a problem since the adversary must *explicitly* ask which string it wishes to receive from the one-time memory devices. Note that Kilian [23] builds a similar notion known as *oblivious circuit evaluation* from OT but the simulator is also assumed to be given this information. In these works, the simulator can simply run the adversary and "know" which strings are requested. However, in our case, it is not clear this extraction is always possible – a complex adversary might not know itself which 1-out-of-2 strings it has received! To make matters worse, our adversary is a quantum and computationally unbounded algorithm. Note that the proof of security for BQS-OT does not provide a method to extract the receiver's chosen bit but rather just shows that such a bit must exist.

To solve this issue, we construct a simulator that (inefficiently) analyzes the circuit of the adversary gate-by-gate to extract the chosen bit. In particular, the simulator takes the circuit of the adversary and runs it an exponential number of times on all possible BQS-OT transmissions. This allows the simulator to approximately determine the distribution of the measurement results obtained by the adversary given the transmission provided to the adversary. This then allows the simulator to determine the distribution of the BQS-OT transmission from the perspective of the adversary given the measurement results obtained. In other words, the simulator can infer which part of the transmission the adversary is uncertain about which can be used to determine which string the adversary is oblivious to.

The rest of the construction involves adapting Kilian's approach [23] for building one-time programs from oblivious transfer to the quantum realm. Readers are referred to Sec. 6 for the formal statement of our result. However, the construction and proof are given in the full paper as they are quite long and detailed.

2.2 Symmetric Key Encryption

We use one-time programs to build a symmetric key encryption scheme that satisfies indistinguishability under (lunchtime) chosen ciphertext attack (IND-CCA1) [8] and with disappearing ciphertexts. IND-CCA1 tests security against an adversary that can query the encryption oracle at any point in the experiment and can query the decryption oracle at any point before the challenge ciphertext is received. To satisfy *disappearing* security under CCA1, we require that such an adversary cannot win the IND-CCA1 experiment even if the private key is revealed after the challenge ciphertext is given.

To achieve IND-CPA (where the adversary is not given access to a decryption oracle), the ciphertext can simply consist of a one-time program that evaluates to $\perp$ on every input except on the private key, where it outputs the message. By the security of one-time programs, an adversary has a negligible chance of guessing the key and learning the message. Upgrading to CCA1-security turns out to be far more involved as we now discuss.

First of all, it is trivial to show that we must rely on quantum ciphertexts for unconditional security in the BQSM. Now, traditionally, a CPA-secure scheme can be upgraded to a CCA-secure one by simply authenticating ciphertexts using message-authentication codes or a signature scheme. However, it seems difficult to authenticate a quantum ciphertext in a way that allows verification without any qmemory. Alternatively, we could authenticate the output of the one-time programs in the CPA construction. Specifically, instead of sending programs that map the secret key to the message, we send programs that map the key to the message along with a tag or signature of the message.

This solution fails due to the following attack: an adversary queries the encryption oracle on an arbitrary message μ, receives a one-time program, modifies the one-time program so that it outputs $\perp$ on any input starting with 0, and forwards the modified one-time program to the decryption oracle. Depending on the response received from the oracle, the adversary can successfully determine the first bit of the secret key. This attack can be repeated a polynomial number of times to deduce the entire secret key. It turns out an adversary can perform a variety of attacks of this sort due to a fundamental problem with the BQSM: this model provides security by ensuring that users only receive partial information from a transmission. Hence, it is difficult to detect whether an adversary has tampered with a transmission.

To thwart such attacks, we rely on a construction where a ciphertext consists of two interdependent one-time programs. The first program initiates some randomness and this randomness is used to ensure that the second one-time program is evaluated on a seemingly random input during decryption. To prevent the adversary from choosing the first one-time program, we authenticate the output of the first program in the second program using the secret key. We show that it is difficult to modify both programs simultaneously in a meaningful way without being detected. Proving this rigorously is somewhat technical as the adversary can still perform various modifications successfully – the reader is referred to Theorem 22 for the formal proof.

2.3 The Obstacle to Asymmetric Cryptography

Asymmetric cryptography utilizes a pair of related keys, a secret and a public key, to enable versatile cryptographic applications. Specifically, our goal is to build information-theoretic secure asymmetric key encryption and digital signatures in the BQSM. However, the task of concealing the secret key from the public key without relying on computational assumptions poses a significant challenge. Such assumptions should be avoided in the BQSM given the goal of this model is to base security purely on the memory limitations of the players.

This implies we need to impose the memory bound in some way during the public key transmission in order to hide the secret key. In the classical bounded storage model, this can be done by announcing a large classical public key [12]. However, in our scenario, imposing the memory bound necessitates the use of quantum public keys. Unfortunately, due to no-cloning, we need to create and distribute multiple copies of the quantum key – this is the general approach taken in the computational setting as well [17, 13, 22]. Here we face a critical problem: if a computationally unbounded adversary gains access to multiple copies of the key, it can repeatedly reuse its quantum memory to process multiple keys. Such an adversary could gradually extract classical information from each key copy until it has learned the entire quantum public key.

To prevent this attack, it is imperative that every public key copy needs *additional* qmemory to process, preventing an adversary from processing an unlimited number of keys. In other words, we need to distribute keys in a way so that all users must process the keys simultaneously or in parallel. More abstractly, we want to distribute quantum information in a way that allows every user to learn some information while preventing a computationally unbounded adversary from learning too much information. This goal is a recurring theme in various settings, hence, we introduce a new notion which we term *program broadcast* that formalizes and captures this objective. We build this primitive unconditionally in the BQSM and employ it to build asymmetric cryptography.

All in all, this is the first work to achieve information-theoretic asymmetric key encryption in any of the bounded (classical or quantum) storage models. This is the main focus and contribution of this paper. In the following, we first discuss program broadcast, which may be of independent interest, and then how this can be used to realize asymmetric cryptography.

2.4 Program Broadcast

In many situations, we would like to send multiple one-time programs to multiple users while ensuring that no adversary can take advantage of all the information to learn the program. This is the goal of program broadcast. Recall that a k-time program broadcast of a function P allows an unbounded polynomial number of users to each (1) obtain at least a single evaluation of the program, and, at the same time, (2) cannot be used by any user to learn more than k evaluations of the program. Essentially, an adversary with access to the broadcast can be simulated with access to k queries to an oracle for P.

In Sec. 7, we present a scheme for an information-theoretically secure program broadcast in the BQSM. The idea is to distribute a polynomial number of augmented one-time programs of P in a fixed time period. Unlike the one-time programs discussed in the previous section, these augmented versions require a small amount of qmemory to process.

More rigorously, there is a set time where the sender distributes one-time programs of the form $\mathcal{O}(P \oplus c_i)$ where $\mathcal{O}$ is our one-time compiler. In each one-time program, the output of P is padded with a different value c_i. A quantum ciphertext is generated which encrypts the value c_i and is sent with the corresponding one-time program. Users can evaluate the one-time program but need to store the ciphertext states until the encryption key is revealed to make use of their evaluation. The key is revealed at a set time after all users have received a one-time program copy. By revealing this classical key at the end, we are essentially forcing users to process the programs in a parallel fashion, limiting the number of evaluations that can be learned. In other words, an adversary with bounded qmemory can store the ciphertext states for a limited number of one-time programs and thus can only learn a limited number of evaluations of P.

To prove this rigorously, we need a new min-entropy lower bound (Lemma 7) which roughly states that if $H_\infty(X_0X_1...X_{p-1}|Q) \geq \alpha$, where $X_i \in \{0,1\}^n$ and Q is some quantum information, then there exists $i \in [p]$ and negligible ϵ such that roughly $H_\infty^\epsilon(X_i) \geq \alpha/p$. Konig and Renner [24] established such a bound but only in the case when n is sufficiently large with respect to p. Unfortunately, in our applications, n is very small with respect to p. Informally, we resolve this issue by using their bound when p is small and then use this as a base case to an inductive argument for larger values of p achieving the required result. This result is of independent interest in the field of information theory.

Note that while one-time programs can be built from one-time memory devices in the plain model, this is insufficient to build program broadcast in the plain model. Specifically, we need to use the qmemory bound outside the one-time programs as well in order to construct this primitive. Hence, program broadcast acts as a more pronounced advantage of the BQSM.

2.5 Asymmetric Key Encryption

We now discuss how we upgrade our symmetric key encryption scheme to a CCA1-secure asymmetric key scheme using program broadcast. We let the private key be a large program P while the public key is a quantum program broadcast of P. A user can use the broadcast to learn a single evaluation of P which can be used to encrypt messages in the same way as in the private key setting. However, an adversary can only learn a limited number of evaluations by the security of the program broadcast. If P is large enough then this knowledge is not sufficient to predict the output of P on a random input. In other words, the adversary cannot predict the encryption key of another user, so security is reduced to the symmetric key setting.

It is not difficult to show that the public keys need to be mixed-state for information-theoretic security in the BQSM. This might seem unfortunate, but we believe that it is not very relevant whether keys are pure or mixed-state. Some works, such as [4], require that the quantum public keys be pure since this would provide some level of certification by allowing users to compare keys with a SWAP test. However, this test is not always feasible to perform in the BQSM given that keys cannot necessarily be stored. Instead, we provide a more secure way to certify mixed-state quantum keys without establishing authenticated quantum channels in a companion work [2]. That being said, in this work, we do not certify public keys and it is always assumed that honest users receive authentic quantum public keys. This is a common assumption for the quantum public key schemes, such as those in [17, 13, 22], as a quantum state cannot be signed [3].

2.6 Impossibility Results

We briefly discuss two interesting impossibility results that we present in the full paper. First, we show that it is impossible to implement asymmetric schemes (with unbounded public key copies) in the BQSM without requiring qmemory from the honest user. The fundamental idea is that if the public keys require no qmemory to process then an adversary can request and process an unbounded number of key copies which would reveal information about the secret key. The technical argument is more involved since the public keys may be mixed-state which could aid in hiding information.

The second impossibility result is concerned with the disappearing property of our constructions. The ciphertexts and one-time programs satisfy disappearing security providing interesting applications as we discussed earlier. However, this disappearing property can also be disadvantageous in some scenarios. For instance, sometimes, it is more beneficial to have an obfuscation that can be stored and reused multiple times instead of a disappearing one-time program. The final contribution of this paper is to provide a negative result showing that non-disappearing obfuscation and one-time programs are impossible in the BQSM.

The proof relies on a new approach since our simulator is computationally unbounded, which nullifies adversarial attacks used in standard impossibility proofs [1, 14, 6, 7]. Essentially, we show that if a computationally unbounded adversary can perform a single evaluation after the qmemory bound applies then it can rewind the program and perform another evaluation. By the gentle measurement lemma [32], this rewinding introduces only a negligible error each time which allows the adversary to learn a super-polynomial number of evaluations with non-negligible probability. These evaluations can be used to learn more information regarding the hidden program than any simulator can learn using only a polynomial number of oracle queries. We refer the reader to the full paper for more details.

3 Preliminaries

3.1 Notation

In the following, we denote Hilbert spaces by calligraphic letters: $\mathcal{H}$, $\mathcal{V}$, etc... The set of positive semi-definite operators in Hilbert space $\mathcal{H}$ are denoted $\text{Pos}(\mathcal{H})$. When Hilbert space $\mathcal{X}$ holds a classical value, we denote the random variable for this value by X.

We denote the density matrix of a quantum state in a register E as ρ_E. If a quantum state depends on a classical variable X, then we denote ρ_E^x as the density matrix of the state given $X = x$. Meanwhile, to an observer who does not know the value of X, the state is given by $\rho_E := \sum_x P(X = x)\rho_E^x$ and the joint state by $\rho_{XE} := \sum_x P(X = x) |x\rangle \langle x| \otimes \rho_E^x$. Such states having a quantum part depending on a classical value are called *cq-states*. The random variable X then corresponds to the classical state $\rho_X := \sum_x P(X = x) |x\rangle \langle x|$.

We denote the trace distance between two density matrices as $\delta(\rho, \sigma)$ and the trace norm as $\|\rho\|_1$.

Notice that $\rho_{XE} = \rho_X \otimes \rho_E$ if and only if ρ_E is independent of X (i.e. $\rho_E^x = \rho_E$) which means that no information can be learned regarding ρ_E from X and vice versa. More generally, if ρ_{XE} is ϵ-close to $\rho_E \otimes \rho_X$ in terms of trace distance denoted as $\delta(\rho_{XE}, \rho_X \otimes \rho_E) \leq \epsilon$ or equivalently $\rho_{XE} \approx_\epsilon \rho_X \otimes \rho_E$, then no observer can distinguish between the two systems with advantage greater than ϵ. We write $\langle \rho_A, \rho_B \rangle$ to denote a sequential transmission where register A is sent and then register B.

The rectilinear or $+$ basis for the complex space $\mathbb{C}^2$ is the pair $\{|0\rangle, |1\rangle\}$ while the diagonal or $\times$ basis is the pair $\{|0\rangle_\times, |1\rangle_\times\}$ where $|0\rangle_\times := \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|1\rangle_\times := \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. To choose between the $+$ and $\times$ basis depending on a bit $b \in \{0, 1\}$, we write $\{+, \times\}_b$.

We say $x \in_R X$ if x is chosen uniformly at random from the values in X. We let $\mathbb{1}_k$ denote the density matrix of a uniformly distributed variable on k elements. Also, let $[n] := [0, 1, \ldots, n - 1]$ and let $\mathsf{negl}(n)$ be denoting any function that is smaller than the inverse of any polynomial for large enough n.

We say an algorithm A is *QPT* if it is a quantum polynomial-time algorithm. We let A^P denote an algorithm with access to a polynomial number of classical oracle queries to the program P and let A^{qP} denote access to q classical oracle queries.

Recall a Pauli pad [27] information-theoretically hides a n-qubit state using a $2n$-bit key k. We denote $\mathsf{PP}_k(\rho)$ to be the Pauli pad of the state with density matrix ρ using key k.

By abuse of notation, if x is a binary string and M is a matrix then, for simplification, we let $M \cdot x$ denote the string representation of the vector $M \cdot \vec{x}$.

3.2 Rényi Entropy

We remind the reader of the notion of quantum Rényi Entropy and its variants. See [25] for more detailed definitions.

▶ **Definition 1** (Conditional Min-Entropy). *Let $\rho_{XB} \in \text{Pos}(\mathcal{H}_X \otimes \mathcal{H}_B)$ be classical on $\mathcal{H}_X$. The min-entropy of ρ_{XB} given $\mathcal{H}_B$ is defined as*

$$H_\infty(X|B)_\rho := -\lg\left(p_{\text{guess}}(X|B)_\rho\right)$$

where $p_{\text{guess}}(X|B)_\rho$ is the maximal probability to decode X from B with a POVM on $\mathcal{H}_B$.

▶ **Definition 2** (Smooth Min-Entropy). *Let $\epsilon \geq 0$ and $\rho_{XB} \in \mathrm{Pos}(\mathcal{H}_X \otimes \mathcal{H}_B)$. The ϵ-smooth min-entropy of ρ_{XB} given $\mathcal{H}_B$ is defined as*

$$H^{\epsilon}_{\infty}(X|B)_{\rho} \coloneqq \sup_{\overline{\rho}} H_{\infty}(X|B)_{\overline{\rho}}$$

where the supremum is taken over all density operators $\overline{\rho}_{XB}$ acting on $\mathcal{H}_X \otimes \mathcal{H}_B$ such that $\delta(\overline{\rho}_{XB}, \rho_{XB}) \leq \epsilon$.

3.3 Uncertainty Relations

This section discusses lower bounds on the average min-entropy and presents a new generalized min-entropy splitting lemma conditioned on quantum states. This will be fundamental in our program broadcast construction.

The min-entropy satisfies the following chain rule.

▶ **Lemma 3** (Chain Rule for Min-Entropy [29]). *Let $\epsilon \geq 0$ and $\rho_{XUE} \in \mathrm{Pos}(\mathcal{H}_X \otimes \mathcal{H}_U \otimes \mathcal{H}_E)$ where register E has size m. Then,*

$$H^{\epsilon}_{\infty}(X|UE)_{\rho} \geq H^{\epsilon}_{\infty}(XE|U)_{\rho} - m.$$

▶ **Lemma 4** (Uncertainty Relation [9]). *Let $\rho \in \mathcal{P}(H_2^{\otimes n})$ be an arbitrary n-qubit quantum state. Let $\Theta \in_R \{+, \times\}^n$ and let X be the outcome when ρ is measured in basis Θ. Then for any $0 < \gamma < \frac{1}{2}$,*

$$H^{\epsilon}_{\infty}(X|\Theta)_{\rho} \geq \left(\frac{1}{2} - 2\gamma\right) n,$$

where $\epsilon = \exp(-\frac{\gamma^2 n}{32(2-\lg(\gamma))^2})$.

If $X = X_0X_1$ has high entropy, then it was shown in [10] that, in a randomized sense, one of X_0 or X_1 has high entropy.

▶ **Lemma 5** (Conditional Min-Entropy Splitting Lemma [10]). *Let $\epsilon \geq 0$ and let X_0, X_1 and Z be random variables. If $H^{\epsilon}_{\infty}(X_0X_1|Z) \geq \alpha$, then there exists a binary random variable C such that $H^{\epsilon+\epsilon'}_{\infty}(X_{1-C}|ZC) \geq \alpha/2 - 1 - \log(1/\epsilon')$ for any $\epsilon' > 0$.*

More generally, min-entropy sampling gives lower bounds on the min-entropy of a subset of a string given the min-entropy of the entire string. In [24], it was shown how to sample min-entropy relative to quantum knowledge which allows us to give a lower bound on min-entropy conditioned on a quantum state.

▶ **Lemma 6** (Quantum Conditional Min-Entropy Splitting Lemma [24]). *Let $\epsilon \geq 0$ and let $\rho_{X_0X_1B}$ be a quantum ccq-state where X_0 and X_1 are classical random variables on alphabet $\mathcal{X}$ of dimension $d = \lg|\mathcal{X}| > 14$. Then, there exists a binary random variable C such that for all $\tau \geq 0$,*

$$H^{\epsilon+\tau}_{\infty}(X_C|CB)_{\rho} \geq \frac{H^{\tau}_{\infty}(X_0X_1|B)_{\rho}}{2} - 2$$

where $\epsilon \coloneqq 2^{-\frac{3d}{2}+1}$.

Proof. Notice that any distribution P_C over $\{0, 1\}$ is a $(2, 3/4, 0)$-sampler (Definition 2.1 [24]). Corollary 6.19 in [24] gives the result. ◀

In Lemma 6 the size of the sampled string X_C is half the size of the original string X. The results in [24] do not give strong lower bounds when the sampled string is small relative to the size of the original string. Hence, we present the following generalization of Lemma 6 which will be used in constructing program broadcast and asymmetric key encryption.

▶ **Lemma 7** (Generalized Min-Entropy Splitting Lemma). *Let $\epsilon \geq 0$. Let ρ_{XB} be a quantum cq-state where $X := X_0X_1...X_\ell$ and each X_i is a classical random variable over alphabet $\mathcal{X}$ of dimension $d > 14$. There exists a random variable C (with $\lceil \lg \ell \rceil$ bits) such that for all $\tau \geq 0$,*

$$H_\infty^{\epsilon+\tau}(X_C|CB)_\rho \geq \frac{H_\infty^\tau(X|B)_\rho}{\ell} - 4$$

where $\epsilon := 2^{-\frac{3d}{2}+2}$.

Proof. We only consider the case $\ell = 2^k$, however, the same argument works for the general case. We prove the following slightly stronger bound using induction on k:

$$H_\infty^{\epsilon_k+\tau}(X_C|CB) \geq \frac{H_\infty^\tau(X|B)}{2^k} - 4(1 - \frac{1}{2^k}).$$

where $\epsilon_k := \sum_{i=0}^{k-1} 2^{-2^i d+1}$. The base case $k = 1$ follows trivially from the Quantum Conditional Min-Entropy Splitting Lemma 6. Assume the claim holds for $k = n$. For $k = n+1$, we apply Lemma 6 on the two variables $Y_0 := X_1...X_{2^n}$ and $Y_1 := X_{2^n+1}...X_{2^{n+1}}$ to deduce that there exists a binary variable C_1 such that $H_\infty^{\epsilon'+\tau}(Y_{C_1}|BC_1) \geq \frac{H_\infty^\tau(X|B)}{2} - 2$ where $\epsilon' := 2^{-2^n d+1}$. By the inductive hypothesis, there exists C' such that:

$$H_\infty^{\epsilon_n+\epsilon'+\tau}(X_{C_1C'}|BC_1C') \geq \frac{H_\infty^{\epsilon'+\tau}(Y_{C_1}|BC_1)}{2^n} - 4\left(1 - \frac{1}{2^n}\right) \geq$$
$$\frac{\frac{H_\infty^\tau(X|B)}{2} - 2}{2^n} - 4\left(1 - \frac{1}{2^n}\right) = \frac{H_\infty^\tau(X|B)}{2^{n+1}} - 4\left(1 - \frac{1}{2^{n+1}}\right).$$ ◀

3.4 Privacy Amplification

For the rest of this work, we let $\mathsf{H}_{m,\ell}$ denote a two-universal class of hash functions from $\{0,1\}^m$ to $\{0,1\}^\ell$. Remember that a class of hash functions is *two-universal* if for every distinct $x, x' \in \{0,1\}^m$ and for $F \in_R \mathsf{H}_{m,\ell}$, we have $Pr[F(x) = F(x')] \leq \frac{1}{2^\ell}$.

▶ **Theorem 8** (Privacy Amplification [29]). *Let $\epsilon \geq 0$. Let $\rho_{XB} \in \text{Pos}(\mathcal{H}_X \otimes \mathcal{H}_B)$ be a cq-state where X is classical and takes values in $\{0,1\}^m$. Let $F \in_R \mathsf{H}_{m,\ell}$ be the random variable for a function chosen uniformly at random in a two-universal class of hash functions $\mathsf{H}_{m,\ell}$. Then,*

$$\delta(\rho_{F(X)FB}, \mathbb{1} \otimes \rho_{FB}) \leq \frac{1}{2} 2^{-\frac{1}{2}(H_\infty^\epsilon(X|B)_\rho - \ell)} + \epsilon.$$

3.5 Algebra

The following result shows that it is difficult to learn a large matrix using a small number of samples and is proven in the full paper.

▶ **Theorem 9.** *Let M be an arbitrary $\ell \times n$ binary matrix. Let A be an algorithm that is given as input: $(a_1, b_1), ..., (a_m, b_m)$ and $(\hat{a}_1, \hat{b}_1), ..., (\hat{a}_p, \hat{b}_p)$ where $a_i, \hat{a}_i \in \{0,1\}^n$, $b_i, \hat{b}_i \in \{0,1\}^\ell$, $m < n$, p is a polynomial in ℓ, $b_i = M \cdot a_i$ and $\hat{b}_i \neq M \cdot \hat{a}_i$. Then the following statements hold:*

1. *For any vector $a \in \{0,1\}^n$ not in the span of $(a_1, ..., a_m)$, if A outputs a guess b' of $b := M \cdot a$, then* $\Pr[b' = b] = O(2^{-\ell})$.
2. *Let $x_0, x_1 \in \{0,1\}^n$ be any two distinct vectors not in the span of $(a_1, ..., a_m)$. Choose $r \in_R \{0,1\}$. If A is additionally given x_0, x_1, y_r where $y_r := M \cdot x_r$ and outputs a guess r' then* $\Pr[r' = r] \leq \frac{1}{2} + O(2^{-\ell})$.

4 Definitions: Obfuscation and Variants in the BQSM

In this section, we adapt the notions of obfuscation and one-time programs to the BQSM and introduce a related notion termed program broadcast.

▶ **Definition 10** (BQS Obfuscation). *A algorithm O is a (r, s)-*BQS obfuscator *of the class of classical circuits $\mathcal{F}$ if it QPT and satisfies the following:*
1. *(functionality) For any circuit $C \in \mathcal{F}$, the circuit described by $O(C)$ can be used to compute C on an input x chosen by the evaluator.*
2. *For any circuit $C \in \mathcal{F}$, the receiver requires r qmemory to learn an evaluation of C using $O(C)$.*
3. *(security) For any computationally unbounded adversary $\mathcal{A}_s$ there exists a computationally unbounded simulator $\mathcal{S}_s$ such that for any circuit $C \in \mathcal{F}$,*

$$|\Pr[\mathcal{A}_s(O(C)) = 1] - \Pr[\mathcal{S}_s^C(|0\rangle^{\otimes|C|}) = 1]| \leq \mathsf{negl}(|C|).$$

One-time programs, introduced in [7], are similar to obfuscation but can only be used to learn a single evaluation of the program. We adapt this notion to the BQSM.

▶ **Definition 11** (BQS One-Time Program). *An algorithm O is a (r, s)-*BQS one-time compiler *for the class of classical circuits $\mathcal{F}$ if it is QPT, satisfies the first three conditions of Definition 10, and the following:*

4. *(security) For any computationally unbounded adversary $\mathcal{A}_s$ there exists a computationally unbounded simulator $\mathcal{S}_s$ such that for any circuit $C \in \mathcal{F}$*

$$|\Pr[\mathcal{A}_s(O(C)) = 1] - \Pr[\mathcal{S}_s^{1C}(|0\rangle^{\otimes|C|}) = 1]| \leq \mathsf{negl}(|C|).$$

We introduce the notion of BQS program broadcast which is similar to one-time programs but additionally requires that multiple copies of the encrypted program can only be used to learn a limited number of evaluations. While one-time programs allow for symmetric key cryptography, program broadcast allows for powerful applications such as asymmetric cryptography and tokens.

▶ **Definition 12** (BQS Program Broadcast). *A (q, s, k)-*BQS program broadcast *for the class of circuits $\mathcal{C}$ consists of the following QPT algorithms:*
1. $\mathsf{KeyGen}(1^\lambda, t_{\mathrm{end}})$ *: Outputs a classical key ek.*
2. $\mathsf{Br}(\mathsf{s}, ek, C)$ *: Outputs a quantum transmission O_C for the circuit $C \in \mathcal{C}$ during broadcast time (before t_{end}). Outputs ek after broadcast time.*
3. $\mathsf{Eval}(\langle O_C, ek\rangle, x)$ *: Outputs an evaluation y on input x from the transmission $\langle O_C, ek\rangle$ using q qmemory.*

Correctness requires that for any circuit $C \in \mathcal{C}$ and input x,

$$\Pr\left[\mathsf{Eval}(\langle O_C, ek\rangle, x) = C(x) \;\middle|\; \begin{array}{rl} ek & \leftarrow \mathsf{KeyGen}(1^\lambda, t_{\mathrm{end}}) \\ O_C & \leftarrow \mathsf{Br}(\mathsf{s}, ek, C) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security requires that for any computationally unbounded adversary $\mathcal{A}_s$ there exists a computationally unbounded simulator $\mathcal{S}_s$ such that for any circuit $C \in \mathcal{C}$, and $ek \leftarrow \mathsf{KeyGen}(1^\lambda, t_{\mathrm{end}})$,

$$|Pr[\mathcal{A}_s^{\mathsf{Br}(\mathsf{s},ek,C)}(|0\rangle) = 1] - Pr[\mathcal{S}_s^{kC}(|0\rangle^{\otimes|C|}) = 1]| \leq \mathsf{negl}(\lambda).$$

5 Definitions: Disappearing Cryptography

In this section, we initiate the study of disappearing cryptography in the BQSM. These concepts were defined earlier in [21, 16] but we adapt the definitions to the BQSM. We define these notions for oblivious transfer, one-time programs, and asymmetric key encryption.

Informally, a state $|\phi\rangle$ is *disappearing* if any adversary that receives $|\phi\rangle$ cannot produce a state with the same "functionality" as $|\phi\rangle$ after a certain point in the transmission. In the BQSM, this point is when the adversary's qmemory bound applies.

5.1 One-Time Programs

We present the following experiment to define disappearing security for one-time programs. Intuitively, the experiment checks whether an adversary that receives a one-time program can retrieve the evaluation on a random input x that is only revealed after the program. The experiment focuses on matrix branching programs but the same experiment can be applied to any circuit class where sampling can be done efficiently.

Experiment $\mathsf{1TP}^{\mathsf{Dis}}_{\Pi,\mathcal{A}}(m,n,\ell)$: Let $\Pi(m,P)$ be the one-time program protocol with security parameter m on the matrix branching program $P:\{0,1\}^n\rightarrow\{0,1\}^\ell$.

1. Sample $x\in_R\{0,1\}^n$ and a matrix branching program $P:\{0,1\}^n\rightarrow\{0,1\}^\ell$ uniformly at random.
2. Protocol $\Pi(m,P)$ is executed between the experiment and adversary $\mathcal{A}$.
3. After the execution ends, send x to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a guess p for $P(x)$.
5. The output of the experiment is 1 if $p=P(x)$ and 0 otherwise.

▶ **Definition 13.** *A one-time program protocol Π_s is disappearing in the BQSM if for any adversary $\mathcal{A}_s$,*

$$\Pr\left[\mathsf{1TP}^{\mathit{Dis}}_{\Pi_s,\mathcal{A}_s}(m,n,\ell)=1\right]\leq\frac{1}{2^\ell}+\mathsf{negl}(\min(n,m)).$$

5.2 Encryption

We first recall the definition of a quantum asymmetric (public) key encryption scheme on classical messages. Note that the public key in this setting is quantum so multiple copies must be created and distributed due to the no-cloning theorem. Hence, we add an algorithm $\mathsf{KeySend}$ that outputs a copy of the quantum public key when queried. In our security experiment, the adversary is allowed to receive a polynomial number of public key copies. We also introduce an algorithm $\mathsf{KeyReceive}$ which describes how to extract a reusable classical key from a public key copy to use for encryption.

▶ **Definition 14** (Quantum Asymmetric Key Encryption)**.** *A quantum asymmetric key encryption scheme Π over classical message space $\mathcal{M}$ consists of the following QPT algorithms:*

- $\mathsf{Gen}(1^\lambda)$: *Outputs a private key sk.*
- $\mathsf{KeySend}(sk)$: *Outputs a quantum public key copy ρ_{pk}.*
- $\mathsf{KeyReceive}(\rho_{pk})$: *Extracts a key k from ρ_{pk}.*
- $\mathsf{Enc}(k,\mu)$: *Outputs a ciphertext ρ_{ct} for $\mu\in\mathcal{M}$.*
- $\mathsf{Dec}(sk,\rho_{ct})$: *Outputs a message μ' by decrypting ρ_{ct}.*

▶ **Definition 15** (Correctness). *A quantum asymmetric key encryption scheme* Π *is* correct *if for any message* $\mu \in \mathcal{M}$*,*

$$\Pr\left[\mathit{Dec}(sk, \rho_{ct}) = \mu \;\middle|\; \begin{array}{c} sk \;\leftarrow \mathit{Gen}(1^\lambda) \\ \rho_{pk} \;\leftarrow \mathit{KeySend}(sk) \\ k \;\leftarrow \mathit{KeyReceive}(\rho_{pk}) \\ \rho_{ct} \;\leftarrow \mathit{Enc}(k, \mu) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

We now present an experiment to test disappearing security under lunchtime chosen ciphertext attack (qDCCA1) in the BQSM. Recall, IND-CCA1 [8] tests security against an adversary that can query the encryption oracle at any point in the experiment and can query the decryption oracles at any point before the challenge ciphertext is received. To satisfy *disappearing* security under CCA1, we require that an adversary cannot win the IND-CCA1 experiment even if the private key is revealed after the challenge ciphertext is given.

Experiment $\mathsf{AsyK}^{\mathsf{qDCCA1}}_{\boldsymbol{\Pi},\boldsymbol{\mathcal{A}}}(\boldsymbol{\lambda})$:

1. Sample a private key $sk \leftarrow \mathsf{Gen}(1^\lambda)$ and bit $b \in_R \{0, 1\}$.
2. Generate a public key $\rho_{pk} \leftarrow \mathsf{KeySend}(sk)$.
3. Extract key $k \leftarrow \mathsf{KeyReceive}(\rho_{pk})$.
4. Adversary outputs two messages $m_0, m_1 \leftarrow \mathcal{A}^{\mathsf{KeySend}(sk),\mathsf{Enc}(k,\cdot),\mathsf{Dec}(sk,\cdot)}$.
5. Send $\mathcal{A}^{\mathsf{KeySend}(sk),\mathsf{Enc}(k,\cdot)}$ the ciphertext $\rho_{ct} \leftarrow \mathsf{Enc}(k, m_b)$.
6. Give $\mathcal{A}^{\mathsf{KeySend}(sk),\mathsf{Enc}(k,\cdot)}$ the private key sk.
7. $\mathcal{A}^{\mathsf{KeySend}(sk),\mathsf{Enc}(k,\cdot)}$ outputs a guess b'.
8. The output of the experiment is 1 if $b' = b$ and 0 otherwise.

We can similarly construct disappearing experiment in the symmetric key case, denoted $\mathsf{SymK}^{\mathsf{qDCCA1}}_{\Pi,\mathcal{A}}(\lambda)$ by deleting steps 2 & 3 and replacing k with sk in the asymmetric experiment.

▶ **Definition 16** (Security). *An asymmetric key encryption scheme* Π_s *satisfies* disappearing indistinguishability against chosen ciphertext attacks (qDCCA1) *if for any adversary* $\mathcal{A}_s$*,*

$$\Pr\left[\mathit{AsyK}^{qDCCA1}_{\Pi_s,\mathcal{A}_s}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

6 One-Time Programs

In this section, we give the result for an information-theoretic secure one-time program in the BQSM. The construction and proof are provided in the full paper as they are long and technical. Our construction is also disappearing meaning a one-time program cannot be evaluated after the transmission ends.

▶ **Theorem 17.** *There exists an algorithm* $\mathcal{O}_s$ *that is a disappearing information-theoretically secure* $(0, \mathtt{s})$ *one-time compiler for the class of polynomial classical circuits against any computationally unbounded adversary with* $\mathtt{s}$ *qmemory bound.*

7 Program Broadcast

In this section, we construct a BQS program broadcaster as introduced in Definition 12. This will allow us to tackle asymmetric key encryption in later sections. Let $\mathcal{C}_m = \{\mathcal{C}_{n,m}\}_{n \leq 2^m}$ where $\mathcal{C}_{n,m}$ is a set of polynomial-size circuits in n, of input size n, and output size m. Note that any polynomial-size circuits belongs to such a class by adding null outputs to ensure that $n \leq 2^m$.

▶ **Construction 18** (BQS Program Broadcaster). *The $(12m, \mathtt{s}, \frac{\mathtt{s}}{2m})$-BQS program broadcast scheme for the class $\mathcal{C}_m$ until time t_{end} is as follows:*

- *$\mathsf{KeyGen}(1^\lambda, t_{\text{end}})$: Choose uniformly at random a $(12m) \times (12m)$ binary matrix M and let $F(x) := M \cdot x$ be the corresponding program. Choose a two-universal hash function $H \in_R \mathsf{H}_{12m,m}$. Output $ek := (M, H, t_{\text{end}})$.*
- *$\mathsf{Br}(\mathtt{s}, ek, P)$: Let $P \in \mathcal{C}_m$. If queried before time t_{end}, then:*
  1. *Randomly choose $\mathbf{r}, \mathbf{x} \in_R \{0,1\}^{12m}$.*
  2. *Compute $\theta := F(\mathbf{r})$ and $c := H(\mathbf{x})$.*
  3. *Send $|\mathbf{x}\rangle_\theta := |x_1\rangle_{\theta_1} \ldots |x_{12m}\rangle_{\theta_{12m}}$.*
  4. *Send $\mathcal{O}_{\mathtt{s}}(P \oplus c)$.*
  5. *Send $\mathbf{r}$.*

  The entire transmission is denoted as $O_P^{r;x}$.
  If queried after time t_{end}, then apply the qmemory bound and output ek.
- *$\mathsf{Eval}(\langle O_P^{r;x}, ek\rangle, v)$:*
  1. *Store $|\mathbf{x}\rangle_\theta$.*
  2. *Evaluate the one-time program $\mathcal{O}_{\mathtt{s}}(P \oplus c)$ on v to obtain $P(v) \oplus c$.*
  3. *After $\mathbf{r}$ and F are received, compute θ.*
  4. *Measure $|\mathbf{x}\rangle_\theta$ in the basis θ to obtain $\mathbf{x}$.*
  5. *Use H to compute $c = H(\mathbf{x})$ and obtain $P(v)$.*

▶ **Theorem 19** (Security). *Construction 18 is a $(12m, \mathtt{s}, \frac{\mathtt{s}}{2m})$-BQS program broadcaster for the class $\mathcal{C}_m$.*

Proof. Let $P \in \mathcal{C}_m$. It is clear that the receiver requires $12m$ qmemory to learn a single evaluation of P from the broadcast.

In terms of security, an adversary $\mathcal{A}_{\mathtt{s}}$ can receive a polynomial number, say p, outputs of the broadcast $\mathsf{Br}(\mathtt{s}, ek, P)$. From these states, the adversary obtains $(|\mathbf{x}_i\rangle_{\theta_i})_{i\in[p]}$ and one-time programs $(\mathcal{O}_{\mathtt{s}}(P \oplus c_i))_{i\in[p]}$ where $c_i := H(\mathbf{x}_i)$.

$\mathcal{A}_{\mathtt{s}}$ can be simulated by $\mathcal{S}_{\mathtt{s}}$ which has access to a single oracle query to $P \oplus c_i$ for each $i \in [p]$ by the security of one-time protocol (Theorem 17). The lemma below shows that $\mathcal{S}_{\mathtt{s}}$ can learn at most $\frac{\mathtt{s}}{2m}$ values in $\{c_i\}_{i\in[p]}$. This implies that $\mathcal{A}_{\mathtt{s}}$ can learn at most $\frac{\mathtt{s}}{2m}$ evaluations of P from the broadcast thus achieving program broadcast security.

▶ **Lemma 20.** *$\mathcal{S}_{\mathtt{s}}$ can distinguish at most $\frac{\mathtt{s}}{2m}$ terms in $\{c_i\}_{i\in[p]}$ from random.*

Proof. Assume that there exists $\frac{\mathtt{s}}{2m}$ values $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{\frac{\mathtt{s}}{2m}}}$ that are distinguishable from random.

Instead of the broadcaster sending $|\mathbf{x}_i\rangle_{\theta_i}$, a standard purification argument shows it is sufficient to show security for the protocol with the following modification. For each qubit supposed to be sent, the broadcaster instead prepares an EPR state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends one half to $\mathcal{S}_{\mathtt{s}}$ and keeps one half. Then the broadcaster measures its halves of the EPR pairs in a random BB84 basis in $\{+, \times\}^{12m}$ after the memory bound of $\mathcal{S}_{\mathtt{s}}$ applies. Let Θ_i and X_i be the random variables representing the broadcaster's choice of measurement basis and outcome. Let $X_I := X_{i_1}X_{i_2}\ldots X_{i_{\frac{\mathtt{s}}{2m}}}$ and $\Theta_I := \Theta_{i_1}\Theta_{i_2}\ldots\Theta_{i_{\frac{\mathtt{s}}{2m}}}$. The Uncertainty Relation (Lemma 4) implies that for $\gamma = \frac{1}{12}$, there exists ϵ (negligible in m) such that $H_\infty^\epsilon(X_I|\Theta_I) \geq (\frac{1}{2} - 2\gamma)(6\mathtt{s}) = 2\mathtt{s}$. Let B be the random $\mathtt{s}$ qubit state the adversary stores when the memory bound applies. By the chain rule for min-entropy (Lemma 3),

$$H_\infty^\epsilon(X_I|\Theta_I B) \geq H_\infty^\epsilon(X_I|\Theta_I) - \mathtt{s} \geq \mathtt{s}.$$

Hence, by the Generalized Min-entropy Splitting Lemma 7, there exists a random variable C and negligible value $\epsilon' > 0$ such that,

$$H_\infty^{\epsilon'}(X_{i_C}|\Theta_I CB) \geq 2m - 4.$$

By the Privacy Amplification Theorem 8,

$$\delta(\rho_{H(X_{i_C})CH\Theta_I B}, \mathbb{1}\otimes\rho_{CH\Theta_I B}) \leq \frac{1}{2}2^{-\frac{1}{2}(H_\infty^\epsilon(X_{i_C}|\Theta_I CB)_\rho - m)} + \epsilon = O(2^{-m}) = \mathsf{negl}(n).$$

The final equality is because $m \geq \lg^2(n)$ which means 2^{-m} is negligible in n. This gives a contradiction so there is less than $\frac{\mathsf{s}}{2m}$ values c_i which are distinguishable from random. ◀

This completes the proof of Theorem 19. ◀

8 Encryption Schemes

We construct information-theoretically secure symmetric encryption in the BQSM and, then, show how to upgrade it to the asymmetric setting.

8.1 Symmetric Key Encryption

In this section, we present a private key encryption scheme that satisfies disappearing indistinguishability under chosen ciphertext attacks (qDCCA1).

▶ **Construction 21.** *The symmetric key encryption scheme Π_{SymK} against adversaries with qmemory bound s is as follows.*

- $\mathsf{Gen}(1^\lambda)$*: Let $m := \lceil(\lg\lambda)^{3/2}\rceil$. Choose at random strings $q, w \in_R \{0,1\}^m$, a $(2m)\times(2m)$ invertible binary matrix M and string $z \in_R \{0,1\}^{2m}$. Define $S(x) := M\cdot x + z$, and $S'(x) := qx + w \mod 2^m$. The private key is $k := (q, w, M, z)$.*
- $\mathsf{Enc}(\mathsf{s}, k, \mu)$*: Let $\ell := |\mu|$. Choose strings $a, b \in_R \{0,1\}^m$ and define $f(x) := ax + b \mod 2^m$. Construct the following program:*

$$E_{k,f,\mu}(y) = \begin{cases} S'(f(x))\|\mu & \text{if } y = S(x\|f(x)) \\ \perp & \text{otherwise.}\end{cases}$$

The encryption is $\rho_{ct} \leftarrow \langle\mathcal{O}_s(f), \mathcal{O}_s(E_{k,f,\mu})\rangle$ sent in sequence; first $\mathcal{O}_s(f)$ and then $\mathcal{O}_s(E_{k,f,\mu})$.
- $\mathsf{Dec}(k, \rho_{ct})$*: Check ρ_{ct} has the correct format (see note below). Evaluate the first one-time program on a random input v to obtain $f(v)$. Evaluate the second one-time program on $S(v\|f(v))$. If the output is of the form $S'(f(v))\|\hat{\mu}$ (for some string $\hat{\mu}$) then output $\hat{\mu}$ and $\perp$ otherwise.*

▶ **Theorem 22** (Security). *Construction 21 (Π_{SymK}) satisfies qDCCA1 security against computationally unbounded adversaries with qmemory bound s.*

Proof. An adversary $\mathcal{A}_\mathsf{s}$ in the qDCCA1-security experiment requests a polynomial number, say Q_e, of encryption queries and a polynomial number, say Q_d, of decryption queries. Denote the ciphertexts produced by all encryption queries to the oracle as $\langle\mathcal{O}(f_i), \mathcal{O}(E_{k,f_i,\mu_i})\rangle_{i\in[Q_e]}$ and denote the decryption queries submitted to the oracle as $\langle\mathcal{O}(\hat{f}_j), \mathcal{O}(\hat{E}_j)\rangle_{j\in[Q_d]}$ for $\mathbf{NC}^1$ functions $\hat{f}_j : \{0,1\}^m \to \{0,1\}^m$ and $\hat{E}_j : \{0,1\}^{2m} \to \{0,1\}^{m+\ell}$. Any decryption query not of this form is immediately rejected as an invalid ciphertext.

$\mathcal{A}_{\mathsf{s}}$ cannot learn anything from the encryption queries alone as such attacks can be simulated with only a single oracle query to each program in $(E_{k,f_i,\mu_i})_{i\in[Q_e]}$. All such queries will yield $\perp$ except with negligible probability by the security of one-time programs (Theorem 17).

Now, consider what $\mathcal{A}_{\mathsf{s}}$ learns from the first decryption query $\langle \mathcal{O}(\hat{f}_0), \mathcal{O}(\hat{E}_0)\rangle$. Suppose $\mathcal{A}_{\mathsf{s}}$ requested the encryption queries $\langle \mathcal{O}(f_i), \mathcal{O}(E_{k,f_i,\mu_i})\rangle_{i\in[d_0]}$ prior to requesting the first decryption query. These ciphertexts may be utilized in the construction of the decryption query. Let A_0 be the sub-algorithm of $\mathcal{A}_{\mathsf{s}}$ which produces the first decryption query using these ciphertexts. Hence, we write $\langle \mathcal{O}_{\mathsf{s}}(\hat{f}_0), \mathcal{O}_{\mathsf{s}}(\hat{E}_0)\rangle \leftarrow \mathsf{A}_0(\langle \mathcal{O}(f_i), \mathcal{O}(E_{k,f_i,\mu_i})\rangle_{i\in[d_0]})$.

As a side note, prior to sending $\mathcal{O}_{\mathsf{s}}(\hat{E}_0)$, the memory bound for $(\mathcal{O}_{\mathsf{s}}(f_i))_{i\in[d_0]}$ must have already been applied; thus there exists values $(v_i)_{i\in[d_0]}$ such that $\mathcal{A}_{\mathsf{s}}$ can only distinguish $f_i(x)$ from random on $x = v_i$ for all $i \in [d_0]$. This is because these functions are of the form $f_i(x) = a_i x + b_i \mod 2^m$ so learning one evaluation is not sufficient to distinguish the evaluation on another input from random.

When the oracle receives the first decryption query, it evaluates the first one-time program on a random input say $\hat{v}_0$ to get $\hat{f}_0(\hat{v}_0)$. Next, it evaluates the second one-time program on $S(\hat{v}_0\|\hat{f}_0(\hat{v}_0))$ and gets an output say $\hat{y}_0\|\hat{\mu}_0$ (or $\perp$). Let Or_0 denote the sub-algorithm of the oracle which performs this evaluation. Then this entire interaction can be described as $\hat{y}_0\|\hat{\mu}_0 \leftarrow \mathsf{Or}_0(\mathsf{A}_0(\langle \mathcal{O}(f_i), \mathcal{O}(E_{k,f_i,\mu_i})\rangle_{i\in[d_0]}))$. Critically, the algorithm A_0 is oblivious of S and Or_0 has access to only a single evaluation of S, namely $S(\hat{v}_0\|\hat{f}_0(\hat{v}_0))$.

The purpose of this description is to highlight that this entire procedure is essentially performed by an algorithm with access to only a single evaluation of S. By Theorem 9, this algorithm cannot guess $S(x)$ on any input $x \neq \hat{v}_0\|\hat{f}_0(\hat{v}_0)$. By the security of one-time programs, this algorithm can be simulated with a simulator that has access to only a single query to each program $(E_{k,f_i,\mu_i})_{i\in[d_0]}$ instead of $(\mathcal{O}_{\mathsf{s}}(E_{k,f_i,\mu_i}))_{i\in[d_0]}$. The query to E_{k,f_i,μ_i} will yield $\perp$ except with negligible probability unless $\hat{v}_0\|f_i(\hat{v}_0) = \hat{v}_0\|\hat{f}_0(\hat{v}_0)$ since the simulator cannot guess $S(x)$ on $x \neq \hat{v}_0\|\hat{f}_0(\hat{v}_0)$. If the condition $\hat{v}_0\|f_i(\hat{v}_0) = \hat{v}_0\|\hat{f}_0(\hat{v}_0)$ is not satisfied for any $i \in [d_0]$ then the simulator will receive $\perp$ from all its oracle queries except with negligible probability and thus there is a negligible probability that $\hat{y}_0 = S'(\hat{f}_0(\hat{v}_0))$. Note that, there is at most a single value i that satisfies this condition except with negligible probability since the parameters of the functions $(f_i)_{i\in[d_0]}$ are chosen independently and at random.

Assume this condition is satisfied only by the function f_n where $n \in [d_0]$. There is negligible chance that $\hat{v}_0 = v_n$ and so $\mathcal{A}_{\mathsf{s}}$ cannot distinguish $f_n(\hat{v}_0)$ or equivalently $\hat{f}_0(\hat{v}_0)$ from random. The other values used by the oracle in the decryption query are: v, $S(v\|\hat{f}_0(\hat{v}_0))$ and $S'(\hat{f}_0(\hat{v}_0))$. Even if $\mathcal{A}_{\mathsf{s}}$ learns all these values, it does not help in determining functions S and S' since $\hat{f}_0(\hat{v}_0)$ is indistinguishable from random.

At the end of this argument, $\mathcal{A}_{\mathsf{s}}$ cannot distinguish the key k from random. Notice that the only requirement to apply this argument on a decryption query is that $\mathcal{A}$ cannot distinguish k from random when it submits the query. Hence, this argument can be applied to all decryption queries and it can be deduced inductively that $\mathcal{A}_{\mathsf{s}}$ cannot distinguish k from random when it receives the challenge ciphertext. Let $\langle \mathcal{O}_{\mathsf{s}}(f), \mathcal{O}_{\mathsf{s}}(E_{k,f,m_b})\rangle$ be the challenge ciphertext. By one-time program security, the adversary's access to these programs can be simulated with a single evaluation to each program. Hence, it is clear that the probability that the adversary guesses b is upper bounded by $\frac{1}{2} + \mathsf{negl}(n)$. This still holds if k is later revealed since one-time programs disappear after their transmission ends by Theorem 17. ◀

8.2 Asymmetric Key Encryption

We now look into the asymmetric key setting and construct an asymmetric key encryption scheme with information-theoretic disappearing security under chosen-ciphertext attacks in the BQSM. The private key is a large matrix and the public key is a program broadcast of the matrix. Since different users will likely learn different evaluations, an evaluation can be used as a secret key to encrypt messages in the same way as in Construction 21. Note that honest users need a small qmemory to process the broadcast as described in Construction 18. This is in contrast to the private key setting where no qmemory is required.

▶ **Construction 23.** *Let $\overline{\Pi} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ be given in Construction 21 for symmetric key encryption and let $\Pi_{Br} = (\mathsf{KeyGen}, \mathsf{Br}, \mathsf{Eval})$ be the algorithms for program broadcast given in Construction 18. The asymmetric key encryption scheme Π_{AsyK} against adversaries with qmemory bound $\mathtt{s}$ is as follows:*

- *$\mathsf{Gen}(1^\lambda, \mathtt{s})$: Let $m = 100\lceil (\lg \lambda)^3 \rceil$ and $n = \max(\lceil \frac{6\mathtt{s}}{m} \rceil + 1, m)$. Choose uniformly at random $m \times n$ binary matrix M and for $x \in \{0,1\}^m$, let $P(x) := M \cdot x$ be the corresponding program. Generate $ek \leftarrow \mathsf{KeyGen}(1^\lambda, t_{\text{end}})$. The private key is $sk = (ek, P)$.*
- *$\mathsf{KeySend}(\mathtt{s}, sk)$: If queried before t_{end}, then output $O_P \leftarrow \mathsf{Br}(\mathtt{s}, ek, P)$.*
  If queried after t_{end}, `the qmemory bound applies` and output ek.
  Let $\rho_{pk} \leftarrow \langle O_P, ek \rangle$ denote the public key copy.
- *$\mathsf{KeyReceive}(\rho_{pk})$: Randomly choose $v \in_R \{0,1\}^n$. Evaluate the public key $P(v) \leftarrow \mathsf{Eval}(\rho_{pk}, v)$ and output the key $k_v = (v, P(v))$.*
- *$\mathsf{Enc}(\mathtt{s}, k_v, \mu)$: Output $\langle v, \overline{\mathsf{Enc}}(\mathtt{s}, P(v), \mu) \rangle$ [3].*
- *$\mathsf{Dec}(sk, \langle v, \rho_{ct} \rangle)$: Obtain $P(v)$ using sk and v. Output $\overline{\mathsf{Dec}}(P(v), \rho_{ct})$.*

The algorithm KeyReceive requires $100(\lg \lambda)^3$ qmemory to run and the scheme is secure against adversaries with $\mathtt{s}$ qmemory, where $\mathtt{s}$ can be any polynomial in λ. Hence, by setting $m := 100(\lg \lambda)^3$, the gap between the qmemory of the honest user and the adversary is m vs. $e^{\sqrt[3]{m}}$.

▶ **Theorem 24** (Security). *Construction 23 (Π_{AsyK}) satisfies qDCCA1 security against computationally unbounded adversaries with qmemory bound $\mathtt{s}$.*

Proof. An adversary $\mathcal{A}_\mathtt{s}$ in the qDCCA1 security experiment can request a polynomial number of public key copies ρ_{pk}. The proof can be realized in the following steps.

1. By the security of our program broadcast protocol (Theorem 19) $\mathcal{A}_\mathtt{s}$ can be simulated with an algorithm $\mathcal{S}_\mathtt{s}$ that is given $\frac{6\mathtt{s}}{m}$ queries to P.
2. By Theorem 9, since M is a matrix of dimension $\lceil \frac{m}{12} \rceil \times n$ and $n > \frac{6\mathtt{s}}{m}$, $\mathcal{S}_\mathtt{s}$ cannot guess the output of $P(x) = M \cdot x$ on a random input except with negligible probability.
3. This means that there is negligible chance $\mathcal{S}_\mathtt{s}$ can determine the sub-key k_v extracted from the public key ρ_{pk} in the experiment which reduces the proof to the private key setting. Theorem 22 gives the result. ◀

9 Note on Noisy Communication

Our protocols assume that the communication between the sender and receiver is error-free. It is assumed that the sender has a perfect quantum source which when requested will produce one and only one qubit in the correct state. Note if the source accidentally produces

[3] The string $P(v)$ is of length $m = 100\lceil (\lg \lambda)^3 \rceil$, so it is of sufficient length such that it can be interpreted as the secret key in the symmetric-key encryption scheme $\overline{\Pi}$.

two copies of a qubit then the adversary can measure the qubit in two bases and break the 1-2 oblivious transfer scheme. Furthermore, it requires the honest evaluator to measure all qubits without error.

Fortunately, in [10] the authors used techniques based on information reconciliation to allow an honest evaluator and sender in the quantum 1-2 oblivious transfer scheme with imperfect apparatus to perform error correction without compromising the security of the scheme. The same techniques are applicable to our protocols since the quantum component of our one-time programs consist of oblivious transfer transmissions. However, we leave a more rigorous treatment of these issues as an avenue for future work.

References

1 Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001. doi:10.1007/3-540-44647-8_1.

2 Mohammed Barhoush and Louis Salvail. How to sign quantum messages. *arXiv preprint arXiv:2304.06325*, 2023. doi:10.48550/arXiv.2304.06325.

3 Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 449–458, 2002. doi:10.1109/SFCS.2002.1181969.

4 Khashayar Barooti, Giulio Malavolta, and Michael Walter. A simple construction of quantum public-key encryption from quantum-secure one-way functions. *arXiv preprint*, 2023. arXiv:2303.01143.

5 Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. *Quantum*, 7:901, 2023. doi:10.22331/Q-2023-01-19-901.

6 Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. In *Annual Cryptology Conference*, pages 71–89. Springer, 2014. doi:10.1007/978-3-662-44381-1_5.

7 Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Annual Cryptology Conference*, pages 344–360. Springer, 2013.

8 Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Annual international cryptology conference*, pages 13–25. Springer, 1998. doi:10.1007/BFB0055717.

9 Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Cryptography in the bounded-quantum-storage model. *SIAM Journal on Computing*, 37(6):1865–1890, 2008. doi:10.1137/060651343.

10 Ivan B Damgård, Serge Fehr, Renato Renner, Louis Salvail, and Christian Schaffner. A tight high-order entropic quantum uncertainty relation with applications. In *Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27*, pages 360–378. Springer, 2007. doi:10.1007/978-3-540-74143-5_20.

11 Yevgeniy Dodis, Willy Quach, and Daniel Wichs. Speak much, remember little: Cryptography in the bounded storage model, revisited. *Cryptology ePrint Archive*, 2021. URL: https://eprint.iacr.org/2021/1270.

12 Yevgeniy Dodis, Willy Quach, and Daniel Wichs. Authentication in the bounded storage model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 737–766. Springer, 2022. doi:10.1007/978-3-031-07082-2_26.

13 Javad Doliskani. Efficient quantum public-key encryption from learning with errors. *arXiv preprint*, 2021. arXiv:2105.12790.

14 Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 553–562. IEEE, 2005. doi:10.1109/SFCS.2005.60.

15 Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. One-time programs. In *Annual International Cryptology Conference*, pages 39–56. Springer, 2008. doi:10.1007/978-3-540-85174-5_3.

16 Daniel Gottesman. Uncloneable encryption. *arXiv preprint*, 2002. arXiv:quant-ph/0210062.

17 Daniel Gottesman and Isaac Chuang. Quantum digital signatures. *arXiv preprint*, 2001. arXiv:quant-ph/0105032.

18 Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7*, pages 308–326. Springer, 2010. doi:10.1007/978-3-642-11799-2_19.

19 Jiaxin Guan, Daniel Wichs, and Mark Zhandry. Incompressible cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–730. Springer, 2022. doi:10.1007/978-3-031-06944-4_24.

20 Jiaxin Guan and Mark Zhandary. Simple schemes in the bounded storage model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 500–524. Springer, 2019. doi:10.1007/978-3-030-17659-4_17.

21 Jiaxin Guan and Mark Zhandry. Disappearing cryptography in the bounded storage model. In *Theory of Cryptography Conference*, pages 365–396. Springer, 2021. doi:10.1007/978-3-030-90453-1_13.

22 Akinori Kawachi, Takeshi Koshiba, Harumichi Nishimura, and Tomoyuki Yamakami. Computational indistinguishability between quantum states and its cryptographic application. In *Eurocrypt*, volume 3494, pages 268–284. Springer, 2005. doi:10.1007/11426639_16.

23 Joe Kilian. Founding cryptography on oblivious transfer. In *Proc. ACM Symposium on Theory of Computing*, STOC '88, pages 20–31, New York, NY, USA, 1988. ACM. doi:10.1145/62212.62215.

24 Robert König and Renato Renner. Sampling of min-entropy relative to quantum knowledge. *IEEE Transactions on Information Theory*, 57(7):4760–4787, 2011. doi:10.1109/TIT.2011.2146730.

25 Robert König, Renato Renner, and Christian Schaffner. The operational meaning of min- and max-entropy. *IEEE Transactions on Information theory*, 55(9):4337–4347, 2009. doi:10.1109/TIT.2009.2025545.

26 Ueli M Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992. doi:10.1007/BF00191321.

27 Michele Mosca, Alain Tapp, and Ronald de Wolf. Private quantum channels and the cost of randomizing quantum information. *arXiv preprint*, 2000. arXiv:quant-ph/0003101.

28 Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *Journal of the ACM (JACM)*, 66(1):1–18, 2018. doi:10.1145/3186563.

29 Renato Renner and Robert König. Universally composable privacy amplification against quantum adversaries. In *Theory of Cryptography Conference*, pages 407–425. Springer, 2005. doi:10.1007/978-3-540-30576-7_22.

30 Christian Schaffner. *Cryptography in the Bounded-Quantum-Storage Model.* PhD thesis, Aarhus Universitet, 2007. arXiv:0709.0289.

31 Stephanie Wehner, Christian Schaffner, and Barbara M. Terhal. Cryptography from noisy storage. *Phys. Rev. Lett.*, 100:220502, June 2008. doi:10.1103/PhysRevLett.100.220502.

32 Andreas Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999. doi:10.1109/18.796385.