

Cloning Games: A General Framework for Unclonable Primitives

Prabhanjan Ananth*
UCSB

Fatih Kaleoglu†
UCSB

Qipeng Liu‡
Simons Institute

Abstract

The powerful no-cloning principle of quantum mechanics can be leveraged to achieve interesting primitives, referred to as unclonable primitives, that are impossible to achieve classically. In the past few years, we have witnessed a surge of new unclonable primitives. While prior works have mainly focused on establishing feasibility results, another equally important direction, that of understanding the relationship between different unclonable primitives is still in its nascent stages. Moving forward, we need a more systematic study of unclonable primitives.

To this end, we introduce a new framework called *cloning games*. This framework captures many fundamental unclonable primitives such as quantum money, copy-protection, unclonable encryption, single-decryptor encryption, and many more. By reasoning about different types of cloning games, we obtain many interesting implications to unclonable cryptography, including the following:

1. We obtain the first construction of information-theoretically secure single-decryptor encryption in the one-time setting.
2. We construct unclonable encryption in the quantum random oracle model based on BB84 states, improving upon the previous work, which used coset states. Our work also provides a simpler security proof for the previous work.
3. We construct copy-protection for single-bit point functions in the quantum random oracle model based on BB84 states, improving upon the previous work, which used coset states, and additionally, providing a simpler proof.
4. We establish a relationship between different challenge distributions of copy-protection schemes and single-decryptor encryption schemes.
5. Finally, we present a new construction of one-time encryption with certified deletion.

*prabhanjan@cs.ucsb.edu

†kaleoglu@ucsb.edu

‡qipengliu0@gmail.com

Contents

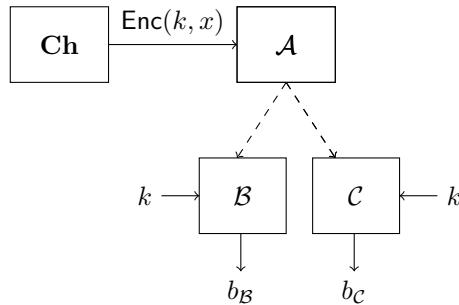
1	Introduction	3
1.1	Complexity of Unclonable Primitives	5
1.2	Our Contributions	7
2	Technical Overview	9
2.1	Definitional Contribution: Cloning Games	9
2.2	Part I: Implications via Classical Reductions	12
2.3	Part II: Generalizing Existing Results	18
2.4	Generalized Cloning Games	20
2.5	Future Directions	21
2.6	Organization	21
3	Preliminaries	22
3.1	Quantum Computing Basics & Query Bounds	22
3.2	Jordan’s Lemma	23
3.3	Applications of Jordan’s Lemma	24
3.4	Unlearnable Distributions	27
4	Cloning Games - Definitions	27
4.1	Correctness	27
4.2	Special Types of Cloning Games	28
4.3	Security	28
4.4	Extended Definitions	30
4.5	Examples	33
5	Constructive Post-Quantum Reductions: The Non-Local Setting	36
5.1	Definitions	36
5.2	Main Theorem	41
5.3	Proof of the Main Theorem.	43
5.4	Applications	53
6	Relating Unclonable-Search and Unclonable-Indistinguishability	56
6.1	Search-to-Indistinguishability	57
6.2	Indistinguishability-to-Search	63
7	From Search to Decision Games	64
8	Asymmetric Cloning Games	69
8.1	Definitions	69
8.2	Deletion Games	72
8.3	Construction of Unclonable Encryption with Certified Deletion	73
A	Alternate Proof of Simultaneous Quantum Goldreich-Levin	84

1 Introduction

Unclonable cryptography is a prominent research area that lies at the intersection of quantum computing and cryptography. This research area consists of many fascinating primitives that solve cryptographic problems using quantum information that are impossible to solve using only classical technology. At the heart of this area is the no-cloning principle of quantum mechanics [WZ82, Die82], which states that no universal cloner can clone arbitrary quantum states. Since Wiesner put forward quantum money in 1983 [Wie83], a novel unclonable primitive that protects digital money against counterfeiting attacks, there have been a myriad of interesting unclonable primitives proposed over the years. They include variants of quantum money [AC12, Zha17, Shm22], quantum one-time programs [BGS13], copy-protection [Aar09, AL20, ALL⁺20, CLLZ21], tokenized signatures [BS16, CLLZ21, Shm22], unclonable encryption [Got02, BL20], secure software leasing [AL20, KNY21, B JL⁺21], encryption with certified deletion [BI20] and certified zero-knowledge [HMNY22].

We discuss three unclonable primitives that are the main focus of this work¹.

Unclonable Encryption. Roughly speaking, an unclonable encryption scheme, introduced by [BL20], is a type of symmetric key encryption scheme that protects ciphertexts, encoded in quantum states, from being illegally distributed. To formalize this, we first consider the following security experiment. The adversary participating in the security experiment is referred to as a cloning adversary, consisting of three algorithms, namely \mathcal{A} , \mathcal{B} and \mathcal{C} . \mathcal{A} receives a quantum state in the *setup phase*. The quantum state is a ciphertext, which is an encryption of a message m computed using a private key k . Then, \mathcal{A} sends a bipartite state to the spatially separated parties (\mathcal{B}, \mathcal{C}) during the *splitting phase*. Finally, \mathcal{B} and \mathcal{C} are asked to simultaneously pass verification in the *challenge phase*. In more detail, in the challenge phase, \mathcal{B} and \mathcal{C} both receive the classical decryption key k and \mathcal{B} outputs $b_{\mathcal{B}}$ while \mathcal{C} outputs $b_{\mathcal{C}}$. We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins if $b_{\mathcal{B}} = b_{\mathcal{C}} = m$.



With the above security experiment in mind, there are two ways to define security.

- *Unclonability:* In this case, m is sampled uniformly at random from the message space. We say that the scheme is ε -secure if the probability that the adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins is ε . Ideally, we would require that ε is negligible in $|m|$.
- *Unclonable-Indistinguishability:* In this case, m is sampled uniformly at random² from some

¹An (impatient) reader familiar with the above primitives could skip directly to Section 1.1. We still recommend going through the discussion before reading Section 1.1.

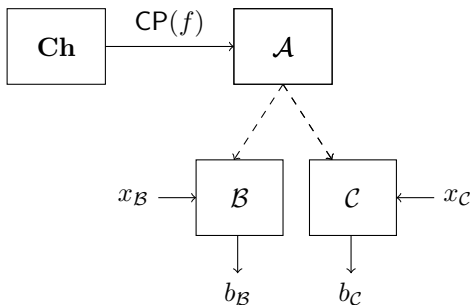
²We note that the security in the literature is stated slightly differently. \mathcal{A} is given encryption of a message m_b , where b is picked uniformly at random and \mathcal{B}, \mathcal{C} is expected to simultaneously guess b . We note that this formulation is identical to the above formulation.

adversarially chosen set $\{m_0, m_1\}$. Similar to the above property, we can define ε -security. Ideally, we would require ε to be negligibly close to 0.5.

Public-key unclonable encryption schemes have also been considered by [AK21, AKL⁺22].

Copy-Protection. Quantum copy-protection, introduced in [Aar09], is a functionality-preserving compiler that transforms programs into quantum states. Moreover, we require that the resulting copy-protected state should not allow the adversary to copy the functionality of the state.

The security experiment against cloning adversaries of the form $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is formalized as follows³. \mathcal{A} receives an unclonable copy-protected program $\rho_f := \text{CP}(f)$, which can be used to evaluate a classical function f ⁴. In the challenge phase, \mathcal{B} and \mathcal{C} receive inputs x_B, x_C , sampled from a *challenge distribution* and are asked to output b_B, b_C . $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins if $b_B = f(x_B)$ and $b_C = f(x_C)$, respectively.



Ideally, we would like to say that a copy-protection scheme is secure if the probability that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins is negligible in the output lengths. However, such a statement would be false if we are not careful in choosing the distributions from which f is sampled and x_B, x_C are sampled. For example, if f is sampled from a distribution with support size one, then the adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ clearly knows the function being copy-protected and can thus easily violate the security. Even if f is sampled from a high-entropy distribution, we should also require x_B and x_C to come from high-entropy distributions for the definition to be meaningful. For example, if we set x_C to be a fixed element, then $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ can always win by \mathcal{A} first computing the value of the function $f(x_C)$ and then handing over the copy-protected state to \mathcal{B} and then handing over $f(x_C)$ to \mathcal{C} . Moreover, our definition should be robust in even handling functions with single-bit outputs. This suggests that we must carefully examine the challenge distribution when evaluating results on constructions of copy-protection schemes.

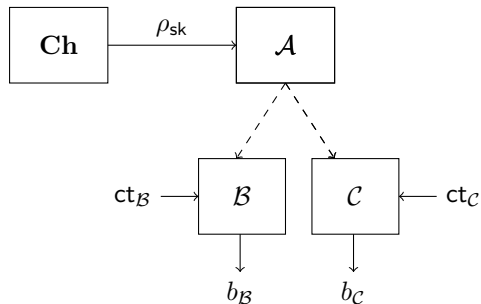
Single-Decryptor Encryption. A single-decryptor encryption, introduced in [GZ20, CLLZ21], enables a user to delegate their decryption key, represented as a quantum state, such that the delegated key cannot be used to illegally distribute two or more decryption keys that can decrypt ciphertexts.

Formally, \mathcal{A} receives an unclonable decryption key ρ_{sk} for an encryption scheme where the encryption procedure and the message space are both classical. In the challenge phase, \mathcal{B} and \mathcal{C}

³The original formulation by [Aar09] is weaker than what is stated here. We follow the game-based definition by [CMP20].

⁴We only consider classes of unlearnable functions which are functions that cannot be efficiently learned from its input and output behavior. Copy-protection for learnable functions is impossible.

respectively receive ciphertexts $\text{ct}_{\mathcal{B}}, \text{ct}_{\mathcal{C}}$ encrypting messages $m_{\mathcal{B}}, m_{\mathcal{C}}$. They are then expected to output $b_{\mathcal{B}}, b_{\mathcal{C}}$ respectively.



Depending on the specification of the distributions from which the ciphertexts are sampled and how $b_{\mathcal{B}}, b_{\mathcal{C}}$ are defined, there are many ways to define security for single-decryptor encryption.

- We could require $\text{ct}_{\mathcal{B}} = \text{ct}_{\mathcal{C}}$ (referred to as *identical* ciphertext distribution), in which case $m_{\mathcal{B}} = m_{\mathcal{C}}$, or we could require that $\text{ct}_{\mathcal{B}}$ and $\text{ct}_{\mathcal{C}}$ to be sampled independently (referred to as *independent* ciphertext distribution).
- Analogous to the unclonable encryption setting, we could require that the messages $m_{\mathcal{B}}, m_{\mathcal{C}}$ are picked from the uniform distribution, or they are sampled from a set of two messages chosen by $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

1.1 Complexity of Unclonable Primitives

Most prior works on unclonable encryption, copy-protection, single-decryptor encryption, and other unclonable primitives mainly focus on feasibility. A few exceptions include the works of [CMP20, AK21, SW22], who make partial progress in understanding the relationship between unclonable encryption, copy-protection, and single-decryptor encryption.

In order to achieve a deeper understanding of the area, we need to move beyond the feasibility results and investigate how different primitives are related to each other. There are many reasons why we should care about understanding the relationship between unclonable primitives, and we discuss some of them below.

Computational Assumptions. Firstly, it leads to a better understanding of the computational assumptions necessary in the conception of unclonable primitives. While some primitives require powerful cryptographic tools such as post-quantum indistinguishability obfuscation, some other primitives can even be conceived information-theoretically. It would be interesting to classify the unclonable primitives based on the computational assumptions necessary to construct them. In classical cryptography, via Impagliazzo’s five worlds [Imp95] and numerous black-box separations [IR90], we have a solid understanding of the minimal computational assumptions necessary for the existence of primitives. We have just begun to understand the assumptions necessary for achieving cryptographic primitives in the quantum world [AQY22, MY22, BCQ22]. Investigating the implications between the unclonable primitives will help us classify these primitives based on their computational hardness.

Types of States. Secondly, not all unclonable primitives use the same types of states. Given any unclonable primitive, it is important to establish the types of states needed to achieve this notion. Some primitives [Wie83, BI20, BL20] use BB84 states [BB20], some utilize subspace states [AC12, Zha17] and others take advantages of coset states [VZ20, CLLZ21, AKL⁺22]. BB84 states are preferred over subspace and coset states due to two facts: (a) they can be prepared easily (the preparation requires only Hadamard and X gates), and (b) each qubit is unentangled with the other qubits. Since maintaining entanglement has been challenging in the existing quantum systems, understanding the feasibility of cryptographic systems using unentangled states is important. We currently have a limited understanding of whether BB84 states are sufficient for constructing many primitives. For instance, copy-protection for point functions with single-bit output seems to require coset states [AKL⁺22] whereas copy-protection for point functions with multi-bit output requires only BB84 states [CMP20].

Challenge Distributions. Unclonable primitives are often associated with challenge distributions. Thus, different feasibility results on the same unclonable primitive assuming different challenge distributions can be qualitatively incomparable. As was seen in the examples of unclonable encryption, copy-protection and single-decryptor encryption, the security of an unclonable primitive can be defined as a game between a challenger and an adversary composed of three parts ($\mathcal{A}, \mathcal{B}, \mathcal{C}$). First, \mathcal{A} receives an unclonable state (a.k.a. a quantum token) from the challenger, and it outputs a bipartite state shared by \mathcal{B} and \mathcal{C} . Then, \mathcal{B} and \mathcal{C} receive samples from a distribution, called a *challenge distribution*, and they output answers.

It is often the case that security proven with respect to one challenge distribution does not necessarily imply security proven for a different challenge distribution. For instance, as was discussed earlier in the context of copy-protection, the choice of challenge distribution can qualitatively affect the type of result we get. Discerning the relationship between security notions of different challenge distributions will enable us to compare different results based on the challenge distributions they consider. Indeed, even in the literature, constructions of copy-protection for point functions have considered different challenge distributions [CMP20, BJL⁺21, AKL⁺22], which makes their results difficult to compare. Besides point functions, copy-protection was only known under certain distributions (product distributions).

Porting Classical Techniques. It turns out to be challenging to adopt many standard techniques employed to prove the security of cryptographic systems in the classical cryptography literature to the unclonable setting. Let us take an example. Traditionally, encrypting multiple bits can be generically reduced to encrypting single-bit messages in parallel using a simple hybrid argument. The same transformation fails when applied to the setting of unclonable encryption. Even standard search-to-decision reductions, such as Goldreich-Levin [GL89], commonly used in the classical cryptography literature, cannot be directly ported to the unclonable setting. In the context of unclonable encryption, [AKL⁺22] discuss the challenges associated with using Goldreich-Levin and more in Section 1.1 of their work.

1.2 Our Contributions

In order to better understand the relationship between the unclonable primitives, we propose a new framework called *Cloning Games*. Firstly, we observe that many⁵

fundamental unclonable primitives can be cast as cloning games. We establish the relationship between large classes of cloning games. There are two directions we undertake to establish the relationship between cloning games.

1. In the first approach, we show that, under some conditions, the relationship between different cloning games can be reduced to the existence of classical reductions between two non-interactive assumptions. This approach gives a new toolkit to help us use classical techniques and computational assumptions to build unclonable primitives. We give an overview of this approach in Section 2.2.
2. In the second approach, using new techniques, we refurbish existing constructions of primitives into generic transformations between cloning games. This approach leads to new constructions of primitives with improved features over prior works. An overview of this approach is given in Section 2.3.

As a consequence of the above two approaches, we obtain new results in unclonable cryptography.

Single-Decryptor Encryption. Existing constructions of single-decryptor encryption in the public-key setting, are based on post-quantum indistinguishability obfuscation [GZ20, CLLZ21]. It is worth investigating whether we can achieve single-decryptor encryption in the private-key setting based on well-studied assumptions. Indeed, even in the one-time setting, it was not known how to achieve single-decryptor encryption without relying on strong assumptions. By one-time setting, we mean that the adversary only gets one ciphertext computed using the private key. We show the following.

Theorem 1 (Informal). *There exists an information-theoretically secure one-time single-decryptor encryption scheme for single-bit messages.*

The ciphertext distribution we consider in the above result is the following:

The challenger chooses the messages $m_{\mathcal{B}} \xleftarrow{\$} \{0, 1\}$ and $m_{\mathcal{C}} \xleftarrow{\$} \{0, 1\}$. It then encrypts $m_{\mathcal{B}}$ (resp., $m_{\mathcal{C}}$) and gives the ciphertext to \mathcal{B} (resp., \mathcal{C}). Then, \mathcal{B} and \mathcal{C} are supposed to simultaneously guess which bit was encrypted. The security of our construction states that the success probability of any adversary is negligibly close to 0.5.

Although our construction is only for 1-bit messages, we hope the toolkit we develop (Theorem 13) can be applied to obtain single-decryptor encryption for multi-bit messages in future work.

Unclonable Encryption and Copy-Protection. We revisit recent works that leveraged coset states to achieve unclonable primitives. Specifically, we focus on two constructions of unclonable-indistinguishable encryption and copy-protection for point functions by [AKL⁺22]. We show that these two constructions can be obtained from *any* encryption scheme satisfying the unclonability property in the quantum random oracle model. Ours is the first work to formally establish the relationship between unclonability and unclonable-indistinguishability properties.

⁵As far as we know, all unclonable primitives can be cast as cloning games by making reasonable minor modifications to the framework.

Theorem 2. *Assuming the existence of one-time encryption satisfying unclonability, there exists an encryption scheme satisfying unclonable-indistinguishability in the quantum random oracle model.*

Assuming the existence of one-time encryption satisfying unclonability, there exists a copy-protection scheme for 1-bit output point functions in the quantum random oracle model.

Unclonable encryption can be constructed from BB84 states [BL20], and hence, as a consequence, we can obtain unclonable-indistinguishable encryption and copy-protection for point functions leveraging just BB84 states.

Corollary 1 (Informal). *There exists a (one-time) encryption scheme satisfying unclonable indistinguishability property, based on BB84 states, in the quantum random oracle model.*

There exists copy-protection for 1-bit output point functions, based on BB84 states, in the quantum random oracle model.

In fact, [AK21] showed that encryption satisfying unclonability can be obtained from a variety of monogamy of entanglement games [TFKW13]. Consequently, we obtain both unclonable-indistinguishable encryption and copy-protection schemes based on a variety of quantum states, not just BB84 states.

Moreover, by plugging in the generic transformation from [AK21], we achieve public-key unclonable encryption based on BB84 states.

Relationship between Challenge Distributions. In both copy-protection and single-decryptor encryption, the choice of challenge distribution plays a role in determining the usefulness of constructions. This makes comparing results difficult. For instance, a priori, it is unclear how to compare two different works constructing copy-protection for the same class of functions but with different challenge distributions. Similarly, even for single-decryptor encryption, schemes with different ciphertext distributions might be incomparable.

We make progress in understanding the relationship between different challenge distributions. Although our result is more general, for the current discussion, let us focus on two types of distributions:

- *Identical:* Both \mathcal{B} and \mathcal{C} get the same challenge (challenge refers to input in the case of copy-protection and ciphertext in the case of single-decryptor encryption), drawn from some distribution.
- *Independent:* \mathcal{B} and \mathcal{C} each get two challenges chosen independently from some distribution.

Although being quite similar, the relationship between security under identical-challenge cloning experiments and independent-challenge cloning experiments was not known, as all the security proofs of general copy-protection schemes [ALL⁺20, CLLZ21] were established with respect to independent-challenge distributions⁶, and their security with respect to identical-challenge distributions was not analyzed. Indeed, it turns out that the proof techniques in [ALL⁺20, CLLZ21] were tailored to the independent challenge setting and they did not generalize to the identical challenge setting.

We address this issue by showing the following.

⁶With the exception of copy-protection of point functions [CMP20, AKL⁺22].

Theorem 3 (Informal). *A copy-protection scheme secure for a class of multi-bit output functions in the independent challenge distribution setting is also secure in the identical challenge distribution setting.*

A single-decryptor encryption scheme in the independent challenge distribution setting is also secure in the identical challenge distribution setting.

For the result on copy-protection, we remark that besides the fact that the output length of the functions is large (more precisely, depends on the security parameter), our result is general and applies to any class of functions. For the result of single-decryptor encryption, we consider the definition where the adversary is given the encryption of a message chosen from the uniform distribution and is supposed to predict the entire message.

In fact, in the technical sections, we prove a stronger theorem that generalizes for arbitrary correlated distributions instead of just identical distributions! More precisely, suppose $\mathcal{D}_{\mathcal{B}}$ (resp., $\mathcal{D}_{\mathcal{C}}$) is the challenge distribution for \mathcal{B} (resp., \mathcal{C}). Let \mathcal{D} be the challenge distribution on \mathcal{B} 's and \mathcal{C} 's challenge spaces, as long as the marginal distribution on \mathcal{B} (or \mathcal{C} respectively) of \mathcal{D} corresponds to $\mathcal{D}_{\mathcal{B}}$ (or $\mathcal{D}_{\mathcal{C}}$, respectively). We show that a secure copy-protection scheme when the challenge distribution is $\mathcal{D}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{C}}$, is also secure when the challenge distribution is \mathcal{D} . Similar conclusions also hold for single-decryptor encryption schemes.

Encryption with Certified Deletion. Another well-studied unclonable primitive is encryption with certified deletion [BI20]. Certified deletion can be thought of as a weaker form of unclonability, where the adversary is asked to provide a classical certificate of deletion before learning the secret key.⁷ While it is unknown whether unclonable encryption is information theoretically possible, encryption with certified deletion is known to be information theoretically possible [BI20, BK22]. We give an alternate construction and proof of security of this construction is based on the techniques used for bounding monogamy-of-entanglement games [TFKW13]. Our techniques are conceptually different from the existing works [BI20, BK22] who used entropic arguments to argue the same. En route, we formally define the notion of *deletion games*, a subclass⁸ of cloning games.

2 Technical Overview

We first discuss our definition of cloning games, why it captures many existing unclonable primitives, and then present techniques to relate different cloning games.

2.1 Definitional Contribution: Cloning Games

A cloning game consists of the following four procedures (**Setup**, **GenT**, **GenC**, **Ver**):

- A setup procedure **Setup**, on input a security parameter, outputs a secret key sk .
- A token generation procedure **GenT** that takes as input the secret key sk , a message m , and outputs a quantum state ρ . As we will see later, ρ is expected to have some unclonability properties.

⁷In contrast, unclonability allows \mathcal{B} and \mathcal{C} to both learn the secret key before passing verification. Note that we use the word "weaker" qualitatively in this sentence, and do *not* claim that unclonability implies the existence of certified deletion in general.

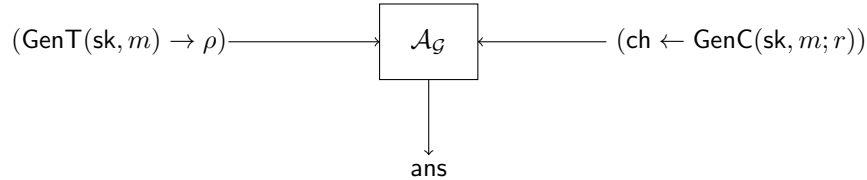
⁸Although this is not true for the initial definition we use to introduce cloning games, deletion games are captured after considering a natural extension of cloning games, where \mathcal{B} and \mathcal{C} are not treated symmetrically.

- A challenge generation procedure GenC , which takes sk, m together with random coins r , and outputs a challenge ch .
- Finally, a verification procedure Ver , that takes sk, m, ch together with an alleged answer (which can be either a classical string or a quantum state) and outputs either 0 (reject) or 1 (accept).

We also consider another (stateful) variant where Ver gets as input r , which are the random coins used in GenC .

We require that a cloning game satisfies two properties: correctness and security. First, we discuss correctness.

Correctness. The correctness property says that there always exists an (efficient) quantum algorithm \mathcal{A}_G if all the procedures are executed honestly and in the order of $(\text{Setup}, \text{GenT}, \text{GenC}, \mathcal{A}_G)$, the verification procedure should almost always output 1 (accept). That is, \mathcal{A}_G takes as input the state produced by GenT and the challenge produced by GenC and outputs an answer ans that is accepted by Ver with probability negligibly close to 1.



2.1.1 Instantiations

Before we discuss security, we demonstrate the power of cloning games by showing a couple of examples. Below, we show that both unclonable encryption and copy-protection can be cast as cloning games.

Unclonable Encryption. We cast unclonable encryption as a cloning game $(\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ below.

- Setup in the cloning game corresponds to the key generation of the unclonable encryption scheme. That is, Setup produces the secret key, denoted by sk , of the encryption scheme,
- GenT corresponds to the encryption algorithm,
- GenC produces the challenge $\text{ch} = \text{sk}$,
- Ver takes as input $(\text{sk}, m, \text{ch}, \text{ans})$ and outputs 1 if and only if $\text{ans} = m$.
- \mathcal{A}_G corresponds to the decryption algorithm. On input the ciphertext state produced by GenT and the secret key, i.e., ch , it outputs the message m .

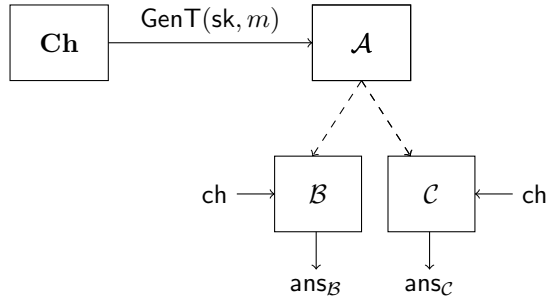
Copy-Protection. We can similarly cast copy-protection using cloning games, as shown below. We do not need to define **Setup** for copy-protection of classical programs and thus $\text{sk} = \perp$.

- **GenT** takes $\text{sk} = \perp$, message $m = f$, where f is the program to be copy-protection and outputs a quantum state ρ . That is, **GenT** corresponds to the copy-protection algorithm,
- **GenC** takes as input $\text{sk} = \perp$, $m = f$ and samples a challenge $\text{ch} := x$ according to the distribution.
- **Ver** corresponds to the evaluation algorithm of the copy-protection scheme. That is, it takes input $\text{ch} := x$ and f , and tests whether $f(x) = \text{ans}$.
- $\mathcal{A}_{\mathcal{G}}$ corresponds to the evaluation algorithm. On input the copy-protected state of f and the challenge input x , it outputs $f(x)$.

Similarly, single decryptor encryption, tokenized signatures, primitives with certified deletion, and many others can be cast as cloning games. We refer the reader to the main body for more details.

2.1.2 Security

In the security experiment, we consider cloning adversaries of the form $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. \mathcal{A} receives as input a quantum state ρ generated using $\text{GenT}(\text{sk}, m)$. \mathcal{A} then computes a bipartite state and sends it to \mathcal{B} and \mathcal{C} . Both \mathcal{B} and \mathcal{C} then receive ch , where ch is produced by $\text{GenC}(\text{sk}, m)$. $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins if $\text{ans}_{\mathcal{B}}$ produced by \mathcal{B} and $\text{ans}_{\mathcal{C}}$ produced by \mathcal{C} are such that **Ver** accepts both $\text{ans}_{\mathcal{B}}$ and $\text{ans}_{\mathcal{C}}$.



To define the security, we first define the *trivial success probability* of the adversaries in the cloning game. We say that the cloning game is secure as long as any cloning adversary cannot succeed with probability significantly larger than the trivial success probability. The trivial success probability is calculated as follows: \mathcal{A} gives the quantum token to \mathcal{B} , and then \mathcal{B} computes the correct answer $\text{ans}_{\mathcal{B}}$. On the other hand, \mathcal{C} outputs its best guess $\text{ans}_{\mathcal{C}}$. The probability that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins is precisely the trivial success probability.

The trivial success probability in an encryption scheme satisfying unclonability⁹ is $\frac{1}{|\mathcal{M}|}$, where \mathcal{M} is the message space, and the trivial success probability in a scheme satisfying unclonable-indistinguishability is $\frac{1}{2}$.

⁹Please refer to the definition of unclonability of an unclonable encryption scheme in the introduction.

Computational Complexity of the Attackers. We did not remark on the computational complexity of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. In this work, we mainly work with attackers where \mathcal{A}, \mathcal{B} , and \mathcal{C} are all computationally unbounded adversaries. Nevertheless, we can consider more general settings, where all of them run in quantum polynomial time.

Message Distributions. In the security experiment, m is sampled from some distribution \mathcal{D} . There are two types of distributions we consider in this work: (1) \mathcal{D} is uniform and, (2) \mathcal{D}_{m_0, m_1} is a distribution parameterized by two messages m_0, m_1 and it outputs m_0 or m_1 with equal probability $1/2$. When considering \mathcal{D}_{m_0, m_1} , we allow the adversary to choose the messages m_0, m_1 .

Search and Decision Games. We consider a specific type of cloning games, called *search* games, where the verification algorithm Ver is defined as follows: on input $(\text{sk}, m, \text{ch}, \text{ans})$, it outputs 1 (or Valid) if and only if $\text{ans} = m$. We can consider two different security notions of search games.

- *Unclonable-Search* security: the message distribution \mathcal{D} is uniform and,
- *Unclonable-Indistinguishability* security: the message distribution is \mathcal{D}_{m_0, m_1} , where (m_0, m_1) is the pair of messages chosen by the cloning adversary.

In the context of unclonable encryption, the above two notions correspond to unclonability and unclonable-indistinguishability properties.

We also define *decision* games, where $\text{ans}_{\mathcal{B}}, \text{ans}_{\mathcal{C}} \in \{0, 1\}$.

Extensions and Stateful games. For some applications, we need to generalize the algorithms of the cloning games further. Firstly, we can generalize the challenge generation phase to the asymmetric setting, where both \mathcal{B} and \mathcal{C} do not necessarily receive the same challenge. This is formalized by defining an extended algorithm $\widetilde{\text{GenC}}$ which samples two random strings $r_{\mathcal{B}}$ and $r_{\mathcal{C}}$ such that \mathcal{B} (resp., \mathcal{C}) receives a challenge generated using $r_{\mathcal{B}}$ (resp., $r_{\mathcal{C}}$). Furthermore, we generalize the verification algorithm Ver to also take as input the randomness used in the challenge generation algorithm. This way, the pair of algorithms $(\widetilde{\text{GenC}}, \text{Ver})$ acts as a *stateful* verifier, hence the term stateful games.

Challenge Distributions. Finally, we need to remark on how the randomness for the challenge generation is generated. There are two popular options:

- *Identical challenge distribution:* in this case, $\widetilde{\text{GenC}}$ generates $r_{\mathcal{B}} = r_{\mathcal{C}}$.
- *Independent challenge distribution:* in this case, $\widetilde{\text{GenC}}$ generates $r_{\mathcal{B}}, r_{\mathcal{C}}$ such that $r_{\mathcal{B}}$ and $r_{\mathcal{C}}$ are chosen independently.

We also consider more general challenge distributions where $r_{\mathcal{B}}$ and $r_{\mathcal{C}}$ are arbitrarily correlated.

2.2 Part I: Implications via Classical Reductions

In the classical cryptography literature, there is an abundance of techniques developed to show the relationship between different primitives. Ideally, we would like to draw inspiration from these techniques and/or rehash them to develop new relationships between unclonable primitives.

We develop a new framework to relate cloning games using classical reductions. This new framework presents a new approach of using classical techniques to build unclonable primitives.

Specifically, we show that the implication of a cloning game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ to another cloning game $\mathcal{G}' = (\text{Setup}', \text{GenT}', \text{GenC}', \text{Ver}')$ can be based on a classical reduction that transforms a probabilistic polynomial-time solver for one assumption into a solver for a different assumption, where the assumptions are closely related to the games $\mathcal{G}, \mathcal{G}'$.

For the implication to hold, we require some extra (and mild) conditions. In the simplest case, $\text{Setup} = \text{Setup}'$ and $\text{GenT}' = \text{GenT}$. More generally, we require that the distribution of the states generated by GenT is close (in trace distance) to the distribution of the states generated by GenT' . Additionally, we require that in both games, the trivial success probability is negligible¹⁰.

Implications. As a result of the above implication, we obtain two interesting sets of results.

Firstly, we can show that many unclonable primitives (for instance, copy-protection and single-decryptor encryption schemes secure with respect to independent challenge distribution are also secure with respect to an arbitrary challenge distribution, as long as the marginals of the latter distribution correspond to the independent challenge distribution. This follows from the fact that changing the challenge distribution corresponds to only modifying the algorithms GenC and Ver .

Secondly, we show that any unclonable encryption scheme generically implies the existence of single-decryptor encryption. The transformation leverages the classic Goldreich-Levin technique [GL89]. The setup and token generation of single-decryptor encryption are the same as the setup and token generation of unclonable encryption. In particular, to generate the unclonable decryption key in a single-decryptor scheme, we sample a long random message x and encrypt x to get $|\text{ct}_x\rangle$, using the unclonable encryption scheme. To encrypt a message m in the single-decryptor scheme, one first sample random coins r (of the same length as m) and let the ciphertext be $(r, \langle r, x \rangle \oplus m)$ together with the key to recover x from the encryption $|\text{ct}_x\rangle$. Since the setup and the token generation algorithms remain the same, and only GenC and Ver need to be modified, we obtain this implication.

2.2.1 From Classical Reductions to Reductions between Cloning Games

We establish the relationship between cloning games using classical reductions in the following steps:

- In the first step, we define a new notion of classical reductions called classical *non-local* reductions. We then show that many natural classical (local) reductions can be upgraded to classical non-local reductions.
- In the second step, we show how to generically lift classical non-local reductions into reductions between cloning games. Specifically, we obtain a reduction between two games \mathcal{G} and \mathcal{G}' such that a cloning adversary for \mathcal{G} can be converted into a cloning adversary for \mathcal{G}' . The transformation only works in the setting when the challenge distribution associated with \mathcal{G} corresponds to an independent challenge distribution.
- In the third and final step, we show that, for any cloning game \mathcal{G} , a cloning adversary succeeding in violating the security of \mathcal{G} with respect to an arbitrary challenge distribution \mathcal{D} can also succeed in violating the security of \mathcal{G} with respect to independent challenge distribution, corresponding to the marginals of \mathcal{D} .

¹⁰Our theorem is more general than what is stated here; refer to Theorem 13 for more details.

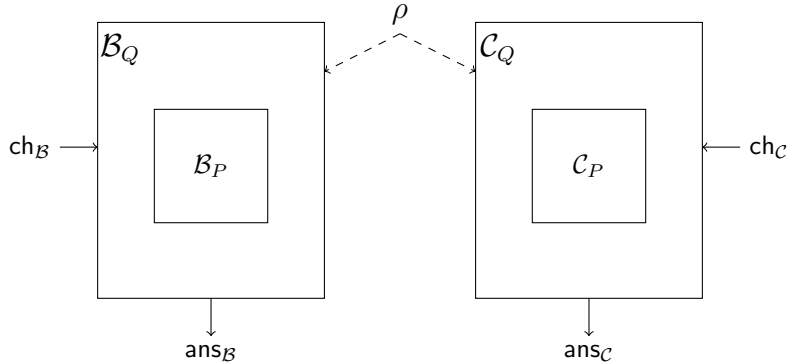
Using the third step, we can now get an improved result in Step 2. Specifically, the reduction between \mathcal{G} and \mathcal{G}' holds even if the challenge distribution associated with \mathcal{G} corresponds to an arbitrary challenge distribution, as long as the marginal distributions for \mathcal{B} and \mathcal{C} remain the same.

We remark that in the third step, we only consider cloning games with trivial success probability to be negligible. Thus, the resulting reductions between cloning games only hold for this setting. Alternately, if we start with a cloning game \mathcal{G} with respect to the independent challenge distribution then we can still apply the first and second step to obtain a reduction between \mathcal{G} and \mathcal{G}' even if the trivial success probability is not negligible.

Step I: From Classical (Local) Reductions to Classical Non-Local Reductions. A reduction transforms a solver for a non-interactive assumption P into a solver for another non-interactive assumption Q . Henceforth, we refer to reductions commonly studied in the literature, as *local* reductions.

In this work, we consider a notion of reductions called *non-local reductions*. First, we need to define non-local solvers. Suppose \mathfrak{A} is a non-interactive assumption (for example, learning with errors). Then, a non-local solver for \mathfrak{A} consists of two algorithms $(\mathcal{B}, \mathcal{C})$ such that each of \mathcal{B} and \mathcal{C} receives samples/challenges $\text{ch}_{\mathcal{B}}, \text{ch}_{\mathcal{C}}$ from \mathfrak{A} and is supposed to solve the samples they receive. Throughout the process, \mathcal{B} and \mathcal{C} cannot speak to each other, although they could have exchanged some common information, denoted by ρ , before receiving $\text{ch}_{\mathcal{B}}, \text{ch}_{\mathcal{C}}$. The samples $\text{ch}_{\mathcal{B}}, \text{ch}_{\mathcal{C}}$ can be arbitrarily correlated. We denote the distribution that samples $(\text{ch}_{\mathcal{B}}, \text{ch}_{\mathcal{C}})$ to be $\mathcal{D}_{\mathfrak{A}}$.

A *non-local* reduction is a transformation that converts a *non-local* \mathcal{D}_P -solver $(\mathcal{B}_P, \mathcal{C}_P)$ for assumption P into a *non-local* \mathcal{D}_Q -solver $(\mathcal{B}_Q, \mathcal{C}_Q)$ for assumption Q , for some challenge distributions \mathcal{D}_P and \mathcal{D}_Q . It turns out that we can lift local reductions into non-local reductions in the classical setting (i.e., when the solvers are classical algorithms) as long as the distribution \mathcal{D}_Q is an independent challenge distribution and the trivial success probability of Q is small (for example, negligible)¹¹.



Step II: Lifting Classical Non-Local Reductions to Reductions Between Cloning Games.

To lift classical non-local reductions to reductions between cloning games, we take inspiration from a recent work by [BBK22] (henceforth, referred to as BBK), who showed a lifting theorem that lifts classical reductions into post-quantum reductions. Suppose we would like to convert a solver for assumption P into a solver for assumption Q . The difficulty in porting classical reductions into

¹¹One example of trivial success probability being large is non-local decision games, where \mathcal{B} and \mathcal{C} try to produce binary answers simultaneously.

post-quantum reductions stems from the fact that the Q -solver could run the P -solver multiple times. Since the state of the P -solver could drastically change from one execution to the other (due to the difficulty of rewinding), potentially, there could no longer be any guarantees from the P -solver after the first execution.

To solve this issue, BBK prove a novel lifting theorem in three steps.

- Persistence theorem: in the first step, they show how to transform a P -solver into another one, where the success probability of the P -solver does not decrease a lot even after executing it multiple times. In other words, the P -solver does not lose the ability to solve instances of P even after multiple executions. In more detail, suppose the P -solver, on input ρ , solved an instance of P with probability p . Then we can convert the P -solver into another one, whose success probability is at least $p - \epsilon$, for some small $\epsilon > 0$, even after multiple executions.

Ideally, we would like the P -solver to be stateless, i.e., it does not know whether it has ever been executed in the past, in order for us to successfully reduce to the problem of solving Q .

- From persistence to memoryless: In the next step, they convert a persistent solver into another one that is indistinguishable from a P -solver that is memoryless. A solver is memoryless if the only thing it can remember is the number of times it has been executed so far.
- From memoryless to stateless: In the final step, they convert the solver from the second step into another solver that is indistinguishable from a stateless solver. Roughly speaking, a stateless solver is one that does not remember any information from one execution to the next.

Our strategy to lift non-local reductions into reductions between cloning games is to use the BBK approach. Similar to their work, we can define the notion of persistent, memoryless, and stateless non-local solvers. Due to some nice structural properties of their transformation, it turns out that the persistent to stateless transformation (the second and third steps above) extends directly to the non-local setting.

Showing the non-local version of their persistence theorem (first bullet above) requires more work. To see why, let us first recall the BBK approach to prove the persistence theorem. They use two procedures, namely value estimation (**ValEst**) and repair (**Repair**) procedures, first defined by [CMSZ21].

- **ValEst** has the guarantee that given an input state ρ and a verification algorithm **Ver**, it outputs a number (probability) p such that $\mathbb{E}[p] = p_{\text{acc}}$ and p_{acc} is the probability that **Ver** accepts ρ . If the output of **ValEst** is p , let the leftover state be ρ_p .
- Suppose we have computed the P -solver on ρ_p . The residual state ρ'_p could be far from ρ_p and more importantly, might not provide any guarantees. We would like to restore the success probability on the residual state ρ'_p obtained after running the P -solver. The procedure **Repair** does just that. It takes as input potentially disturbed state ρ'_p and outputs another state ρ^* such that the success probability on ρ^* is close to the success probability on the original state ρ .

The persistence theorem is proven as follows: each time before computing the P -solver, first run **ValEst** procedure, and then after the execution of the P -solver run the **Repair** procedure. Roughly speaking, by the guarantees of **ValEst** and **Repair**, we have that the underlying P -solver does not

lose its ability to solve the assumption P even after executing it multiple times.

Non-local Persistence Theorem. Before we describe the non-local persistence theorem, we first set up some terminology. We start with a non-local *classical* non-local reduction which reduces a P -non-local solver to a Q -non-local solver. A P -solver consists of $(\mathcal{B}_P, \mathcal{C}_P)$ and is associated with the challenge distribution \mathcal{D}_P . On the other hand, a Q -solver consists of $(\mathcal{B}_Q, \mathcal{C}_Q)$ and is associated with the challenge distribution \mathcal{D}_Q . We want to upgrade this *classical* non-local reduction to the setting when both the P -solver and Q -solver can be quantum. For simplicity, we consider the case when both \mathcal{D}_P and \mathcal{D}_Q are both product distributions.

We now consider a non-local version of the persistence theorem. Informally speaking, we require that the P -non-local solver continues to be a good solver for P even after multiple executions. A natural approach to prove this theorem would be to extend the BBK approach to the non-local setting:

- Before computing the P -non-local solver on its state, first run ValEst procedure.
- After computing on the state, run the Repair procedure.

Unfortunately, we do not know how to execute the above steps. The reason is that the Q -solver itself is non-local and hence, cannot perform any global operations on the state. However, what it can do is to alternately apply value estimation and repair procedures locally. That is, \mathcal{B}_Q (resp., \mathcal{C}_Q) applies the value estimation and repair procedures on \mathcal{B}_P (resp., \mathcal{C}_P). While this sounds promising, this leads to a new issue: we need the guarantee that the P -solver $(\mathcal{B}_P, \mathcal{C}_P)$ is *simultaneously* persistent. Even if we locally apply the procedures above on $(\mathcal{B}_P, \mathcal{C}_P)$ such that both \mathcal{B}_P and \mathcal{C}_P are persistent, this does not mean that they are simultaneously persistent! It could very well be the case when \mathcal{B}_P succeeds, then \mathcal{C}_P does not (or vice versa), but still both of them are persistent.

To address this issue, let us first consider a simple case when the state shared by \mathcal{B}_P and \mathcal{C}_P are unentangled. In this special case, there is a clear relationship between the local and global value estimation and repair procedures. In particular, the following holds:

- Suppose applying $(\text{ValEst} \otimes \text{ValEst})$ on the shared state of $(\mathcal{B}_P, \mathcal{C}_P)$ yields (p_B, p_C) then it holds that $\mathbb{E}[p_B \cdot p_C]$ equals the output of the (global) ValEst on the initial state of $(\mathcal{B}_P, \mathcal{C}_P)$.

Using this, we can relate global persistence to local persistence.

To generalize this to the case when the initial states of $(\mathcal{B}_P, \mathcal{C}_P)$ could be entangled, we look at the specific implementation details of the estimation procedure ValEst by [CMSZ21]. The value estimation procedure ValEst is a sequence of alternating projections, denoted by Π_1, Π_2 , followed by a computational basis measurement determining the success probability p .

Suppose the initial state of $(\mathcal{B}_P, \mathcal{C}_P)$ is in the Hilbert space $\mathcal{H} = \mathcal{H}_B \otimes \mathcal{H}_C$. We decompose both \mathcal{H}_B and \mathcal{H}_C into subspaces that are invariant under the projections Π_1, Π_2 using Jordan's lemma [Jor75]. Therefore, we can rewrite the initial state of $(\mathcal{B}_P, \mathcal{C}_P)$ to be in the span of $\{|\psi_i^B\rangle |\psi_j^C\rangle\}$, where $\{|\psi_i^B\rangle\}$ (resp., $\{|\psi_j^C\rangle\}$) is in the corresponding Jordan subspaces of \mathcal{H}_B (resp., \mathcal{H}_C).

Using an observation made by [CMSZ21], we can think of ValEst as first performing a Jordan subspace measurement (that projects the state onto one of the Jordan subspaces) followed by performing a sequence of alternating measurements Π_1, Π_2 . In other words, we can think of applying value estimation locally, i.e., $(\text{ValEst} \otimes \text{ValEst})$, as first performing the Jordan subspace measurement to obtain a joint state $|\psi_i^B\rangle |\psi_j^C\rangle$, for some i, j , followed by alternating measurements. Notice that once we apply the Jordan subspace measurement, the states become unentangled! Thus, we reduce to the above simple case, and the rest of the analysis follows.

Step III: Relating Challenge Distributions: From Independent to Identical. In Step II, in order to be able to run the value estimation and the repair procedures locally, it was crucial that the underlying P -solver was defined for an *independent* challenge distribution. It would be interesting to generalize to the case when the underlying challenge distribution is arbitrary. For this overview, we focus on the case when the challenge distribution is identical, although the proof generalizes to arbitrary challenge distributions as well. Specifically, we demonstrate a reduction from a cloning game \mathcal{G} satisfying unclonable security with respect to independent challenges to \mathcal{G} satisfying unclonable security with respect to identical challenges. For the reduction to work, we crucially use the fact that the trivial success probability in both the games is negligible. An interesting point to note here is that the reduction does not change the description of the game.

We give an overview of our reduction. Let $|\sigma\rangle_{\mathbf{BC}}$ be the (entangled) quantum state shared by Bob and Charlie (the two non-local quantum adversaries) after Alice's (the splitting adversary) stage. We additionally define projections Π_r^B and Π_r^C for every possible random coins r :

Π_r^B : Run Bob on its own register $\sigma[\mathbf{B}]$ with the challenge corresponding to random coins r , project onto Bob's output being accepted and uncompute;

Π_r^C : Run Charlie on its own register $\sigma[\mathbf{C}]$ with the challenge corresponding to random coins r , project onto Charlie's output being accepted and uncompute;

By definition, the success probability in the independent challenge case is:

$$\delta = \text{Tr} \left[\left(\frac{1}{|R|} \sum_r \Pi_r^B \right) \otimes \left(\frac{1}{|R|} \sum_r \Pi_r^C \right) |\sigma\rangle \langle \sigma| \right], \quad (1)$$

where R is the random coin space.

Since $\Pi^B := \frac{1}{|R|} \sum_r \Pi_r^B$ is a POVM, let $\{|\phi_p\rangle\}_{p \in \mathbb{R}}$ be the set of eigenvectors with eigenvalues $p \in [0, 1]$ ¹². Similarly, let $\{|\psi_q\rangle\}_{q \in \mathbb{R}}$ be the set of eigenvectors with eigenvalues $q \in [0, 1]$ for $\Pi^C := \frac{1}{|R|} \sum_r \Pi_r^C$. Therefore, we can always write $|\sigma\rangle_{\mathbf{BC}}$ under the bases $\{|\phi_p\rangle\}$ and $\{|\psi_q\rangle\}$ ¹³:

$$|\sigma\rangle = \sum_{p,q} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle.$$

From the above decomposition of $|\sigma\rangle$ and Equation (1), we have $\delta = \sum_{p,q} |\alpha_{p,q}|^2 pq$.

Let $\eta \in [0, 1]$ be a threshold we will pick later. The quantum state can be written as the summation of three terms:

$$|\sigma\rangle = \sum_{q < \eta} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle + \sum_{p < \eta, q > \eta} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle + \sum_{p > \eta, q > \eta} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle.$$

We denote the first term by $|\sigma_C\rangle$, indicating that Charlie's success probability is bounded by η ; the second term by $|\sigma_B\rangle$, indicating that Bob's success probability is bounded by η ; and the last term by $|\rho\rangle$, none of the probabilities is below η . Thus, $|\sigma\rangle = |\sigma_C\rangle + |\sigma_B\rangle + |\rho\rangle$.

¹²There can be multiple eigenvectors with the same eigenvalues. In the overview, we assume that eigenvalues are unique.

¹³There is a one-to-one mapping between $\{|\phi_p\rangle, |\psi_q\rangle\}$ and the vectors $\{|\psi_i^B\rangle, |\psi_j^C\rangle\}$ defined in the Jordan's lemma.

First, we note that the success probability when executed on the state $|\sigma_C\rangle + |\sigma_B\rangle$ is at most η under both independent challenges and identical challenges. However, $\frac{|\rho\rangle}{\|\rho\rangle\|^2}$ could be such that the success probability when executed on this state maybe large (even as large as 1). In the next step, we show that although ρ may have a large probability under identical challenges, $\|\rho\rangle\|^2$ is relatively small. Because $\delta := \sum_{p,q} |\alpha_{p,q}|^2 pq$, we have:

$$\|\rho\rangle\|^2 = \sum_{p>\eta, q>\eta} |\alpha_{p,q}|^2 \implies \|\rho\rangle\|^2 \leq \delta/\eta^2.$$

Therefore, the success probability of $|\sigma\rangle$ under identical challenges is:

$$\begin{aligned} \delta &= \frac{1}{|R|} \sum_r \|\Pi_r^B \otimes \Pi_r^C |\sigma\rangle\|^2 \\ &\leq \frac{3}{|R|} \sum_r \left(\|\Pi_r^B \otimes \Pi_r^C |\sigma_C\rangle\|^2 + \|\Pi_r^B \otimes \Pi_r^C |\sigma_B\rangle\|^2 + \|\rho\|^2 \right) \\ &\leq 3(\eta + \delta/\eta^2). \end{aligned}$$

By picking $\eta = \delta^{1/3}$, the resulting probability is $6 \cdot \delta^{1/3}$. Specifically, if δ is negligible then so is the resulting quantity.

2.3 Part II: Generalizing Existing Results

2.3.1 Unclonable search to unclonable indistinguishability

Our first focus is a cloning game with unclonable search security (a concrete example is unclonable encryption with standard unclonable security) whose distribution \mathcal{D} is uniform over all possible messages. We show a generic reduction that turns such a game into another cloning game with unclonable indistinguishability security whose underlying distribution is \mathcal{D}_{m_0, m_1} for any m_0, m_1 in the message space, in the quantum random oracle model (QROM, introduced by [BDF⁺11]). Since unclonable encryption with standard unclonable security exists [BL20], this gives a direct corollary for unclonable encryption with unclonable indistinguishability security in the QROM, from BB84/Wiesner states, improving the previous result by [AKL⁺22].

Unclonable Security for High-Entropy Message Distributions. As a first step in the reduction, we make the following observation. Suppose we start with a cloning game satisfying unclonable security. If the message is sampled from a high min-entropy distribution instead of being sampled from random, unclonable security still holds. In particular, we prove that when m is sampled from a source with min-entropy h instead of from a uniform source, its unclonable search security will be $2^h \cdot \delta$; where δ is the unclonable search security under the uniform message distribution. For instance, if m is sampled uniformly at random from a set S then by appropriately choosing $|S|$ (for example, it is exponential sized), we can prove that $2^h \delta$ is still negligible and thus establish its augmented unclonable security.

As a concrete example, we obtain the following corollary: the unclonable encryption with standard unclonable security in [BL20] also satisfies this augmented unclonable security. In other words, even if P_m is provided as oracle, it will not help Bob and Charlie to simultaneously recover m .

Augmented Security. Next, we first define stronger unclonable search security, which we call *augmented unclonable security*. The cloning game is defined in the same way, except now all attackers have oracle access to a point function $P_m(\cdot)$, which outputs 1 if and only if the input equals to m , where m is the message used to generate the token given to the adversary. We claim that the definition of unclonable search security can be generically upgraded to obtain augmented security.

Our first observation is that, we can enlarge the set of all accepting inputs of $P_m(\cdot)$ (originally only m) to a large random set S consisting of m , with its security staying roughly the same. More concretely, S will be defined as an exponentially large (but negligibly small compared to the number of all possible messages) set consisting of a single m , and the rest are random messages. As P_m and P_S only differ on exponentially many but sparse random inputs, query-bounded adversaries can not distinguish which oracle is given.

Next, the augmented unclonable security is then argued under a random message m and oracle access to P_S . In this case, we can instead think of an alternate but equivalent process of sampling m : first sample an exponentially large random set S then sample a message m is uniformly at random from S . After changing the sampling order, we can argue that even if the adversary is given the description of the set S , unclonable security still holds. This holds from our earlier observation that unclonable security holds even if the message is sampled from a high min-entropy distribution.

From Augmented Security to Unclonable-Indistinguishability Security. Finally, we show that starting from a cloning game satisfying augmented unclonability property \mathcal{G} , we can obtain a game \mathcal{G}' satisfying unclonable indistinguishability property. The token generation of \mathcal{G}' on input a message m , first samples a long random message x , runs the token generation of \mathcal{G} on x and then outputs this token along with $H(x) \oplus m$, where H is a hash function. In the proof of security, H is treated as a random hash function that the adversary has oracle access to.

To prove unclonable indistinguishability, we look at the state $|\psi\rangle_{\mathcal{BC}}$ output by Alice, where Alice has oracle access to H punctured at the input x . For the sake of the proof, we treat the hash function both Bob and Charlie have access to, separately. We use $H_{\mathcal{B}}$ to denote the hash function Bob has access to and $H_{\mathcal{C}}$ to denote the hash function Charlie has access to. Correspondingly, we can define the POVM $\Pi^{\mathcal{B}}$ that runs \mathcal{B} with oracle access to $H_{\mathcal{B}}$ that is programmed on x to output 0 or 1 with equal probability, projects onto the output being correct and then uncomputes. Similarly, we define $\Pi^{\mathcal{C}}$ as well. In order to make sure we can implement $\Pi^{\mathcal{B}}$ and $\Pi^{\mathcal{C}}$ efficiently, we give the adversary oracle access to $P_x(\cdot)$.

Let $\{|\phi_p\rangle\}_{p \in \mathbb{R}}$ be the set of eigenvectors with respect to $\Pi^{\mathcal{B}}$ with eigenvalues $p \in [0, 1]$. Similarly, let $\{|\psi_q\rangle\}_{q \in \mathbb{R}}$ be the set of eigenvectors with eigenvalues $q \in [0, 1]$ with respect to $\Pi^{\mathcal{C}}$. We can then rewrite $|\psi\rangle_{\mathcal{BC}}$ in terms of the eigenbases of $\Pi^{\mathcal{B}}$ and $\Pi^{\mathcal{C}}$.

$$|\psi\rangle = \sum_{q \approx 0.5} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle + \sum_{p \approx 0.5, |q-0.5| \gg 0} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle + \sum_{|p-0.5| \gg 0, |q-0.5| \gg 0} \alpha_{p,q} |\phi_p\rangle |\psi_q\rangle.$$

Once we do this, we show the following:

- $\sum_{p,q} |\alpha_{p,q}|^2$ is negligible. We show this by reducing to the unclonability property.
- Once we show bullet 1, we can then show that Bob and Charlie cannot simultaneously succeed with probability significantly better than 0.5 in the case when it receives as input $|\psi\rangle$. The analysis of this was shown in [AKL⁺22].

2.4 Generalized Cloning Games

Another way we can extend the notion of cloning games is by allowing asymmetric verification for \mathcal{B} and \mathcal{C} by allowing different algorithms $(\text{GenC}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{B}}, \text{Ver}_{\mathcal{C}})$ in the verification phase. We call this more general class of games *asymmetric cloning games*.

2.4.1 Deletion Games

As a special case, we define *deletion games*, in which $\text{GenC}_{\mathcal{B}}$ outputs no challenge, so that \mathcal{B} is effectively supposed to produce a classical certificate of deletion. In this context, we can define search games based on the algorithms $(\text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$, with \mathcal{C} understood to be the party tasked to perform the intended functionality of the token. With these modifications, unclonable search security and unclonable indistinguishable security are defined the same as before. We show how to go from the former to the latter using the Quantum Goldreich-Levin Lemma¹⁴ [AC02]. In order to achieve unclonable search security, we show that the construction¹⁵ of [BI20] satisfies unclonable search security using monogamy-of-entanglement games [TFKW13], which have been commonly used in unclonable cryptography [BL20, CLLZ21, CV21]. Specifically, we show that the success probability of any adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ in the following game is exponentially small in λ :

- $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ prepares a bipartite state ρ shared between \mathcal{A} and the referee Ref. \mathcal{A} splits the state between \mathcal{B} and \mathcal{C} .
- Ref makes a measurement in basis H^θ for a random $\theta \in \{0, 1\}^\lambda$
- \mathcal{B} outputs $x_{\mathcal{B}}$. \mathcal{C} receives θ and outputs $x_{\mathcal{C}}$.
- $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins if $x_{\mathcal{C}} = x$ and $x_{\mathcal{B},i} = x_i$ whenever $\theta_i = 1$.

This suffices due to a well-known reduction from cloning games to monogamy-of-entanglement games using EPR pairs. Thus, we provide a different method to achieve information theoretic encryption with certified deletion. Although our method is incomparable to previous methods for achieving the same result [BI20, BK22], we believe our approach may be more intuitive for some readers.

2.4.2 Relating Search and Decision Games

We give one more transformation, which starts with a cloning search game and ends up with a cloning decision game. The first one uses augmented security above and applies it to the construction of [AKL⁺22] for copy-protection in the QROM. We generalize the proof for a class of cloning games, and as a special case, we achieve copy-protection for point functions using BB84 states in the QROM. Since the ideas employed in this part are similar to Section 2.3, we omit the details.

¹⁴Unlike our result on single-decryptor encryption, which asks for the usual, stronger property of unclonability, here we do not need the simultaneous version of the Goldreich-Levin Lemma because we are in the weaker, certified deletion setting.

¹⁵A simplified version of it without additional properties. The authors show in [BI20] that the construction already satisfies the stronger notion of unclonable indistinguishable security, yet the proof is more involved.

2.5 Future Directions

Relationship Between Challenge Distributions for Decision Games. In this work, we show that when a cloning game has negligible soundness (similar to a search game) with an independent distribution, the cloning game with the corresponding identical distribution is also secure. We leverage this theorem to many applications, including copy-protection and single-decryptor encryption schemes. However, this theorem does not apply when the soundness is a constant. An interesting open problem is to generalize the result to the case with constant security error (for example, unclonable-indistinguishability). Generalizing this result would present a pathway towards achieving unclonable encryption scheme with unclonable-indistinguishability in the plain model, that is currently open.

Removing Random Oracles from BB84-based Constructions. Another approach to obtain unclonable encryption with unclonable-indistinguishability in the plain model is to remove the need for random oracle in the Corollary 1. Currently, the random oracle is essential, and we do not know how to get rid of it. Still, we believe that removing QROM in the theorem statement is a promising direction and will help us understand the relationship between various unclonable primitives and the computational assumptions they need.

Domain Extension. Suppose we have a cloning game for messages of n bits. Is it possible to generically transform this into another cloning game for $2n$ bits? Naive repetition does not work well with cloning games and hence, it would be interesting to come up with interesting techniques for domain extension. One application of this is domain extension for unclonable encryption. Suppose we have an encryption scheme that can encrypt n bits and we would like to transform this into a different scheme encrypting $2n$ bits. It would also be interesting to study domain extension for the challenge space as well. This would have implications to domain extension for single-decryptor encryption.

Generalizing the Non-Local Lifting Theorem Our non-local lifting theorem Theorem 22 is restricted in that the classical reductions need to be black-box and non-adaptive. These restrictions propagate from the work of [BBK22], and removing them will allow for more classical reductions to be lifted to the quantum setting.

2.6 Organization

In Section 3, we define basic terminology and give relevant results from previous work that are used in this work. In Section 4, we introduce the cloning games framework and provide formal definitions. In Section 5, we describe how to lift classical reductions to quantum reductions in the non-local setting, as well as applications to cloning games. In Sections 6 and 7, we show how to use our framework to generalize existing results in unclonable cryptography via generic transformations between cloning search/decision games. In Section 8, we show how to extend our framework to capture asymmetric unclonable primitives, and give an alternative construction of unclonable encryption with certified deletion.

Acknowledgments. PA and FK are supported by a gift from Cisco.

3 Preliminaries

We denote the security parameter by λ . We say a classical algorithm \mathcal{A} is efficient if it is a probabilistic polynomial-time (PPT) algorithm. We write $\mathcal{A}(x; r)$ to mean that \mathcal{A} runs on input x with random coins $r \in \{0, 1\}^{\text{poly}(\lambda)}$. We say a quantum algorithm \mathcal{A} is efficient if it runs in quantum polynomial time (QPT).

We write $\mathcal{A}^{\mathcal{O}}$ to denote an oracle algorithm \mathcal{A} that makes queries to an oracle \mathcal{O} . If \mathcal{A} is a quantum algorithm and \mathcal{O} is a classical oracle, then it is understood that \mathcal{A} can make superposition queries. We call \mathcal{A} *query-bounded* if it makes only polynomially many queries.

We denote by \mathcal{U}_X the uniform distribution over a set X . $\text{negl}(\cdot)$ denotes a negligible function and $\text{poly}(\cdot)$ is a function upper-bounded by a polynomial. We say that an event occurs with *overwhelming probability* if it happens with probability $1 - \text{negl}(\lambda)$.

We denote by $P_x(\cdot)$ the point function, defined as

$$P_x(x') := \begin{cases} 1, & x' = x \\ 0, & x' \neq x \end{cases}.$$

Trivial Guess with Auxiliary Information: We define

$$\text{OPT}(X | Y) := \sum_y \Pr[Y = y] \cdot \max_x \Pr[X = x | Y = y].$$

It is the optimal probability of guessing the value of variable X after observing the value of variable Y . Clearly, $\text{OPT}(X | Y) \geq \text{OPT}(X | 0) = 2^{-H_{\min}(X)}$, where H_{\min} denotes the min-entropy, defined as follows:

$$H_{\min}(X) := -\log_2 \left[\max_x \Pr[X = x] \right]$$

3.1 Quantum Computing Basics & Query Bounds

Given Hilbert space \mathcal{H} , $\mathcal{D}(\mathcal{H})$ denotes the set of density operators on \mathcal{H} . We write \mathcal{H}_X to denote the Hilbert space associated with a quantum register X . We write $\rho[X]$ to denote the X register of a quantum state ρ . Given two quantum states ρ, σ , we denote the (normalized) trace distance between them by

$$\text{TD}(\rho, \sigma) := \frac{1}{2} \|\rho - \sigma\|_{\text{tr}}.$$

We say that two states ρ, σ are δ -close if $\text{TD}(\rho, \sigma) \leq \delta$.

A k -outcome (general) *quantum measurement* is a k -tuple of quantum operators $\mathcal{M} = (M_i)_{i \in [k]}$ satisfying $\sum_{i \in [k]} M_i^\dagger M_i = I$. The probability of obtaining outcome i after measuring a mixed state ρ is given by $\text{Tr}(M_i \rho M_i^\dagger)$, and the post-measurement state is given by $M_i \rho M_i^\dagger / \text{Tr}(M_i \rho M_i^\dagger)$. If M_i is a projector for each $i \in [k]$, then we call \mathcal{M} a projective measurement. A *positive operator valued measurement* (POVM) is k -tuple of positive semi-definite operators $(E_i)_{i \in [k]}$. It is used to describe quantum measurement when the post-measurement state is irrelevant in the context. The

probability of outcome i equals $\text{Tr}(E_i\rho)$. Every general quantum measurement defines a POVM by setting $E_i = M_i^\dagger M_i$.

A common technique in quantum computation is *uncomputing* [BBBV97]. A quantum algorithm \mathcal{A} can be modeled as a unitary U acting on some hilbert space \mathcal{H} , followed by a measurement on output registers without loss of generality. We refer to applying U as running \mathcal{A} *coherently*, and to uncomputing \mathcal{A} as applying U^\dagger on \mathcal{H} .

Quantum Random Oracle Model In the quantum random oracle model (QROM), we assume that there exists a random function $H : \{0, 1\}^m \rightarrow \{0, 1\}^n$, where $m = \text{poly}(\lambda)$, $n = \text{poly}(\lambda)$, such that all parties (honest and malicious) have oracle access to the unitary \mathcal{O}^H , defined as $\mathcal{O}^H |x\rangle |y\rangle = |x\rangle |y \oplus H(x)\rangle$. Such a random oracle H can be efficiently simulated on the fly for a query-bounded adversary [Zha19], or if a query bound t was known beforehand, it can be simulated efficiently by using a $2t$ -wise independent hash function in lieu of the random oracle [Zha12].

The following theorem, paraphrased from [BBBV97], will be used for reprogramming oracles without adversarial detection on inputs that are not queried with large weight:

Theorem 4 ([BBBV97]). *Let \mathcal{A} be an oracle algorithm which makes at most T oracle queries to a function $H : \{0, 1\}^m \rightarrow \{0, 1\}^n$. Define $|\phi_i\rangle$ as the global state after \mathcal{A} makes i queries, and $W_y(|\phi_i\rangle)$ as the sum of squared amplitudes in $|\phi_i\rangle$ of terms in which \mathcal{A} queries H on input y . Let $\epsilon > 0$ and let $F \subseteq \{0, 1, \dots, T-1\} \times \{0, 1\}^m$ be a set of time-input pairs such that $\sum_{(i,y) \in F} W_y(|\phi_i\rangle) \leq \epsilon^2/T$.*

For $i \in \{0, 1, \dots, T-1\}$, let H'_i be an oracle obtained by reprogramming H on inputs in $\{y \in \{0, 1\}^m : (i, y) \in F\}$ to arbitrary outputs. Let $|\phi'_T\rangle$ be the global state after \mathcal{A} is run with oracle H'_i on the i th query (instead of H). Then, $\text{TD}(|\phi_T\rangle, |\phi'_T\rangle) \leq \epsilon/2$.

Note that the theorem can be straightforwardly generalized to mixed states by convexity.

We will typically use Theorem 4 by contrapositive, i.e., if a query-bounded adversary \mathcal{A} outputs states with non-negligible trace distance when given oracle access to H or H' , then \mathcal{A} must have non-negligible query weight on inputs for which H and H' differ. Hence one can extract such input by measuring a random query. We list a particular corollary of interest below.

Corollary 2 (Subset Hiding). *Let $S \subset \{0, 1\}^\lambda$. Let $m > |S|$ such that $m/(2^\lambda - |S|)$ is negligible. Let T_S^m be the set of all $T \subset \{0, 1\}^\lambda$ such that $|T| = m + |S|$ and $S \subset T$. Then, for any query-bounded algorithm \mathcal{A} , we have*

$$\left| \Pr \left[1 \leftarrow \mathcal{A}^{P_S(\cdot)}(1^\lambda) \right] - \Pr \left[1 \leftarrow \mathcal{A}^{P_T(\cdot)}(1^\lambda) : T \stackrel{\$}{\leftarrow} T_S^m \right] \right| \leq \text{negl}(\lambda). \quad (2)$$

Proof. Suppose \mathcal{A} violates eq. (2). By Theorem 4, $\mathcal{A}^{P_S(\cdot)}$ must have non-negligible query weight on $T \setminus S$, which is a random subset of $\{0, 1\}^\lambda \setminus S$ of size m . By measuring a random query of \mathcal{A} , a query-bounded adversary $\mathcal{A}^{P_S(\cdot)}$ can output a value $t \in T \setminus S$ with non-negligible probability. However, $T \setminus S$ is information-theoretically hidden from $\mathcal{A}^{P_S(\cdot)}$, this probability is upper-bounded by $m/(2^\lambda - |S|)$, a contradiction. \square

3.2 Jordan's Lemma

We state Jordan's Lemma, paraphrased from [CMSZ21].

Lemma 5 ([Jor75]). Let \mathcal{H} be a Hilbert space and let Π_A, Π_B be two orthogonal projectors on \mathcal{H} . There exists an orthogonal decomposition $\mathcal{H} = \bigoplus_j S_j$ into one-dimensional or two-dimensional subspaces, where each S_j is invariant under both Π_A and Π_B . Moreover:

- If $\dim S_j = 1$, then Π_A and Π_B act as rank-0 or rank-1 projectors on S_j .
- If $\dim S_j = 2$, then there exist distinct orthogonal bases $\{|v_{j,1}^A\rangle, |v_{j,0}^A\rangle\}$ and $\{|v_{j,1}^B\rangle, |v_{j,0}^B\rangle\}$ of S_j , where $\Pi_A = \sum_j |v_{j,1}^A\rangle\langle v_{j,1}^A|$ and $\Pi_B = \sum_j |v_{j,1}^B\rangle\langle v_{j,1}^B|$.

We will denote by $\Pi^{\text{Jor}} = \{\Pi_j^{\text{Jor}}\}_j$ the projective measurement that measure the index j of S_j , i.e. $\Pi_j^{\text{Jor}} = |v_{j,0}^A\rangle\langle v_{j,0}^A| + |v_{j,1}^A\rangle\langle v_{j,1}^A| = |v_{j,0}^B\rangle\langle v_{j,0}^B| + |v_{j,1}^B\rangle\langle v_{j,1}^B|$. An important fact is that Π^{Jor} commutes with both the projective measurements $(\Pi_A, I - \Pi_A)$ and $(\Pi_B, I - \Pi_B)$.

We cite the following lemma from [AKL⁺22], which is a corollary of Jordan's Lemma.

Lemma 6. For any two projectors Π_0, Π_1 and $w \in [0, 1]$, let $|\phi_0\rangle$ and $|\phi_1\rangle$ be two eigenvectors of $w\Pi_0 + (1-w)\Pi_1$ with eigenvalues λ_0, λ_1 . If $\lambda_0 + \lambda_1 \neq 1$ and $\lambda_0 \neq \lambda_1$, then

$$\langle \phi_0 | \Pi_0 | \phi_1 \rangle = \langle \phi_0 | \Pi_1 | \phi_1 \rangle = 0.$$

3.3 Applications of Jordan's Lemma

In this section, we state two applications of Jordan's lemma [Jor75] which use the techniques of [Zha20], and date back to the QMA amplification techniques of [MW05]. While they use similar techniques, the applications are different in their syntax and flavor. Threshold measurement (first application) aims to project a state onto eigenstates with eigenvalues larger than some threshold, whereas state repair (second application) involves estimating the average eigenvalue, and restoring the state to another state with similar value after a collapsing measurement occurs.

Threshold Measurement We cite the following theorems regarding how to test the success probability of a quantum token from [AKL⁺22], originally due to [Zha20].

Theorem 7 (Inefficient Threshold Measurement). Let $\mathcal{P} = (P, Q)$ be a binary outcome POVM. Let P have eigenbasis $\{|\psi_i\rangle\}$ with eigenvalues $\{\lambda_i\}$. Then, for every $\gamma \in (0, 1)$ there exists a projective measurement $\mathcal{E}_\gamma = (E_{\leq \gamma}, E_{> \gamma})$ such that:

- (1) $E_{\leq \gamma}$ projects a quantum state into the subspace spanned by $\{|\psi_i\rangle\}$ whose eigenvalues λ_i satisfy $\lambda_i \leq \gamma$;
- (2) $E_{> \gamma}$ projects a quantum state into the subspace spanned by $\{|\psi_i\rangle\}$ whose eigenvalues λ_i satisfy $\lambda_i > \gamma$.

Similarly, for every $\gamma \in (0, 1/2)$, there exists a projective measurement $\mathcal{E}'_\gamma = (\tilde{E}_{\leq \gamma}, \tilde{E}_{> \gamma})$ such that:

- (1) $\tilde{E}_{\leq \gamma}$ projects a quantum state into the subspace spanned by $\{|\psi_i\rangle\}$ whose eigenvalues λ_i satisfy $|\lambda_i - \frac{1}{2}| \leq \gamma$;
- (2) $\tilde{E}_{> \gamma}$ projects a quantum state into the subspace spanned by $\{|\psi_i\rangle\}$ whose eigenvalues λ_i satisfy $|\lambda_i - \frac{1}{2}| > \gamma$.

Theorem 8 (Efficient Threshold Measurement). *Let $\mathcal{P}_b = (P_b, Q_b)$ be a binary outcome POVM over Hilbert space \mathcal{H}_b that is a mixture of projective measurements for $b \in \{1, 2\}$. Let P_b have eigenbasis $\{|\psi_i^b\rangle\}$ with eigenvalues $\{\lambda_i^b\}$. For every $\gamma_1, \gamma_2 \in (0, 1)$, $0 < \epsilon < \min(\gamma_1/2, \gamma_2/2, 1 - \gamma_1, 1 - \gamma_2)$ and $\delta > 0$, there exist efficient binary-outcome quantum algorithms, interpreted as the POVM element corresponding to outcome 1, $\text{ATI}_{\mathcal{P}_b, \gamma}^{\epsilon, \delta}$ such that for every quantum program $\rho \in \mathcal{D}(\mathcal{H}_1) \otimes \mathcal{D}(\mathcal{H}_2)$ the following are true about the product algorithm $\text{ATI}_{\mathcal{P}_1, \gamma_1}^{\epsilon, \delta} \otimes \text{ATI}_{\mathcal{P}_2, \gamma_2}^{\epsilon, \delta}$:*

(0) *Let $(E_{\leq \gamma}^b, E_{> \gamma}^b)$ be the inefficient threshold measurement in Theorem 7 for \mathcal{H}_b .*

(1) *The probability of measuring 1 on both registers satisfies*

$$\text{Tr} \left[\left(\text{ATI}_{\mathcal{P}_1, \gamma_1}^{\epsilon, \delta} \otimes \text{ATI}_{\mathcal{P}_2, \gamma_2}^{\epsilon, \delta} \right) \rho \right] \geq \text{Tr} \left[(E_{> \gamma_1 + \epsilon}^1 \otimes E_{> \gamma_2 + \epsilon}^2) \cdot \rho \right] - 2\delta.$$

(2) *The post-measurement state ρ' after getting outcome (1,1) is 4δ -close to a state in the support of $\{|\psi_i^1\rangle |\psi_j^2\rangle\}$ such that $\lambda_i^1 > \gamma_1 - 2\epsilon$ and $\lambda_j^2 > \gamma_2 - 2\epsilon$.*

(3) *The running time of the algorithm is polynomial in the running time of P_1, P_2 , $1/\epsilon$ and $\log(1/\delta)$.*

Theorem 9 (Efficient Symmetric Threshold Measurement). *Let $\mathcal{P}_b = (P_b, Q_b)$ be a binary outcome POVM over Hilbert space \mathcal{H}_b that is a mixture of projective measurements for $b \in \{1, 2\}$. Let P_b have eigenbasis $\{|\psi_i^b\rangle\}$ with eigenvalues $\{\lambda_i^b\}$. For every $\gamma_1, \gamma_2 \in (0, 1/2)$, $0 < \epsilon < \min(\gamma_1/2, \gamma_2/2)$, and $\delta > 0$, there exist efficient binary-outcome quantum algorithms, interpreted as the POVM element corresponding to outcome 1, $\text{SATI}_{\mathcal{P}_b, \gamma}^{\epsilon, \delta}$ such that for every quantum program $\rho \in \mathcal{D}(\mathcal{H}_1) \otimes \mathcal{D}(\mathcal{H}_2)$ the following are true about the product algorithm $\text{SATI}_{\mathcal{P}_1, \gamma_1}^{\epsilon, \delta} \otimes \text{SATI}_{\mathcal{P}_2, \gamma_2}^{\epsilon, \delta}$:*

(0) *Let $(\tilde{E}_{\leq \gamma_b}^b, \tilde{E}_{> \gamma_b}^b)$ be the inefficient threshold measurement in Theorem 7 for \mathcal{H}_b .*

(1) *The probability of measuring 1 on both registers satisfies*

$$\text{Tr} \left[\left(\text{SATI}_{\mathcal{P}_1, \gamma_1}^{\epsilon, \delta} \otimes \text{SATI}_{\mathcal{P}_2, \gamma_2}^{\epsilon, \delta} \right) \rho \right] \geq \text{Tr} \left[\left(\tilde{E}_{> \gamma_1 + \epsilon}^1 \otimes \tilde{E}_{> \gamma_2 + \epsilon}^2 \right) \cdot \rho \right] - 2\delta.$$

(2) *The post-measurement state ρ' after getting outcome (1,1) is 4δ -close to a state in the support of $\{|\psi_i^1\rangle |\psi_j^2\rangle\}$ such that $|\lambda_i^1 - 1/2| > \gamma_1 - 2\epsilon$ and $|\lambda_j^2 - 1/2| > \gamma_2 - 2\epsilon$.*

(3) *The running time of the algorithm is polynomial in the running time of P_1, P_2 , $1/\epsilon$ and $\log(1/\delta)$.*

State Repair. We state the state-repairing results (Lemma 4.9 and Lemma 4.10) from [CMSZ21] below to be used for achieving persistence. We adapt the formulation by [BBK22] with some additional modifications: unlike [BBK22], we need some additional structural properties of value estimation and state repair procedures of [CMSZ21] formalized in the third bullet below.

Lemma 10. *There exist efficient quantum algorithms ValEst and Repair with the following syntax and guarantees:*

- $\text{ValEst}_{V,\mathcal{A}}(\rho, 1^{1/\varepsilon})$ takes as input the description of a verifier algorithm $V : \{0, 1\}^d \times \{0, 1\}^n \rightarrow \{0, 1\}$, a quantum algorithm \mathcal{A} , a quantum state ρ , and an accuracy parameter ε . It outputs a quantum state ρ^* and a value $p^* \in [0, 1]$.
- $\text{Repair}_{V,\mathcal{A},\Pi}(\sigma, y, p, 1^{1/\varepsilon}, 1^k)$ takes as input a verifier algorithm V , a quantum algorithm \mathcal{A} , a k -outcome quantum measurement Π with outcomes \mathcal{Y} , an outcome $y \in \mathcal{Y}$, a value $p \in [0, 1]$, and an accuracy parameter ε . It outputs a quantum state σ^* .

1. **Value Estimation:**

$$\mathbb{E}_{(\rho^*, p^*) \leftarrow \text{ValEst}_{V,\mathcal{A}}(\rho, 1^{1/\varepsilon})} [p^*] = \Pr \left[V(y; r) = 1 : \begin{array}{l} r \xleftarrow{\$} \{0, 1\}^d \\ y \leftarrow \mathcal{A}(\rho, r) \end{array} \right]. \quad (3)$$

2. **Almost-Projective Estimation:** For any $\varepsilon \geq \varepsilon' > 0$,

$$\Pr \left[|p^* - p^{**}| \geq \varepsilon : \begin{array}{l} (\rho^*, p^*) \leftarrow \text{ValEst}_{V,\mathcal{A}}(\rho, 1^{1/\varepsilon}) \\ (\rho^{**}, p^{**}) \leftarrow \text{ValEst}_{V,\mathcal{A}}(\rho^*, 1^{1/\varepsilon'}) \end{array} \right] \leq \varepsilon.$$

3. **2-Projection Implementation:** For every $(V, \mathcal{A}, \varepsilon)$, there exist projective measurements $\mathcal{M}_0 = (\Pi_A, I - \Pi_A)$ and $\mathcal{M}_1 = (\Pi_B, I - \Pi_B)$ and classical deterministic algorithms f, g such that the execution of $\text{ValEst}_{V,\mathcal{A}}(\rho, 1^{1/\varepsilon})$ does the following:

- Initialize an empty database $L = \emptyset$.
- Initialize an auxiliary register as $|\psi_0\rangle$, so that the current mixed state is $\rho \otimes |\psi_0\rangle\langle\psi_0|$.
- For $1 \leq i \leq \text{poly}(|V|, |\mathcal{A}|, 1/\varepsilon)$:
 - Compute $b = f(i, L) \in \{0, 1, \perp\}$
 - If $b \in \{0, 1\}$, apply \mathcal{M}_b and obtain outcome ℓ_i . Set $L = L \sqcup \{\ell_i\}$, where \sqcup denotes disjoint union.
 - If $b = \perp$, end the loop. Output the current residual state ρ^* and the value $p^* = g(L)$.

Furthermore, the measurements satisfy the following:

- $\mathcal{M}_0(\rho')$ can be described as:
 - Pick $r \xleftarrow{\$} \{0, 1\}^d$.
 - Compute $y \leftarrow \mathcal{A}(\rho', r)$ coherently.
 - Measure if $V(y; r) = 1$.
 - Uncompute.
- $\Pi_B = I \otimes |\psi_0\rangle\langle\psi_0|$.

4. **State Repair:** For any $\varepsilon > 0$,

$$\Pr \left[|p^* - p^{**}| \geq \varepsilon : \begin{array}{l} (\rho^*, p^*) \leftarrow \text{ValEst}_{V,\mathcal{A}}(\rho, 1^{1/\varepsilon}) \\ (\sigma, y) \leftarrow \Pi(\rho^*) \\ \sigma^* \leftarrow \text{Repair}_{V,\mathcal{A},\Pi}(\sigma, y, p, 1^{1/\varepsilon}, 1^k) \\ (\rho^{**}, p^{**}) \leftarrow \text{ValEst}_{V,\mathcal{A}}(\sigma^*, 1^{1/\varepsilon}) \end{array} \right] \leq \varepsilon. \quad (4)$$

3.4 Unlearnable Distributions

Definition 1 (Unlearnability). A distribution $\mathcal{D} = \mathcal{D}(\lambda)$ is called unlearnable if for any query-bounded adversary $\mathcal{A}^{P_y(\cdot)}$ with oracle access to $P_y(\cdot)$, we have:

$$\Pr \left[y' = y : y' \leftarrow \mathcal{A}^{P_y(\cdot)}(1^\lambda) \right] \leq \text{negl}(\lambda).$$

4 Cloning Games - Definitions

We would like to capture all cryptographic games where the adversary needs to clone a particular functionality of a given quantum token. The quantum token could be a copy-protected program, signature token, unclonable ciphertext, unclonable decryption key, or any quantum state that serves some functionality which could only be used by one party at a given time. We start off with basic definitions and give generalizations in Section 4.4.

Definition 2 (Cloning Game). A cloning game consists of a tuple of efficient algorithms $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$:

- **Key Generation:** $\text{Setup}(1^\lambda)$ is a PPT algorithm which takes as input a security parameter 1^λ in unary. It outputs a secret key $\text{sk} \in \{0, 1\}^*$. We will assume without loss of generality¹⁶ that sk always contains the security parameter 1^λ .
- **Token Generation:** $\text{GenT}(\text{sk}, m)$ is a QPT algorithm that takes as input a secret key sk and a message $m \in \{0, 1\}^*$. It outputs a quantum token ρ .
- **Challenge Generation:** $\text{GenC}(\text{sk}, m)$ takes as input a secret key sk and a message m . It outputs a classical challenge $\text{ch} \in \{0, 1\}^*$.
- **Verification:** $\text{Ver}(\text{sk}, m, \text{ch}, \text{ans})$ takes as input a secret key sk , a message m , a challenge ch , and an answer ans . It outputs either 0 (reject) or 1 (accept).

4.1 Correctness

Before we talk about cloning experiments, we should specify what property of a quantum token ρ we would like to be unclonable. Intuitively, the property will be captured by the ability to honestly pass verification using the token ρ . This brings us to the definition of correctness for a cloning game:

Definition 3 (Correctness). Let $\delta : \mathbb{Z}^+ \rightarrow [0, 1]$. We say that \mathcal{G} has δ -correctness if there exists an efficient quantum algorithm $\mathcal{A}_{\mathcal{G}}$ such that for all messages $m \in \mathcal{M}$:

$$\Pr \left[\text{Ver}(\text{sk}, m, \text{ch}, \text{ans}) = 1 : \begin{array}{l} \text{sk} \leftarrow \text{Setup}(1^\lambda) \\ \rho \leftarrow \text{GenT}(\text{sk}, m) \\ \text{ch} \leftarrow \text{GenC}(\text{sk}, m) \\ \text{ans} \leftarrow \mathcal{A}_{\mathcal{G}}(\rho, \text{ch}) \end{array} \right] \geq \delta(\lambda)$$

If $\delta = 1$ (or $\delta(\lambda) = 1 - \text{negl}(\lambda)$), we say \mathcal{G} has perfect (or statistical) correctness. In this work, we will mainly focus on statistically correct cloning games.

¹⁶This is in order to simplify the notation for the rest of the algorithms. We will sometimes make this inclusion explicit, and other times it is understood implicitly.

Note: In the correctness definition above, \mathcal{A}_G should be considered an honest user of the primitive.

4.2 Special Types of Cloning Games

Next, we define some special cases, with terminology borrowed from classical security notions.

Definition 4 (Cloning Search Game). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning game such that $\text{Ver}(\text{sk}, m, \text{ch}, \text{ans})$ accepts if and only if $\text{ans} = m$. Then, \mathcal{G} is called a cloning search game.*

Definition 5 (Cloning Decision Game). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning game such that the answer ans taken as input by Ver is one bit, i.e. $\text{ans} \in \{0, 1\}$. Then, \mathcal{G} is called a cloning decision game.*

We additionally define the notion of a *cloning encryption game* when we discuss unclonable encryption in Section 4.5.2.

4.3 Security

Cloning Experiment. We will define notions of security for a cloning game in terms of a security experiment. Given a token ρ , an adversary should not be able to generate two (possibly entangled) quantum tokens which can simultaneously pass verification. We will formalize this intuition below.

Definition 6 (Cloning Experiment). *A cloning experiment, denoted by $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}$, is a security game played between a referee Ref and a cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. It is parameterized by a cloning game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ and a distribution \mathcal{D} over the message space \mathcal{M} . The experiment is described as follows:*

- **Setup Phase:**
 - All parties get a security parameter 1^λ as input.
 - Ref samples a message $m \leftarrow \mathcal{D}$.
 - Ref computes $\text{sk} \leftarrow \text{Setup}(1^\lambda)$ and $\rho \leftarrow \text{GenT}(\text{sk}, m)$.
 - Ref sends ρ to \mathcal{A} .
- **Splitting Phase:**
 - \mathcal{A} computes a bipartite state ρ' over registers B, C .
 - \mathcal{A} sends $\rho'[B]$ to \mathcal{B} and $\rho'[C]$ to \mathcal{C} .
- **Challenge Phase:**
 - Ref independently samples $\text{ch}_B, \text{ch}_C \leftarrow \text{GenC}(\text{sk}, m)$.
 - Ref sends ch_B to \mathcal{B} and ch_C to \mathcal{C} .
 - \mathcal{B} and \mathcal{C} send back answers ans_B and ans_C , respectively.
 - Ref computes bits $b_B \leftarrow \text{Ver}(\text{sk}, m, \text{ch}_B, \text{ans}_B)$ and $b_C \leftarrow \text{Ver}(\text{sk}, m, \text{ch}_C, \text{ans}_C)$.
 - The outcome of the game is denoted by $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$, which equals 1 if $b_B = b_C = 1$, indicating that the adversary has won, and 0 otherwise, indicating that the adversary has lost.

Trivial Success. As a baseline for unclonable security, we will consider *trivial* attacks that do not require any cloning operation. The best we can hope is that such attacks are optimal, hence the definitions below.

Definition 7 (Trivial Cloning Attack). *We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is a trivial cloning attack against a cloning experiment $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}$ if \mathcal{A} upon receiving a token ρ , sends the product state $|\perp\rangle\langle\perp| \otimes \rho$ to \mathcal{B} and \mathcal{C} . In other words, only \mathcal{C} gets the token ρ . We denote by $\text{TRIV}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}})$ the set of trivial attacks against $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}$.*

Remark 1. *Note that due to the symmetry between \mathcal{B} and \mathcal{C} , the definition of trivial cloning attack could be equivalently defined so that only \mathcal{B} gets the token ρ .*

Definition 8 (Trivial Success Probability for Cloning Games). *We define the trivial success probability of a cloning experiment $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}$ as*

$$p^{\text{triv}}(\mathcal{G}, \mathcal{D}) := \sup_{(\mathcal{A}, \mathcal{B}, \mathcal{C}) \in \text{TRIV}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}})} \Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right].$$

Unclonable Security. We present the security definition of cloning games below.

Definition 9 (Unclonable Security). *Let \mathcal{G} be a cloning game, \mathcal{D} be a distribution over the message space \mathcal{M} , and $\varepsilon : \mathbb{Z}^+ \rightarrow [0, 1]$. We say that \mathcal{G} has $(\mathcal{D}, \varepsilon)$ unclonable security if for all QPT cloning adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ we have:*

$$\Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq p^{\text{triv}}(\mathcal{G}, \mathcal{D}) + \varepsilon(\lambda).$$

If $|\mathcal{M}| = 1$, we will simply write ε unclonable security.

4.3.1 Security for Search Games

For the special case of search games, we consider two definitions below.

Definition 10 (Unclonable Search Security). *If \mathcal{G} is a cloning search game with $(\mathcal{D}, \varepsilon)$ unclonable security, we additionally say that \mathcal{G} has $(\mathcal{D}, \varepsilon)$ unclonable search security.*

Remark 2. *Note that even though the definitions above are valid for any distribution \mathcal{D} , to get meaningful security one needs to choose \mathcal{D} appropriately for the context. For instance, if the cloning game \mathcal{G} represents copy-protection for point functions, it is appropriate to pick \mathcal{D} in a balanced way so that the trivial success probability $p^{\text{triv}}(\mathcal{G}, \mathcal{D})$ is bounded away from 1. As long as this is the case, $(\mathcal{D}, \varepsilon)$ unclonable security (for small ε) is non-trivial¹⁷ in the sense that it is classically impossible and it uses the power of no-cloning. On the other hand, when $p^{\text{triv}}(\mathcal{G}, \mathcal{D}) \approx 1$ unclonable security becomes trivial and achieved by uninteresting constructions including classical games.*

Definition 11 (Unclonable Indistinguishable Security). *Let \mathcal{D}_{m_0, m_1} denote the distribution that outputs messages m_0 and m_1 with probability $1/2$ each. We say that a search game \mathcal{G} has ε unclonable indistinguishable security if it has $(\mathcal{D}_{m_0, m_1}, \varepsilon)$ unclonable search security for any pair of messages $m_0, m_1 \in \mathcal{M}$.*

In Definitions 9 to 11, if $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is not required to be efficient, then we say that \mathcal{G} has *information theoretic* unclonable (search/indistinguishable) security.

¹⁷We assume statistical correctness here.

4.4 Extended Definitions

Stateful Games. Most cloning games can be captured by Definition 2. Yet, some cloning games are *stateful* in the sense that verification takes as additional input the random coins used in the challenge generation. With this in mind, we define a generalization of the cloning game below, highlighting the differences in blue. Throughout this section, we will assume that GenC is a classical algorithm.¹⁸

Definition 12 (Stateful Cloning Game). *A stateful cloning game consists of a tuple of efficient algorithms $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$.*

- **Key Generation:** $\text{Setup}(1^\lambda)$ takes as input a security parameter 1^λ in unary. It outputs a secret key sk .
- **Token Generation:** $\text{GenT}(\text{sk}, m)$ takes as input a secret key sk and a message m . It outputs a quantum token ρ .
- **Challenge Generation:** $\text{GenC}(\text{sk}, m; r_{\text{GenC}})$ takes as input a secret key sk and a message m .
It outputs a classical challenge ch .
- **Verification:** $\text{Ver}(\text{sk}, m, \text{ch}, \text{ans}, r_{\text{GenC}})$ takes as input a secret key sk , a message m , a challenge ch , an answer ans , random coins r_{GenC} used by GenC when generating ch . It outputs either 0 (reject) or 1 (accept).

When we talk about cloning games, we will always implicitly mean stateful cloning games. In fact, all of our results easily generalize to stateful cloning games. However, we will omit r_{GenC} above and use the syntax in Definition 2 when appropriate, for simplicity.

Security Against Correlated Distributions. When we defined security in Section 4.3, we assumed that \mathcal{B} and \mathcal{C} in the security experiment receive independently generated challenges. We will define security more broadly and refer to the aforementioned definition as *independent-challenge security*. For simplicity, we will assume that challenge generation is classical, i.e., GenC is a PPT algorithm, which is true for all the primitives considered in this work.

Definition 13 (Challenge Extension). *Let GenC be a challenge generation algorithm that takes as input randomness from $\mathcal{R} = \{0, 1\}^{\text{poly}(\lambda)}$. We say that $\widetilde{\text{GenC}}$ is an extension of GenC if:*

- On input a secret key sk and a message m , it outputs a pair of random strings $(r_{\mathcal{B}}, r_{\mathcal{C}}) \in \mathcal{R}^2$.
- For any (sk, m) , if $(r_{\mathcal{B}}, r_{\mathcal{C}}) \leftarrow \widetilde{\text{GenC}}(\text{sk}, m)$, then the marginal distributions of both $r_{\mathcal{B}}$ and $r_{\mathcal{C}}$ are equal to $\mathcal{U}_{\mathcal{R}}$.

We will also refer to $\widetilde{\text{GenC}}$ as an extension of a game \mathcal{G} whenever GenC is the challenge generation algorithm for \mathcal{G} . We sometimes will omit and not specify the extension $\widetilde{\text{GenC}}$, in which case it is either clear from the context or assumed to be $\widetilde{\text{GenC}} = \text{GenC}_{\text{ind}}$ by default (see Definition 15).

¹⁸If GenC is quantum, one can similarly define statefulness by having GenC output some random coins that it sampled during its execution. One would need this because unlike classical algorithms, the randomness of a quantum algorithm can inherently result from collapsing measurements and hence cannot be modeled as an auxiliary random input string.

Extended Cloning Experiment. Next, we give a more general definition of unclonable security, highlighting the differences to the corresponding definitions in Section 4.3 in blue.

Definition 14 (Extended Cloning Experiment). An (extended) cloning experiment, denoted by $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$, is a security game played between a referee Ref and a cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. It is parameterized by a cloning game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$, a distribution \mathcal{D} over the message space \mathcal{M} , and an extension $\widetilde{\text{GenC}}$ of GenC. The experiment is described as follows:

- **Setup Phase:**

- All parties get a security parameter 1^λ as input.
- Ref samples a message $m \leftarrow \mathcal{D}$.
- Ref computes $\text{sk} \leftarrow \text{Setup}(1^\lambda)$ and $\rho \leftarrow \text{GenT}(\text{sk}, m)$.
- Ref sends ρ to \mathcal{A} .

- **Splitting Phase:**

- \mathcal{A} computes a bipartite state ρ' over registers B, C .
- \mathcal{A} sends $\rho'[B]$ to \mathcal{B} and $\rho'[C]$ to \mathcal{C} .

- **Challenge Phase:**

- Ref samples $(r_{\mathcal{B}}, r_{\mathcal{C}}) \leftarrow \widetilde{\text{GenC}}(\text{sk}, m)$ and then computes $\text{ch}_{\mathcal{B}} = \text{GenC}(\text{sk}, m; r_{\mathcal{B}})$, $\text{ch}_{\mathcal{C}} = \text{GenC}(\text{sk}, m; r_{\mathcal{C}})$.
- Ref sends $\text{ch}_{\mathcal{B}}$ to \mathcal{B} and $\text{ch}_{\mathcal{C}}$ to \mathcal{C} .
- \mathcal{B} and \mathcal{C} send back answers $\text{ans}_{\mathcal{B}}$ and $\text{ans}_{\mathcal{C}}$, respectively.
- Ref computes bits $b_{\mathcal{B}} \leftarrow \text{Ver}(\text{sk}, m, \text{ch}_{\mathcal{B}}, \text{ans}_{\mathcal{B}}, r_{\mathcal{B}})$ and $b_{\mathcal{C}} \leftarrow \text{Ver}(\text{sk}, m, \text{ch}_{\mathcal{C}}, \text{ans}_{\mathcal{C}}, r_{\mathcal{C}})$.
- The outcome of the game is denoted by $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$, which equals 1 if $b_{\mathcal{B}} = b_{\mathcal{C}} = 1$, indicating that the adversary has won, and 0 otherwise, indicating that the adversary has lost.

Next, we discuss two important special cases for the extension $\widetilde{\text{GenC}}$. In the first case, \mathcal{B} and \mathcal{C} get the same challenge, whereas in the second case, they get independently generated challenges. Keep in mind that the second case is the default assumption when we do not mention extensions in Section 4.3.

Definition 15 (Identical/Independent-Challenge Cloning Experiment). Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ and define the following extensions of GenC, which takes randomness from the set \mathcal{R} :

- $\text{GenC}_{\text{id}}(\text{sk}, m)$ samples $r \leftarrow \mathcal{U}_{\mathcal{R}}$ and outputs (r, r) .
- $\text{GenC}_{\text{ind}}(\text{sk}, m)$ computes $r_{\mathcal{B}}, r_{\mathcal{C}} \leftarrow \mathcal{U}_{\mathcal{R}}$ independently. It outputs $(r_{\mathcal{B}}, r_{\mathcal{C}})$.

Then, we call the cloning experiments $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\text{GenC}_{\text{id}}}$ and $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\text{GenC}_{\text{ind}}}$ an identical-challenge cloning experiment or an independent-challenge cloning experiment, respectively.

Trivial Success Probability. We will slightly modify the definition of trivial attacks to account for the potential asymmetry introduced by a challenge extension.

Definition 16 ((Extended) Trivial Cloning Attack). *We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is a \mathcal{B} -trivial cloning attack against a cloning experiment $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$ if \mathcal{A} upon receiving a token ρ , sends the product state $|\perp\rangle\langle\perp| \otimes \rho$ to \mathcal{B} and \mathcal{C} . In other words, only \mathcal{C} gets the token ρ . We denote by $\text{TRIV}_{\mathcal{B}}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}})$ the set of \mathcal{B} -trivial cloning attacks against $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$. We similarly define $\text{TRIV}_{\mathcal{C}}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}})$ as the set of \mathcal{C} -trivial attacks.*

Finally, we define

$$\text{TRIV}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}) := \text{TRIV}_{\mathcal{B}}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}) \cup \text{TRIV}_{\mathcal{C}}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}})$$

as the set of trivial cloning attacks against $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$.

Definition 17 ((Extended) Trivial Success Probability for Cloning Games). *We define the \mathcal{B} -trivial success probability of a cloning experiment $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$ as*

$$p_{\mathcal{B}}^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) := \sup_{(\mathcal{A}, \mathcal{B}, \mathcal{C}) \in \text{TRIV}_{\mathcal{B}}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}})} \Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}((\mathcal{A}, \mathcal{B}, \mathcal{C})) \right].$$

We similarly define $p_{\mathcal{C}}^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}})$ as the \mathcal{C} -trivial success probability of $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$. Accordingly, we define the trivial success probability of $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$ as

$$p^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) = \max \left(p_{\mathcal{B}}^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}), p_{\mathcal{C}}^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) \right).$$

Remark 3. *One may consider mixtures of \mathcal{B} -trivial and \mathcal{C} -trivial cloning attacks as trivial, but such attacks cannot do better than trivial cloning attacks by convexity.*

Definition 18 ((Extended) Unclonable Security). *Let \mathcal{G} be a cloning game with extension $\widetilde{\text{GenC}}$, \mathcal{D} be a distribution over the message space \mathcal{M} , and $\varepsilon : \mathbb{Z}^+ \rightarrow [0, 1]$. We say that \mathcal{G} has $(\mathcal{D}, \varepsilon, \widetilde{\text{GenC}})$ unclonable security if for all QPT cloning adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ we have*

$$\Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq p^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) + \varepsilon(\lambda).$$

We define $(\mathcal{D}, \varepsilon, \widetilde{\text{GenC}})$ unclonable search security and $(\varepsilon, \widetilde{\text{GenC}})$ unclonable indistinguishable security similarly. If $\widetilde{\text{GenC}} = \text{GenC}_{\text{ind}}$ (resp., $\widetilde{\text{GenC}} = \text{GenC}_{\text{id}}$), then we say \mathcal{G} has $(\mathcal{D}, \varepsilon)$ independent-challenge (resp., identical-challenge) unclonable security. Likewise, information theoretic security is defined by removing the efficiency requirement from $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ as before.

A good number of cloning games of interest will have the following additional property.

Definition 19 (Evasiveness). *A cloning game \mathcal{G} is called \mathcal{D} -evasive if $p^{\text{triv}}(\mathcal{G}, \mathcal{D})$ is a negligible function in λ .*

Note that the definition above is independent of the extension $\widetilde{\text{GenC}}$ for statistically correct games, i.e. $p^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}})$ is negligible for any extension $\widetilde{\text{GenC}}$. We keep the definition simple given that we only use it for statistically correct games.

Asymmetric Verification. Another way to generalize the cloning games is to allow asymmetric verification for \mathcal{B} and \mathcal{C} , which we will define and discuss in Section 8.

4.5 Examples

In this section, we demonstrate the comprehensiveness of cloning games by casting popular unclonable primitives as cloning games. We restrict our attention to primitives with symmetric verification, and those with asymmetric verification, such as secure software leasing or certified deletion, require a slightly more general syntax, which will be defined in Section 8.

4.5.1 Copy-Protection

Let \mathcal{F} be the class of functions of the form $f : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized implicitly by a security parameter λ , and let \mathcal{D} be a distribution over \mathcal{F} . A copy-protection scheme for \mathcal{D} is a pair of efficient algorithms (CP, Eval) :

- $\text{CP}(1^\lambda, d_f)$ takes as input description d_f of a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ and outputs a copy-protected quantum program ρ_f .
- $\text{Eval}(1^\lambda, \rho_f, x)$ takes as input a quantum program ρ_f and an input $x \in \mathcal{X}$. It outputs a value $y \in \mathcal{Y}$.

(CP, Eval) defines a cloning game $\mathcal{G}_{\text{CP}}^{\mathcal{D}'} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ for any family of distributions $\mathcal{D}' = \left(\mathcal{D}'_f\right)_{f \in \mathcal{F}}$ over \mathcal{F} as follows. Note that, \mathcal{D}' defines a distribution on challenge inputs, therefore, it specifies GenC .

- The message space \mathcal{M} is the set of function descriptions d_f for all $f \in \mathcal{F}$.
- $\text{Setup}(1^\lambda)$ outputs $\text{sk} = 1^\lambda$, i.e. there is no secret key.
- $\text{GenT}(\text{sk}, m)$ parses the input as $m = d_f$, then it computes $\rho_f \leftarrow \text{CP}(1^\lambda, d_f)$ and outputs ρ_f .
- $\text{GenC}(\text{sk}, m)$ parses $m = d_f$ and samples input $x \leftarrow \mathcal{D}'_f$.¹⁹
- $\text{Ver}(\text{sk}, m, \text{ch}, \text{ans})$ parses $m = d_f$, $\text{ch} = x$. It accepts if and only if $\text{ans} = f(x)$.

Correctness: We require that \mathcal{G}_{CP} has statistical correctness, to ensure that the copy-protected program is reusable. More specifically, $\mathcal{A}_{\mathcal{G}_{\text{CP}}}(\rho, \text{ch})$ runs $\text{Eval}(\rho, \text{ch})$.²⁰

Security: We consider a game-based definition of copy-protection, first defined by [CMP20, BJJ⁺21]. We say that (CP, Eval) is secure for a class of distributions \mathcal{D}' if $\mathcal{G}_{\text{CP}}^{\mathcal{D}'}$ has $(\mathcal{D}, \varepsilon)$ unclonable security. For optimal security, we require ε to be negligible.

¹⁹Here we make the natural assumption that correctness and security are defined with respect to the same distribution \mathcal{D}'_f . Intuitively, the scheme should protect against cloning the functionality of the honest evaluator.

²⁰Note that this captures the average-input correctness as opposed to per-input correctness.

4.5.2 Unclonable Encryption

Below, we define unclonable encryption [BL20] as a cloning game. We focus our attention to one-time secret-key setting, in which case unclonable encryption is synonymous with cloning encryption games defined in Definition 20. It is known in the literature that construction in this simple setting can be generically lifted to achieve unclonable encryption with additional properties, such as public-key encryption [AK21]. We note, however, that unclonable encryption with such properties can still be expressed as a cloning game by modifying the syntax of a cloning encryption game. We state the correspondence below, which is easy to verify.

Definition 20 (Cloning Encryption Game). *A cloning search game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ is called a cloning encryption game if $\text{GenC}(\text{sk}, m)$ outputs sk with probability 1 for all (sk, m) .*

Fact 11 (Informal). *An unclonable encryption scheme for a message space \mathcal{M} exists with unclonable (unclonable indistinguishable) security if and only if a cloning encryption game \mathcal{G} for \mathcal{M} with unclonable (unclonable indistinguishable) security exists.*

There are two types of security we will consider for unclonable encryption: (1) $(\mathcal{U}_{\mathcal{M}}, \varepsilon)$ unclonable security and (2) ε unclonable indistinguishable security. These security definitions together with δ -correctness are on par with the original definitions of [BL20].²¹ Note that since GenC is deterministic, it has a unique extension.

A particular case of interest is adapted²² from the *conjugate encryption* of [BL20] and uses Wiesner (BB84) states:

Definition 21 (BB84 Cloning Game). $\mathcal{G}_{\text{BB84}} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ is a cloning encryption game with message space $\mathcal{M} = \{0, 1\}^\lambda$, defined as follows:

- $\text{Setup}(1^\lambda)$ outputs $\theta \xleftarrow{\$} \{0, 1\}^\lambda$
- $\text{GenT}(\theta, m)$ takes as input $\theta, m \in \{0, 1\}^\lambda$ and outputs $\rho = |m^\theta\rangle\langle m^\theta|$, where $|m^\theta\rangle = H^\theta |m\rangle$
- GenC and Ver are defined as part of a cloning encryption game.

Lemma 12 (Security of BB84 Cloning Game [BL20]). *The game $\mathcal{G}_{\text{BB84}}$ above has $(\mathcal{U}_{\mathcal{M}}, |\mathcal{M}|^{-\delta})$ unclonable security for some constant $\delta > 0$.*

4.5.3 Single Decryptor Encryption

We define single-decryptor encryption as a tuple of efficient algorithms $(\text{Gen}, \text{GenT}', \text{Enc}, \text{Dec})$, adapted from the definition of (secret-key) single-decryptor encryption (with honestly generated keys) in [GZ20]:

1. $\text{Gen}(1^\lambda)$ takes as input a security parameter and outputs a classical secret key sk .
2. $\text{GenT}'(sk)$ takes as input a classical secret key and it outputs a quantum decryption key ρ_{dk} .
3. $\text{Enc}(sk, x)$ takes as input a secret key and a classical message. It outputs a classical ciphertext ct .

²¹Although (2) was defined in a slightly different way in [BL20], the difference is inconsequential, and our version has been used in follow-up works such as [AKL⁺22]. We also mention that [BL20] considered perfect correctness.

²²We omit a classical one-time-pad on the message m , which is irrelevant for the purposes of unclonability.

4. $\text{Dec}(\rho, ct)$ takes as input a quantum decryption key and a classical ciphertext. It outputs a classical message x' .

$(\text{Gen}, \text{GenT}', \text{Enc}, \text{Dec})$ defines a stateful cloning game $\mathcal{G}_{\text{SDE}}^{\mathcal{D}_{\mathcal{X}}} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$, parameterized by a distribution $\mathcal{D}_{\mathcal{X}}$, where \mathcal{X} is the set of classical messages encrypted by this scheme, as follows:

- $\text{Setup}(1^\lambda)$ runs $sk \leftarrow \text{Gen}(1^\lambda)$ and outputs $sk = sk$.
- There is no message, i.e. $m = \perp$.
- $\text{GenT}(sk, m)$ computes $\rho_{dk} \leftarrow \text{GenT}'(sk)$ and outputs ρ_{dk} .
- $\text{GenC}(sk, m; r_{\text{GenC}})$ samples $x \xleftarrow{\$} \mathcal{X}$ using random coins r_{GenC} . It outputs $c \leftarrow \text{Enc}(sk, x)$.
- $\text{Ver}(sk, m, \text{ch}, \text{ans}, r_{\text{GenC}})$ computes x as above using r_{GenC} . Then it accepts if and only if $\text{ans} = x$.

Correctness. We say that $(\text{Gen}, \text{GenT}', \text{Enc}, \text{Dec})$ has correctness if $\mathcal{G}_{\text{SDE}}^{\mathcal{D}_{\mathcal{X}}}$ has perfect correctness for any distribution $\mathcal{D}_{\mathcal{X}}$. More specifically, $\mathcal{A}_{\mathcal{G}_{\text{SDE}}}(\rho_{dk}, \text{ch})$ runs $\text{Dec}(\rho_{dk}, \text{ch})$.

Security. We say that $(\text{Gen}, \text{GenT}', \text{Enc}, \text{Dec})$ has ε unclonable security if $\mathcal{G}_{\text{SDE}}^{\mathcal{U}_{\mathcal{X}}}$ has ε unclonable security²³.

In other words, the ability to decrypt a random classical message is the unclonable property of the quantum decryption key. For optimal security, we require that ε is negligible.

4.5.4 Quantum Money

Next, we give examples of cloning games with quantum verification. We focus on quantum money, first introduced by Wiesner [Wie83]. We consider a public-key variant of quantum money considered by [AC12, Zha17]. We note that the description below can be suitably adapted to case private-key quantum money as a cloning game. A public-key quantum money scheme is a tuple of efficient algorithms $(\text{Gen}, \text{Mint}, \text{VerToken})$:

- $\text{Gen}(1^\lambda)$ takes as input a security parameter and outputs a public-secret key pair (pk, sk) .
- $\text{Mint}(sk)$ takes as input a secret key and outputs a classical serial number s and a quantum banknote ρ_s .
- $\text{VerToken}(pk, s, \rho)$ takes as input a public key, a serial number, and a quantum state. It outputs 0 (reject) or 1 (accept).

$(\text{Gen}, \text{Mint}, \text{VerToken})$ defines a cloning game $\mathcal{G}_{\text{QM}} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ as follows:

- We set $m = \perp$, i.e. there is no message.
- $\text{Setup}(1^\lambda)$ runs $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$ and outputs $sk = (pk, sk)$

²³We omit the message distribution due to the lack of message.

- $\text{GenT}(\text{sk}, m)$ parses the input as $\text{sk} = (pk, sk)$, runs $(s, \rho_s) \leftarrow \text{Mint}(sk)$, and outputs $\rho \otimes |s\rangle\langle s| \otimes |pk\rangle\langle pk|$.
- $\text{GenC}(\text{sk}, m)$ outputs $\text{ch} = \perp$, i.e. no challenge.
- $\text{Ver}(\text{sk}, m, \text{ch}, \sigma_{\text{ans}})$ parses the input as $\text{sk} = (pk, sk)$ and outputs $b \leftarrow \text{VerToken}(pk, \sigma_{\text{ans}})$

Correctness. We say that the quantum money scheme has correctness if \mathcal{G}_{QM} has statistical correctness. Note that $\mathcal{A}_{\mathcal{G}_{\text{QM}}}$ can simply output the quantum banknote it receives to satisfy Definition 3 (correctness). Furthermore, it can be assumed without loss of generality that the optimal $\mathcal{A}_{\mathcal{G}_{\text{QM}}}$ acts as identity (i.e. outputs ρ as is) since there is no challenge. Therefore, this fully captures the usual definition of correctness for quantum money schemes.

Security. We say that the quantum money scheme is secure if \mathcal{G}_{QM} has ε unclonable security. For optimal security, we require that ε is negligible. Note that unclonable security as we defined only gives 1-to-2 unclonability, but it can be generalized to k -to- $k + 1$ unclonability. Alternatively, one can define a quantum-money mini scheme in our framework, which is necessary and sufficient for constructing public-key quantum money [AC12].

5 Constructive Post-Quantum Reductions: The Non-Local Setting

We present a new toolkit to understand the relationship between different cloning games. We first define a notion of non-local classical reductions. Roughly speaking, a classical non-local reduction transforms a non-local classical algorithm²⁴, solving a problem P to a non-local classical algorithm solving a problem Q . The reason we consider non-local classical reductions is that it turns out to be simpler to come up with non-local reductions in the classical setting. We then show how to generically upgrade some classes of classical non-local reductions to quantum non-local reductions, the analogous notion for quantum non-local algorithms. The resulting quantum non-local reductions are useful in analyzing the relationship between different cloning games.

5.1 Definitions

In this section, we borrow definitions from Section 3 of [BBK22] and adapt them to the non-local setting²⁵.

Extension. Similar to Definition 13, we define the notion of an extension over random coins in this context.

Definition 22 (d -extension). *We say that a distribution $\tilde{\mathcal{D}}$ over $\{0, 1\}^d \times \{0, 1\}^d$ is a d -extension if for $(r_{\mathcal{B}}, r_{\mathcal{C}}) \leftarrow \tilde{\mathcal{D}}$, the marginal distributions of both $r_{\mathcal{B}}$ and $r_{\mathcal{C}}$ are $\mathcal{U}_{\{0,1\}^d}$.*

²⁴Refer to Section 3 for the definition of a non-local algorithm.

²⁵For simplicity, we omit explicit purification of quantum solvers (see Definition 3.5), as we will make nearly black-box use of the results of [BBK22].

5.1.1 Classical Assumptions

The first two definitions below are commonly used in the cryptography literature. Note that we use GenC, Ver to denote the algorithms in order to point out the connection between a non-interactive assumption and the challenge phase of a cloning game (see Definitions 2 and 6).

Definition 23 (Non-Interactive Assumption). *A non-interactive assumption $\mathfrak{A} = (d, n, m, \text{GenC}, \text{Ver}, c)$ is associated with polynomials $d(\lambda), n(\lambda), m(\lambda)$ and a tuple $P = (\text{GenC}, \text{Ver}, c)$ with the following syntax. Here, GenC and Ver are classical algorithms, and $c : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\}$ is the assumption's threshold.*

- *Challenge generator, $\text{GenC}(1^\lambda; r)$: on input security parameter λ and random coins $r \in \{0, 1\}^d$, outputs a challenge $x \in \{0, 1\}^n$.*
- *Verifier, $\text{Ver}(1^\lambda, r, y)$: on input security parameter λ , random coins $r \in \{0, 1\}^d$, and answer $y \in \{0, 1\}^m$, outputs 1 (accept) or 0 (reject).*

We will sometimes use P and \mathfrak{A} interchangeably, with the understanding that the polynomials d, n, m are implicit. We say that P is evasive if c is a negligible function.

Another important property is called *verifiably polynomial image*, and informally it requires that it is possible to verify if a solution lies in a polynomial-size superset of valid solutions without the random coins r of GenC .

Definition 24 (Verifiably Polynomial Image). *A non-interactive assumption $\mathfrak{A} = (d, n, m, \text{GenC}, \text{Ver}, c)$ has a verifiably polynomial image if there exists a polynomial $k(\lambda)$ and an efficient verifier K such that for every $x \in \{0, 1\}^n$, the set $Y_x := |\{y : K(1^\lambda, x, y) = 1\}|$ has size at most k and for any valid challenge $x = G(1^\lambda; r)$ and answer $\text{Ver}(1^\lambda, r, y) = 1$, it holds that $y \in Y_x$.*

Remark 4. *At a high level, the above definition states that a non-interactive assumption with verifiably polynomial range has a public verification algorithm K that accepts all the solutions also accepted by Ver (and possibly, more) such that for any x , the set of strings accepted by K should be of polynomial size.*

5.1.2 Solvers

Local Solvers. A local solver for a non-interactive assumption P , which we call a P -solver, is a pair $\mathcal{A} = (\widetilde{\text{alice}}, \text{state}_0)$, where $\widetilde{\mathcal{A}}$ is an algorithm which takes as input a challenge $x \in \{0, 1\}^n$ and outputs an answer $y \in \{0, 1\}^m$, whereas state_0 is an initial state. We define the value and advantage of a local solver below:

Definition 25 (Value and Advantage of a Local Solver (Definition 3.3 in [BBK22])). *Let $\mathfrak{A} = (d, n, m, \text{GenC}, \text{Ver}, c)$ be a non-interactive assumption, with $P = (\text{GenC}, \text{Ver}, c)$, and let $\mathcal{A} = (\widetilde{\mathcal{A}}, \text{state}_0)$ be a P -solver. We define the (one-shot) value and advantage of \mathcal{A} , parameterized by the security parameter λ , as*

$$\text{val}_P[\mathcal{A}] = \Pr \left[\text{Ver}(1^\lambda, r, y) = 1 \left| \begin{array}{l} r \xleftarrow{\$} \{0, 1\}^d \\ x = \text{GenC}(1^\lambda; r) \\ y \leftarrow \widetilde{\mathcal{A}}(1^\lambda, x, \text{state}_0) \end{array} \right. \right], \quad \text{adv}_P[\mathcal{A}] = |\text{val}_P[\mathcal{A}] - c|.$$

Non-Local Solvers. Below, we adapt Definition 25 to the non-local setting. A non-local adversary consists of two (possibly inefficient) algorithms $(\mathcal{B}, \mathcal{C})$, where both \mathcal{B} and \mathcal{C} are given (possibly correlated) instances from a non-interactive assumption. Each adversary is expected to solve the instance they get. Note that both \mathcal{B} and \mathcal{C} are not allowed to communicate with each other. In the end, the adversary wins if both \mathcal{B} and \mathcal{C} win. In general, for algorithms \mathcal{B} and \mathcal{C} acting on separate registers, we call the algorithm $\mathcal{B} \otimes \mathcal{C}$ a non-local algorithm²⁶. We consider the non-uniform setting, where both \mathcal{B} and \mathcal{C} could share some auxiliary information (either a string or a quantum state, depending on whether the adversary is classical or quantum) in the very beginning.

In order to define reductions, we need to define stateful solvers. Suppose in a reduction, we use the solver for a problem P to design a solver for problem Q . Suppose the Q solver runs the P solver multiple times. In the classical setting, the P solver could be stateless, whereas in the quantum setting the P solver is inherently stateful and thus, we need to define stateful solvers appropriately below. Formally, we adapt Definition 3.4 from [BBK22] to the non-local setting.

Definition 26 (Non-Local Stateful Solvers: Syntax). *Let P be a non-interactive assumption with d -bit random coins.*

Let $\ell = \ell(\lambda)$, $\ell_{\mathcal{B}} = \ell_{\mathcal{B}}(\lambda)$, $\ell_{\mathcal{C}} = \ell_{\mathcal{C}}(\lambda)$ be functions such that $\ell_{\mathcal{B}}(\lambda) + \ell_{\mathcal{C}}(\lambda) = \ell(\lambda)$ and let $\tilde{\mathcal{D}}$ be a d -extension. An $(\ell, \ell_{\mathcal{B}}, \ell_{\mathcal{C}})$ -stateful non-local $(P, \tilde{\mathcal{D}})$ -solver $\mathcal{A} = (\mathcal{B}, \mathcal{C}, \text{state}_0 = \{\text{state}_{0,\lambda}\}_{\lambda})$ is defined as follows.

- $\text{state}_0 = \{\text{state}_{0,\lambda}\}_{\lambda}$ is a sequence of bipartite ℓ -qubit states (with the partitions being $\ell_{\mathcal{B}}$ -qubit register \mathbf{B} and $\ell_{\mathcal{C}}$ -qubit register \mathbf{C}).
- \mathcal{A} is a quantum algorithm that takes as input a security parameter 1^λ , a step 1^t , a pair of challenges $x_{\mathcal{B}}, x_{\mathcal{C}} \in \{0, 1\}^n$, and $\text{state} = (\text{state}_{\mathcal{B}}, \text{state}_{\mathcal{C}})$, which is an ℓ -qubit bipartite state. It runs $(\mathcal{B}(1^\lambda, x_{\mathcal{B}}, \cdot) \otimes \mathcal{C}(1^\lambda, x_{\mathcal{C}}, \cdot))(\text{state})$, with \mathcal{B} getting as input register \mathbf{B} and \mathcal{C} getting as input \mathbf{C} , to obtain $(y_{\mathcal{B}}, y_{\mathcal{C}}, \text{state}')$, where $y_{\mathcal{B}} \in \{0, 1\}^m$ is \mathcal{B} 's output, $y_{\mathcal{C}} \in \{0, 1\}^m$ is \mathcal{C} 's output and $\text{state}' = (\text{state}'_{\mathcal{B}}, \text{state}'_{\mathcal{C}})$ is a bipartite ℓ -qubit state on $\ell_{\mathcal{B}}$ -qubit register \mathbf{B} and $\ell_{\mathcal{C}}$ -qubit register \mathbf{C} with \mathbf{B} (resp., \mathbf{C}) being the output of \mathcal{B} (resp., \mathcal{C}). \mathcal{A} outputs $(y_{\mathcal{B}}, y_{\mathcal{C}}, \text{state}')$.

If the states $\text{state}_{i,\lambda}$ are classical strings and $(\mathcal{B}, \mathcal{C})$ are classical algorithms, then \mathcal{A} is called a classical non-local stateful solver, otherwise it is called a quantum non-local stateful solver. If $\tilde{\mathcal{D}} = \mathcal{U}_{\{0,1\}^d \times \{0,1\}^d}$, then \mathcal{A} is called an independent-challenge P -solver.

Remark 5. *Even though the definition above of a non-local $(P, \tilde{\mathcal{D}})$ -solver does not depend on $\tilde{\mathcal{D}}$, it will be used to define the value and advantage of the solver below.*

Examples of Non-Local Solvers: For instance, if P is inverting a one-way function f , then \mathcal{B} and \mathcal{C} respectively get $x_{\mathcal{B}} = f(z_{\mathcal{B}})$ and $x_{\mathcal{C}} = f(z_{\mathcal{C}})$ as challenges, where $(z_{\mathcal{B}}, z_{\mathcal{C}})$ is pair of uniform inputs arbitrarily correlated depending on $\tilde{\mathcal{D}}$. In order to pass verification, \mathcal{B} needs to output $y_{\mathcal{B}} = z_{\mathcal{B}}$ and \mathcal{C} needs to output $y_{\mathcal{C}} = z_{\mathcal{C}}$. Another example is when P is distinguishing the output of a PRG G from a random string. In this case, $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ are each either a uniform output of G or a uniformly random string. In order to pass verification, \mathcal{B} and \mathcal{C} need to correctly guess which one. The correct answer for \mathcal{B} and \mathcal{C} , may be arbitrarily correlated depending on $\tilde{\mathcal{D}}$. Furthermore,

²⁶A non-local algorithm can be implemented by two spatially separated and non-communicating parties, hence the name.

$x_{\mathcal{B}}, x_{\mathcal{C}}$ could be correlated still conditioned on the answers.

We formalize the interaction between a non-local solver and a non-local algorithm below. Formally, we adapt Definition 3.6 from [BBK22].

Definition 27 (Non-Local Solver Interaction). *Let $P = (\text{GenC}, \text{Ver}, c)$ be a non-interactive assumption. For any stateful $(P, \tilde{\mathcal{D}})$ -solver $\mathcal{A} = (\mathcal{B}_{\mathcal{A}}, \mathcal{C}_{\mathcal{A}}, \text{state}_0)$, and $\mathcal{S} = (\mathcal{B}_{\mathcal{S}}, \mathcal{C}_{\mathcal{S}})$, where $\mathcal{B}_{\mathcal{S}}$ and non-local algorithm $\mathcal{C}_{\mathcal{S}}$ are a pair of algorithms, with input $z = (z_{\mathcal{B}}, z_{\mathcal{C}}) \in \{0, 1\}^*$, we consider the process $\mathcal{S}_z^{\mathcal{A}}(1^\lambda)$ of the algorithm interacting with the solver. We define this process below.*

- \mathcal{S} is invoked on the input $(1^\lambda, z_{\mathcal{B}}, z_{\mathcal{C}}, \text{state}_0)$, where $\mathcal{B}_{\mathcal{S}}$ receives as input $z_{\mathcal{B}}$ and $\mathcal{C}_{\mathcal{S}}$ receives as input $z_{\mathcal{C}}$. Moreover, state_0 is defined on two registers \mathbf{B} and \mathbf{C} , with $\mathcal{B}_{\mathcal{S}}$ receiving the register \mathbf{B} and $\mathcal{C}_{\mathcal{S}}$ receiving the register \mathbf{C} . Initialize $\tau_{\mathcal{B}}$ to contain $z_{\mathcal{B}}$ and $\tau_{\mathcal{C}}$ to contain $z_{\mathcal{C}}$. At every step $i \geq 1$:
 1. $\mathcal{B}_{\mathcal{S}}(1^\lambda, \tau_{\mathcal{B}})$ makes a query $x_{\mathcal{B}}^{(i)}$ and $\mathcal{C}_{\mathcal{S}}(1^\lambda, \tau_{\mathcal{C}})$ makes a query $x_{\mathcal{C}}^{(i)}$,
 2. Run $(y_{\mathcal{B}}^{(i)}, y_{\mathcal{C}}^{(i)}, \text{state}_i) \leftarrow (\mathcal{B}_{\mathcal{A}}(1^\lambda, 1^i, x_{\mathcal{B}}^{(i)}, \cdot) \otimes \mathcal{C}_{\mathcal{A}}(1^\lambda, 1^i, x_{\mathcal{C}}^{(i)}, \cdot))(\text{state}_{i-1})$, where $y_{\mathcal{B}}^{(i)}$ is $\mathcal{B}_{\mathcal{A}}$'s output, $y_{\mathcal{C}}^{(i)}$ is $\mathcal{C}_{\mathcal{A}}$'s output and state_i is the joint output of $\mathcal{B}_{\mathcal{A}}$ and $\mathcal{C}_{\mathcal{A}}$.
 3. Add $(x_{\mathcal{B}}^{(i)}, y_{\mathcal{B}}^{(i)})$ to $\tau_{\mathcal{B}}$ and $(x_{\mathcal{C}}^{(i)}, y_{\mathcal{C}}^{(i)})$ to $\tau_{\mathcal{C}}$.
- At the end of the interaction, \mathcal{S} may produce the output $(w_{\mathcal{B}}, w_{\mathcal{C}})$, where $w_{\mathcal{B}}$ is output by $\mathcal{B}_{\mathcal{S}}$ and $w_{\mathcal{C}}$ is output by $\mathcal{C}_{\mathcal{S}}$.

We will sometimes refer to \mathcal{S} as a solver-aided non-local algorithm.

Non-Local Solver Value, Advantage, and Persistence. For stateful solvers, we explicitly define their value after interacting with the assumption many times.

Definition 28 (Non-local Stateful Value and Advantage). *Let $P = (\text{GenC}, \text{Ver}, c)$ be a non-interactive assumption and let $\mathcal{A} = (\mathcal{B}_{\mathcal{A}}, \mathcal{C}_{\mathcal{A}}, \text{state}_0)$ be a stateful $(P, \tilde{\mathcal{D}})$ -non-local solver. Let \mathcal{S} be a solver-aided non-local algorithm with input $z = (z_{\mathcal{B}}, z_{\mathcal{C}})$. Let state_i be defined as in Definition 27 for the interaction $\mathcal{S}_z^{\mathcal{A}}(1^\lambda)$. Then, we define the (many-shot) value of $\mathcal{S}_z^{\mathcal{A}}$ as*

$$\text{val}_P^{\tilde{\mathcal{D}}} [i, \mathcal{S}_z^{\mathcal{A}}] := \Pr \left[\begin{array}{c} \text{Ver}(1^\lambda, y_{\mathcal{B}}; r_{\mathcal{B}}) = 1 \\ \wedge \\ \text{Ver}(1^\lambda, y_{\mathcal{C}}; r_{\mathcal{C}}) = 1 \end{array} \middle| \begin{array}{c} (r_{\mathcal{B}}, r_{\mathcal{C}}) \leftarrow \tilde{\mathcal{D}} \\ x_{\mathcal{B}} = \text{GenC}(1^\lambda; r_{\mathcal{B}}) \\ x_{\mathcal{C}} = \text{GenC}(1^\lambda; r_{\mathcal{C}}) \\ (y_{\mathcal{B}}, y_{\mathcal{C}}, \text{state}_{i+1}) \leftarrow (\mathcal{B}_{\mathcal{A}}(1^\lambda, 1^{i+1}, x_{\mathcal{B}}, \cdot) \otimes \mathcal{C}_{\mathcal{A}}(1^\lambda, 1^{i+1}, x_{\mathcal{C}}, \cdot))(\text{state}_i) \end{array} \right],$$

and for any threshold $c^* : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\}$, we define the (many-shot) advantage of $\mathcal{S}_z^{\mathcal{A}}$ as

$$\text{adv}_P^{(\tilde{\mathcal{D}}, c^*)} [i, \mathcal{A}] := \left| \text{val}_P^{\tilde{\mathcal{D}}} [i, \mathcal{S}_z^{\mathcal{A}}] - c^*(\lambda) \right|$$

We define the one-shot value of the solver interaction $\mathcal{S}_z^{\mathcal{A}}$ as $\text{val}_P^{\tilde{\mathcal{D}}} [\mathcal{A}] := \text{val}_P^{\tilde{\mathcal{D}}} [0, \mathcal{S}_z^{\mathcal{A}}]$ and the one-shot advantage of \mathcal{A} with respect to threshold c^* as $\text{adv}_P^{(\tilde{\mathcal{D}}, c^*)} [\mathcal{A}] := \text{adv}_P^{(\tilde{\mathcal{D}}, c^*)} [0, \mathcal{S}_z^{\mathcal{A}}]$. Note that neither value depends on (\mathcal{S}, z) .

Remark 6 (Non-Local vs. Local Threshold). *Above, the non-local threshold c^* , which is a parameter of the advantage $\text{adv}_P^{(\tilde{\mathcal{D}}, c^*)}$ of the adversary \mathcal{A} , can depend on the local threshold c as well as the d -extension $\tilde{\mathcal{D}}$.*

Definition 29 (Non-Local Persistence). *Let P be a non-interactive assumption. A distribution \mathbb{B} on $(P, \tilde{\mathcal{D}})$ -non-local solvers $\{\mathcal{A}^\alpha = (\mathcal{B}_A^\alpha, \mathcal{C}_A^\alpha, \text{state}_0^\alpha)\}_\alpha$ is called (p, η) -persistent if for any solver-aided non-local algorithm $\mathcal{S} = (\mathcal{B}_S, \mathcal{C}_S)$ with input $z = (z_B, z_C)$, we have*

$$\Pr \left[\sup_i \left| \text{val}_P^{\tilde{\mathcal{D}}} [i, \mathcal{S}_z^{\mathcal{A}^\alpha}] - p \right| \leq \eta \right] \geq 1 - \eta,$$

where the probability is taken over the randomness of α and the solver interaction $\mathcal{S}_z^{\mathcal{A}^\alpha}$.

5.1.3 Reductions

A reduction is a transformation that converts a solver for one problem to a solver for another (possibly different) problem. More precisely, suppose P and Q be two assumptions. A reduction leverages the solver for P to design a solver for Q . We consider the setting when the solver for Q uses the solver for P as a black-box. As in [BBK22], we consider reductions where the solver for Q runs the solver for P multiple times. We define analogous notions of one-shot advantage and persistent advantage below.

We first define the traditional notion of reduction before defining non-local reductions.

Definition 30 (Reduction; Definition 3.12 in [BBK22]). *A classical (resp., quantum) reduction from solving a non-interactive assumption Q to solving a non-interactive assumption P is an efficient classical (resp., quantum) uniform algorithm \mathcal{R} with the following guarantee.*

For any classical (resp., quantum) P -solver $\mathcal{A}_P = (A_P, \text{state}_0)$ with one-shot advantage ε and runtime T , let $\text{state}'_0 = (\text{state}_0, A_P, 1^{1/\varepsilon}, 1^T)$. Then, $\mathcal{A}_Q = (\mathcal{R}, \text{state}'_0)$ is a classical (resp., quantum) Q -solver with one-shot advantage $\varepsilon' = \text{poly}(\varepsilon, T^{-1}, \lambda^{-1})$ and runtime $\text{poly}(T, \varepsilon^{-1}, \lambda)$. An inefficient reduction is defined the same way without the runtime requirement and the dependency on T .

Non-Local Reductions. We now generalize the above definition to the non-local setting.

Definition 31 (Non-Local Reduction). *A non-local classical (resp., quantum) $(\tilde{\mathcal{D}}_P, c_P^*, \tilde{\mathcal{D}}_Q, c_Q^*)$ -reduction from solving a non-interactive assumption Q to solving a non-interactive assumption P is an efficient classical (resp., quantum) uniform non-local algorithm $\mathcal{R} = (\mathcal{R}_1, \mathcal{R}_2)$ with the following guarantee.*

Given any non-local classical (resp., quantum) $(P, \tilde{\mathcal{D}})$ -solver $\mathcal{A}_P = (\mathcal{B}_P, \mathcal{C}_P, \text{state}_0)$ with one-shot advantage $\varepsilon = \text{adv}_P^{(\tilde{\mathcal{D}}, c_P^)} [\mathcal{A}_P]$ and runtime T , let $\text{state}'_0 = (\text{state}_0, \mathcal{B}_P, \mathcal{C}_P, 1^{1/\varepsilon}, 1^T)$. Then, $\mathcal{A}_Q = (\mathcal{R}_1, \mathcal{R}_2, \text{state}'_0)$ is a $(Q, \tilde{\mathcal{D}}_Q)$ -solver with one-shot advantage $\varepsilon' = \text{adv}_Q^{(\tilde{\mathcal{D}}, c_Q^*)} [\mathcal{A}_Q] = \text{poly}(\varepsilon, T^{-1}, \lambda^{-1})$ and runtime $\text{poly}(T, \varepsilon^{-1}, \lambda)$. Here, it is understood that description of \mathcal{B}_P (resp., \mathcal{C}_P) is included as part of \mathcal{R}_1 's (resp., \mathcal{R}_2 's) register. An inefficient non-local reduction is defined the same way without the runtime requirement and the dependency of T .*

Definition 32 (Non-Local Black-Box Reduction). *A non-local black-box $(\tilde{\mathcal{D}}_P, c_P^*, \tilde{\mathcal{D}}_Q, c_Q^*)$ -reduction $\mathcal{R} = (\mathcal{R}_1, \mathcal{R}_2)$ from solving $Q = (\text{GenC}_Q, \text{Ver}_Q, c_Q)$ to solving $P = (\text{GenC}_P, \text{Ver}_P, c_P)$ is a non-local $(\tilde{\mathcal{D}}_P, c_P^*, \tilde{\mathcal{D}}_Q, c_Q^*)$ -reduction such that \mathcal{R}_1 (resp., \mathcal{R}_2) interacts with \mathcal{B}_P (resp., \mathcal{C}_P) as defined in Definition 27.*

We further say that \mathcal{R} is non-adaptive if \mathcal{R}_1 (resp., \mathcal{R}_2) produces its queries to \mathcal{B}_P (resp., \mathcal{C}_P) independent of the answers to its previous queries (or equivalently all at once).

5.2 Main Theorem

In our main theorem, we show how to relate the unclonable security of two cloning games $\mathcal{G}, \mathcal{G}'$ which satisfy a similarity condition in the setup phase as well as some additional properties. Before we state the formal theorem, we first describe how the notions of non-interactive assumption and unclonable security are connected by introducing relevant notation.

Assumption Induced by a Cloning Experiment. Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a stateful²⁷ cloning game with message space \mathcal{M} . For every $m \in \mathcal{M}$, message distribution $\mathcal{D}_{\mathcal{M}}$, and secret key sk in the support of $\text{Setup}(1^\lambda)$, we consider the induced non-interactive assumption $\mathfrak{A}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}} = (d, n, \ell, \text{GenC}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}, \text{Ver}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}, c)$. Here d is the length of the random coins used by GenC , n is the length of the challenges output by GenC , and ℓ is the length of the answers received by GenC . In addition, $\text{GenC}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}(1^\lambda; r) := \text{GenC}(\text{sk}, m; r)$ and $\text{Ver}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}(1^\lambda, r, \text{ans}) := \text{Ver}(\text{sk}, m, \text{ch}, \text{ans}, r)$.²⁸ Finally, we will set $c := p^{\text{triv}}(\mathcal{G}, \mathcal{D}_{\mathcal{M}})$ for the message distribution \mathcal{M} considered for unclonable security. As before, we write $P_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}} = (\text{GenC}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}, \text{Ver}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}, c)$ to denote the assumption when the parameters d, n, ℓ are implicit.

Theorem 13 (Main Theorem). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ and $\mathcal{G}' = (\text{Setup}', \text{GenT}', \text{GenC}', \text{Ver}')$ be cloning games with the same message space \mathcal{M} . Let $\mathcal{D}_{\mathcal{M}}$ be a message distribution, and $\widetilde{\text{GenC}}$ be an extension of GenC . Suppose that the following conditions are satisfied:*

- (1) $\text{Setup} = \text{Setup}'$ and $\text{GenT} = \text{GenT}'$.
- (2) \mathcal{G}' is $\mathcal{D}_{\mathcal{M}}$ -evasive.
- (3) Either \mathcal{G} is $\mathcal{D}_{\mathcal{M}}$ -evasive or $\widetilde{\text{GenC}} = \widetilde{\text{GenC}}_{\text{ind}}$.

For sk in the support of Setup and message $m \in \mathcal{M}$, consider the induced non-interactive assumptions

$$P = P_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}} = (\text{GenC}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}, \text{Ver}_{\text{sk},m}^{\mathcal{G},\mathcal{D}_{\mathcal{M}}}, p^{\text{triv}}(\mathcal{G}, \mathcal{D}_{\mathcal{M}}, \widetilde{\text{GenC}}))$$

and

$$Q = Q_{\text{sk},m}^{\mathcal{G}',\mathcal{D}_{\mathcal{M}}} = (\text{GenC}_{\text{sk},m}^{\mathcal{G}',\mathcal{D}_{\mathcal{M}}}, \text{Ver}_{\text{sk},m}^{\mathcal{G}',\mathcal{D}_{\mathcal{M}}}, p^{\text{triv}}(\mathcal{G}', \mathcal{D}_{\mathcal{M}})).$$

Suppose further that the following conditions are satisfied:

- (a) For all (sk, m) , P has a verifiably polynomial image.
- (b) For all (sk, m) , there exists a classical non-adaptive black-box reduction \mathcal{R} from solving non-interactive assumption Q to solving non-interactive assumption P . Furthermore, the description of \mathcal{R} does not depend on (sk, m) .
- (c) \mathcal{G}' has $(\mathcal{D}_{\mathcal{M}}, \text{negl})$ independent-challenge unclonable security.

²⁷Recall that stateful cloning games (Definition 12) generalize cloning games, so that this definition applies the same to cloning games that are not stateful.

²⁸Note that this is well-defined since ch can be computed deterministically given sk, m, r .

Then, \mathcal{G} has $(\mathcal{D}_{\mathcal{M}}, \text{negl}, \widetilde{\text{GenC}})$ unclonable security.

The proof of the main theorem consists of 3 technical steps: (1) going from the classical reduction given in item (b) to a classical non-local reduction, (2) going from non-local $(Q, \widetilde{\mathcal{D}}_Q)$ -solver to independent-challenge Q -solver, and (3) lifting classical reduction to quantum. We will formally discuss each step in Section 5.3. We include the final proof below to aid the reader in reading Section 5.3.

Proof of Theorem 13. Define $c_P = p^{\text{triv}}(\mathcal{G}, \mathcal{D}_{\mathcal{M}}, \widetilde{\text{GenC}})$ and $c_Q = p^{\text{triv}}(\mathcal{G}', \mathcal{D}_{\mathcal{M}})$. By condition (2), Q is evasive, i.e. c_Q is negligible. By conditions (2),(b) and Lemma 16, there exists a classical non-adaptive black-box non-local $(\widetilde{\mathcal{D}}_O, c_P, \widetilde{\mathcal{D}}_Q, c_Q)$ -reduction from solving Q to solving P , where $\widetilde{\mathcal{D}}_P, \widetilde{\mathcal{D}}_Q$ denote uniform extensions.

By Theorem 22, there exists a quantum non-local $(\widetilde{\mathcal{D}}_O, c_P, \widetilde{\mathcal{D}}_Q, c_Q)$ -reduction \mathcal{R}' from solving Q to solving P . Combining this with Corollary 3 if \mathcal{G} is $\mathcal{D}_{\mathcal{M}}$ -evasive, and trivially if $\widetilde{\text{GenC}} = \widetilde{\text{GenC}}_{\text{ind}}$, \mathcal{R}' is also a quantum non-local $(\widetilde{\text{GenC}}, c_P, \widetilde{\mathcal{D}}_Q, c_Q)$ -reduction from solving Q to solving P .

Now, let $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ be a QPT adversary which breaks $(\mathcal{D}_{\mathcal{M}}, \varepsilon)$ unclonable security of \mathcal{G} for a non-negligible function ε . Let $\varepsilon_{\text{sk},m}$ be the one-shot value of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ in assumption P defined above with respect to c_P , so that $|\mathbb{E}_{\text{sk},m} [\varepsilon_{\text{sk},m} - c_P]| > \varepsilon$. Note that since the description of \mathcal{R} does not depend on (sk, m) , neither does the description of \mathcal{R}' .

Let $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ be the induced QPT adversary obtained by giving the solver $(\mathcal{B}', \mathcal{C}', \mathcal{A}(\rho_{\text{sk},m}))$ as input to \mathcal{R} , where $\rho_{\text{sk},m} \leftarrow \text{GenT}(\text{sk}, m)$ is the token received by \mathcal{A} in \mathcal{G} . That is, \mathcal{A}' is defined in terms of \mathcal{A} and how the state $\mathcal{A}(\rho_{\text{sk},m})$ is modified by the reduction \mathcal{R}' , whereas $(\mathcal{B}', \mathcal{C}')$ is the non-local algorithm output by \mathcal{R}' . Note that this is well-defined since the description of \mathcal{R}' does not depend on (sk, m) . Define $\varepsilon'_{\text{sk},m}$ as the one-shot value of $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ in \mathcal{G}' .

By the guarantee of \mathcal{R}' , we have $|\varepsilon'_{\text{sk},m} - c_Q| \geq q(|\varepsilon_{\text{sk},m} - c_P|)$ for all (sk, m) and some polynomial q . Without loss of generality, we can take $q(x) = x^\beta$ for some constant $\beta > 1$, so that q is a monotone, convex function. Taking the expectation and using Jensen's Inequality, we obtain

$$\begin{aligned} q(\varepsilon) &< q\left(\left|\mathbb{E}_{\text{sk},m} [\varepsilon_{\text{sk},m} - c_P]\right|\right) \leq q\left(\mathbb{E}_{\text{sk},m} [|\varepsilon_{\text{sk},m} - c_P|]\right) \leq \mathbb{E}_{\text{sk},m} [q(|\varepsilon_{\text{sk},m} - c_P|)] \leq \mathbb{E}_{\text{sk},m} [|\varepsilon'_{\text{sk},m} - c_Q|] \\ &\leq \mathbb{E}_{\text{sk},m} [\varepsilon'_{\text{sk},m} + c_Q] \leq 2c_Q + \left|\mathbb{E}_{\text{sk},m} [\varepsilon'_{\text{sk},m} - c_Q]\right|, \end{aligned}$$

hence $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ breaks the $(\mathcal{D}_{\mathcal{M}}, q(\varepsilon) - 2c_Q)$ unclonable security of \mathcal{G}' , which suffices for the proof since c_Q is negligible and ε is non-negligible. \square

Alternate Versions of the Main Theorem. We give two more versions of the main theorem, which could be useful for applications. The first one (Theorem 14) is for information theoretic applications, including our single-decryptor encryption construction in Section 5.4.1. The second one (Theorem 15) concerns the case when $\mathcal{G}, \mathcal{G}'$ do not have identical setup phases, yet they are equivalent modulo a local quantum transformation applied by \mathcal{A} , hence relaxing condition (1) of Theorem 13.

Theorem 14 (Main Theorem - Inefficient Version). *Let $P, Q, \mathcal{G}, \mathcal{G}', \mathcal{D}_{\mathcal{M}}, \widetilde{\text{GenC}}$ be defined as in Theorem 13. Suppose that the conditions in items (1) to (3) and item (b) are satisfied. If \mathcal{G}' has*

information theoretic $(\mathcal{D}_{\mathcal{M}}, \text{negl})$ independent-challenge unclonable security, then \mathcal{G} has information theoretic $(\mathcal{D}_{\mathcal{M}}, \text{negl}, \text{GenC})$ unclonable security.

Proof. The proof follows the same as that of Theorem 13. The only difference is that since P has no verifiably polynomial image, the non-local reduction is inefficient as per Theorem 22. \square

Theorem 15 (Main Theorem - General Version). *Theorem 13 holds true if condition (1) is replaced with the following: There exists a quantum channel Γ such that,*

$$\text{TD}(\rho_{\text{sk},m}, \Gamma(\rho'_{\text{sk},m})) \leq \text{negl}(\lambda)$$

for all key-message pairs (sk, m) , where $\rho_{\text{sk},m} \leftarrow \text{GenT}(\text{sk}, m)$ and $\rho'_{\text{sk},m} \leftarrow \text{GenT}'(\text{sk}, m)$.

Proof (sketch). Follows by modifying the proof of Theorem 13 so that the non-local reduction first applies Γ . \square

We note that the same generalization can be applied to Theorem 14 in the information theoretic setting.

5.3 Proof of the Main Theorem.

In this section, we give the technical details of the 3 steps of the proof of the main theorem.

Step 1: Lifting Classical Local Reduction to Non-Local Reduction. In Section 5.3.1, we show that any classical reduction from solving Q to solving P implies a classical non-local reduction, where the assumption Q is assigned the independent-challenge distribution.

Step 2: Converting Non-Local Solver to Independent-Challenge Solver. In Section 5.3.2, we show that any non-local solver with negligible non-local threshold is an independent-challenge solver. This step is needed because the third step below requires independent-challenge mode to lift a classical non-local reduction to quantum.

Step 3: Lifting Classical Non-Local Reductions to Quantum Reductions. In Sections 5.3.3 to 5.3.5, we adapt the work of [BBK22], which shows how to lift local non-adaptive black-box reductions to quantum, to the non-local setting. The main technical contribution occurs in Section 5.3.3, where we do non black-box analysis of the state-repair procedure of [CMSZ21] to show that it can be used in the non-local setting.

5.3.1 Upgrading Classical Local Reductions to Classical Non-Local Reductions

Most classical reductions known are for local assumptions, and we will show below how to obtain a non-local reduction from a local reduction. The non-local reduction simply runs the local reduction in each register.

Lemma 16. *Suppose there exists a classical reduction \mathcal{R} from solving non-interactive evasive assumption $Q = (\text{GenC}_Q, \text{Ver}_Q, c_Q)$ to solving non-interactive assumption $P = (\text{GenC}_P, \text{Ver}_P, c_P)$. Let $\tilde{\mathcal{D}}_Q = \mathcal{U}_{\{0,1\}^{d_Q} \times \{0,1\}^{d_Q}}$ be the uniform extension. Then, there exists a classical non-local $(\tilde{\mathcal{D}}_P, c_P, \tilde{\mathcal{D}}_Q, c_Q)$ -reduction $\mathcal{R}' = (\mathcal{R}'_1, \mathcal{R}'_2)$ from solving Q to solving P for any d_P -extension $\tilde{\mathcal{D}}_P$. Furthermore, \mathcal{R}' is non-adaptive (resp., black-box) if \mathcal{R} is non-adaptive (resp., black-box).*

Proof. The non-local reduction locally runs the classical reduction, i.e. $\mathcal{R}'_1 = \mathcal{R}'_2 = \mathcal{R}$. Let $\mathcal{A} = (\mathcal{B}, \mathcal{C})$ be a $(P, \tilde{\mathcal{D}}_P)$ -solver with one-shot advantage $\varepsilon = \text{adv}_P^{(\tilde{\mathcal{D}}_P, c_P)}[\mathcal{A}]$. Let $\mathcal{A}' = (\mathcal{B}', \mathcal{C}')$ be the non-local adversary output by \mathcal{R}' . Assume that $\mathcal{A}, \mathcal{A}'$ are both stateless. Since $\varepsilon + c_P$ is the probability of $(\mathcal{B}, \mathcal{C})$ simultaneously passing verification, it follows that \mathcal{B} and \mathcal{C} are both P -solvers with advantage ε . Thus, by assumption \mathcal{B}' and \mathcal{C}' are both Q -solvers with advantage $\varepsilon' = \text{poly}(\varepsilon, T^{-1}, \lambda)$, where T is the runtime of \mathcal{A} . Therefore, since \mathcal{A}' is an independent-challenge non-local Q -solver, it follows that the advantage of \mathcal{A}' is at least $(\varepsilon' + c_Q)^2 - c_Q \geq (\varepsilon')^2 - \text{negl}(\lambda)$ as desired.

The argument can be generalized to stateful non-local solvers by convexity. \square

5.3.2 From Non-Local Solvers to Independent-Challenge Solvers

For the second step of the proof of the main theorem, we will show that any non-local solver is an independent-challenge solver, up to polynomial loss.

Theorem 17. *Let P be a non-interactive assumption and let $c : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\}$ be a negligible non-local threshold. Then, any non-local quantum $(P, \tilde{\mathcal{D}})$ -solver $\mathcal{A} = (\mathcal{B}, \mathcal{C}, \text{state}_0)$ with one-shot advantage $\varepsilon = \text{adv}_P^{(\tilde{\mathcal{D}}, c)}[\mathcal{A}]$ is a quantum independent-challenge P -solver with advantage $\varepsilon' = \text{adv}_P^{(\tilde{\mathcal{D}}_{\text{ind}}, c)}[\mathcal{A}] = \text{poly}(\varepsilon) - \text{negl}(\lambda)$, where $\tilde{\mathcal{D}}_{\text{ind}}$ is the uniform extension.*

Proof. Since c is negligible, it suffices to show $\text{val}_P^{(\tilde{\mathcal{D}}_{\text{ind}}, c)}[\mathcal{A}] = \text{poly}(\text{val}_P^{(\tilde{\mathcal{D}}, c)}[\mathcal{A}])$. Since the success of a non-local solver can be implemented as a product measurement, this follows from Lemma 18 below, with the polynomial above given by $q(x) = x^3/216$. \square

Lemma 18. *Let \mathcal{R} be a finite set of random coins, $\tilde{\mathcal{D}}$ be a distribution over $\mathcal{R} \times \mathcal{R}$ with marginals $(\mathcal{D}, \mathcal{D})$, and ρ be a mixed bipartite state. Define the following quantities:*

$$p_{\text{cor}} = \mathbb{E}_{(r, r') \leftarrow \tilde{\mathcal{D}}} \text{Tr}[(B_r \otimes C_{r'}) \rho], \quad p_{\text{ind}} = \mathbb{E}_{r, r' \leftarrow \mathcal{D}} \text{Tr}[(B_r \otimes C_{r'}) \rho],$$

where for each $r \in \mathcal{R}$, we have $0 \leq B_r \leq I$ and $0 \leq C_r \leq I$. Then,

$$p_{\text{id}} \leq 6\sqrt[3]{p_{\text{ind}}}$$

Proof. We can interpret B_r and C_r as POVM elements. Thus, by Naimark Dilation theorem, without loss of generality we can assume that B_r and C_r are projections. We can also assume that $\rho = |\psi\rangle\langle\psi|$ is a pure state, and the mixed state case follows by convexity, since $f(x) = 6\sqrt[3]{x}$ is a concave function.

Define PSD operators

$$P_B := \mathbb{E}_{r \leftarrow \mathcal{D}} B_r, \quad P_C := \mathbb{E}_{r \leftarrow \mathcal{D}} C_r,$$

and let

$$|\psi\rangle = \sum_{i,j} \alpha_{i,j} |\phi_i\rangle |\sigma_j\rangle$$

be the spectral decomposition of $|\psi\rangle$ with respect to $P_B \otimes P_C$, where $|\phi_i\rangle$ is an eigenvector of P_B with eigenvalue λ_i and $|\sigma_j\rangle$ is an eigenvector of P_C with eigenvalue γ_j . Set $\eta = \sqrt[3]{p_{ind}}$ and define subnormalized states

$$|\psi_B\rangle = \sum_{\substack{i: \lambda_i \leq \eta \\ j: \gamma_j > \eta}} \alpha_{i,j} |\phi_i\rangle |\sigma_j\rangle, \quad |\psi_C\rangle = \sum_{\substack{i \\ j: \gamma_j \leq \eta}} \alpha_{i,j} |\phi_i\rangle |\sigma_j\rangle, \quad |\psi_{BC}\rangle = \sum_{\substack{i: \lambda_i > \eta \\ j: \gamma_j > \eta}} \alpha_{i,j} |\phi_i\rangle |\sigma_j\rangle,$$

so that $|\psi\rangle = |\psi_B\rangle + |\psi_C\rangle + |\psi_{BC}\rangle$.

We first show the following claim:

Claim 1. $\|\psi_{BC}\|^2 \leq \frac{p_{ind}}{\eta^2}$, where $\|\cdot\|$ is the Euclidean Norm.

Proof.

$$\begin{aligned} p_{ind} &= \mathbb{E}_{r,r' \leftarrow \mathcal{D}} \langle \psi | (B_r \otimes C_{r'}) | \psi \rangle \\ &= \langle \psi | P_B \otimes P_C | \psi \rangle \\ &= \sum_{i,j} |\alpha_{i,j}|^2 \lambda_i \gamma_j \\ &\geq \sum_{\substack{i: \lambda_i > \eta \\ j: \gamma_j > \eta}} |\alpha_{i,j}|^2 \lambda_i \gamma_j \\ &\geq \eta^2 \sum_{\substack{i: \lambda_i > \eta \\ j: \gamma_j > \eta}} |\alpha_{i,j}|^2 \\ &= \eta^2 \|\psi_{BC}\|^2 \end{aligned}$$

□

We have:

$$\begin{aligned} p_{cor} &= \mathbb{E}_{(r,r') \leftarrow \tilde{\mathcal{D}}} \|(B_r \otimes C_{r'}) (|\psi_B\rangle + |\psi_C\rangle + |\psi_{BC}\rangle)\|^2 \\ \text{(Cauchy-Schwartz)} &\leq \mathbb{E}_{(r,r') \leftarrow \tilde{\mathcal{D}}} 3 \left(\|(B_r \otimes C_{r'}) |\psi_B\rangle\|^2 + \|(B_r \otimes C_{r'}) |\psi_C\rangle\|^2 + \|(B_r \otimes C_{r'}) |\psi_{BC}\rangle\|^2 \right) \\ &\leq \mathbb{E}_{(r,r') \leftarrow \tilde{\mathcal{D}}} 3 \left(\|(B_r \otimes I) |\psi_B\rangle\|^2 + \|(I \otimes C_{r'}) |\psi_C\rangle\|^2 + \|\psi_{BC}\|^2 \right) \\ &= \mathbb{E}_{r,r' \leftarrow \mathcal{D}} 3 \left(\|(B_r \otimes I) |\psi_B\rangle\|^2 + \|(I \otimes C_{r'}) |\psi_C\rangle\|^2 + \|\psi_{BC}\|^2 \right) \\ &= 3 \left(\langle \psi_B | (P_B \otimes I) \psi_B \rangle + \langle \psi_C | (I \otimes P_C) \psi_C \rangle + \|\psi_{BC}\|^2 \right) \\ \text{(Claim 1)} &\leq 3 \left(\eta \left(\|\psi_B\|^2 + \|\psi_C\|^2 \right) + \frac{p_{ind}}{\eta^2} \right) \\ &\leq 3 \left(\eta + \frac{p_{ind}}{\eta^2} \right) \\ &= 6 \sqrt[3]{p_{ind}} \end{aligned}$$

□

Classical Case. We list the special case when \mathcal{A} in Theorem 17 is a classical algorithm explicitly below because (1) we will need it in Section 5.4.2 for our second application of the main theorem, and (2) we can show a quadratic (instead of cubic) loss for this special case.

Corollary 3. *Let P be a non-interactive assumption and let $c : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\}$ be a negligible non-local threshold. Then, any non-local classical $(P, \tilde{\mathcal{D}})$ -solver $\mathcal{A} = (\mathcal{B}, \mathcal{C}, \text{state}_0)$ with one-shot advantage $\varepsilon = \text{adv}_P^{(\tilde{\mathcal{D}}, c)}[\mathcal{A}]$ is a classical independent-challenge P -solver with advantage $\varepsilon' = \text{adv}_P^{(\tilde{\mathcal{D}}, c)}[\mathcal{A}] = \varepsilon^2 - \text{negl}(\lambda)$.*

Proof. Although it follows as a special (classical) case of Theorem 17, we will give a direct proof, as the classical concrete bound trumps the quantum bound. We will show that if the one-shot value of \mathcal{A} with respect to $\tilde{\mathcal{D}}$ is δ , then its one-shot value in the independent-challenge setting is at least δ^2 , which suffices for the proof. The classical state state_0 can be modeled as a shared random string. We assume that \mathcal{A} is stateless and the general case follows by convexity.

Let \mathcal{R} be the space of random coins used by the challenge generation algorithm **GenC** of P . Let p_r and q_r be the probability that \mathcal{B} and \mathcal{C} pass verification conditioned on the challenge being generated using randomness r , respectively. Since \mathcal{A} is stateless, we can write the one-shot value of \mathcal{A} in the independent-challenge setting as²⁹

$$\delta' = \frac{1}{|\mathcal{R}|^2} \sum_{r, r'} p_r q_{r'}$$

On the other hand, the one-shot value of \mathcal{A} with respect to the extension $\tilde{\mathcal{D}}$ can be written as

$$\delta = \sum_{r, r'} \alpha_{r, r'} p_r q_{r'},$$

where $\{\alpha_{r, r'}\}_{r, r'}$ are non-negative coefficients satisfying

$$\sum_{r'} \alpha_{r', r} = \sum_{r'} \alpha_{r, r'} = \frac{1}{|\mathcal{R}|}$$

for all $r \in \mathcal{R}$. Thus, we have

$$\begin{aligned} \delta^2 &= \sum_{r, r', s, s'} \alpha_{r, r'} \alpha_{s, s'} p_r q_{r'} p_s q_{s'} \leq \sum_{r, r', s, s'} \alpha_{r, r'} \alpha_{s, s'} p_r q_{s'} \\ &= \sum_{r, s'} p_r q_{s'} \sum_{r', s} \alpha_{r, r'} \alpha_{s, s'} \\ &= \frac{1}{|\mathcal{R}|^2} \sum_{r, s'} p_r q_{s'} \\ &= \delta' \end{aligned}$$

as desired. □

²⁹This equation can be thought of as a no-signalling condition.

5.3.3 Main Lemma

We will extend Lemma 10 to the non-local setting.

Lemma 19 (Main Lemma). *Let V be a classical algorithm and let $\mathcal{A} = \mathcal{B} \otimes \mathcal{C}$ be a non-local algorithm, where \mathcal{B} (resp., \mathcal{C}) acts on register \mathbf{B} (resp., \mathbf{C}). Define the following algorithms (NLValEst, NLRepair) using the algorithms ValEst and Repair described in Lemma 10:*

- $(\rho_{\mathbf{BC}}^*, p_{\mathbf{B}}^*, p_{\mathbf{C}}^*) \leftarrow \text{NLValEst}_{V, \mathcal{A}}(\rho, 1^{1/\varepsilon})$: compute $(\text{ValEst}_{V, \mathcal{B}}(\cdot, 1^{1/(\varepsilon/2)}) \otimes \text{ValEst}_{V, \mathcal{C}}(\cdot, 1^{1/(\varepsilon/2)}))(\rho_{\mathbf{BC}})$ to obtain $(p_{\mathbf{B}}^*)$ (\mathcal{B} 's output), $p_{\mathbf{C}}^*$ (\mathcal{C} 's output) and $\rho_{\mathbf{BC}}^*$ is the joint output of \mathcal{B} and \mathcal{C} , where \mathcal{B} (resp., \mathcal{C}) output the register \mathbf{B} (resp., \mathbf{C}).
- $\sigma_{\mathbf{BC}}^* \leftarrow \text{NLRepair}_{V, \mathcal{A}, \Pi_{\mathbf{B}}, \Pi_{\mathbf{C}}}(\sigma_{\mathbf{BC}}, y_{\mathbf{B}}, p_{\mathbf{B}}, y_{\mathbf{C}}, p_{\mathbf{C}}, 1^{1/\varepsilon}, 1^k)$: for k -outcome projections $\Pi_{\mathbf{B}}, \Pi_{\mathbf{C}}$ on register \mathbf{B}, \mathbf{C} , respectively, compute the following:

$$\left(\text{Repair}_{V, \mathcal{B}, \Pi_{\mathbf{B}}}(\cdot, y_{\mathbf{B}}, p_{\mathbf{B}}, 1^{1/\varepsilon}, 1^k) \otimes \text{Repair}_{V, \mathcal{C}, \Pi_{\mathbf{C}}}(\cdot, y_{\mathbf{C}}, p_{\mathbf{C}}, 1^{1/\varepsilon}, 1^k) \right) (\sigma_{\mathbf{BC}})$$

Denote the result by $\sigma_{\mathbf{BC}}^*$.

Note that (NLValEst, NLRepair) are efficient³⁰ algorithms given that (ValEst, Repair) are efficient algorithms by Lemma 10. In addition, we have the following guarantees:

1. **Value Estimation:** For any $\varepsilon > 0$,

$$\mathbb{E}_{(\rho_{\mathbf{BC}}^*, p_{\mathbf{B}}^*, p_{\mathbf{C}}^*) \leftarrow \text{NLValEst}_{V, \mathcal{A}}(\rho_{\mathbf{BC}}, 1^{1/\varepsilon})} [p_{\mathbf{B}}^* p_{\mathbf{C}}^*] = \Pr \left[\begin{array}{l} V(y_{\mathbf{B}}; r_{\mathbf{B}}) = 1 \\ \wedge \\ V(y_{\mathbf{C}}; r_{\mathbf{C}}) = 1 \end{array} : \begin{array}{l} r_{\mathbf{B}}, r_{\mathbf{C}} \leftarrow \{0, 1\}^d \\ (y_{\mathbf{B}}, y_{\mathbf{C}}) \leftarrow \mathcal{B}(\cdot, r_{\mathbf{B}}) \otimes \mathcal{C}(\cdot, r_{\mathbf{C}}) (\rho_{\mathbf{BC}}) \end{array} \right]. \quad (5)$$

2. **Almost-Projective Estimation:** For any $\varepsilon \geq \varepsilon' > 0$,

$$\Pr \left[|p_{\mathbf{B}}^* - p_{\mathbf{B}}^{**}| \geq \varepsilon \vee |p_{\mathbf{C}}^* - p_{\mathbf{C}}^{**}| \geq \varepsilon : \begin{array}{l} (\rho_{\mathbf{BC}}^*, p_{\mathbf{B}}^*, p_{\mathbf{C}}^*) \leftarrow \text{NLValEst}_{V, \mathcal{A}}(\rho_{\mathbf{BC}}, 1^{1/\varepsilon}) \\ (\rho_{\mathbf{BC}}^{**}, p_{\mathbf{B}}^{**}, p_{\mathbf{C}}^{**}) \leftarrow \text{NLValEst}_{V, \mathcal{A}}(\rho_{\mathbf{BC}}^*, 1^{1/\varepsilon'}) \end{array} \right] \leq \varepsilon.$$

3. **Repairing:** For any state $\rho_{\mathbf{BC}}$ and any $\varepsilon > 0$,

$$\Pr \left[|p_{\mathbf{B}}^* - p_{\mathbf{B}}^{**}| \geq \varepsilon \vee |p_{\mathbf{C}}^* - p_{\mathbf{C}}^{**}| \geq \varepsilon : \begin{array}{l} (\rho_{\mathbf{BC}}^*, p_{\mathbf{B}}^*, p_{\mathbf{C}}^*) \leftarrow \text{NLValEst}_{V, \mathcal{A}}(\rho_{\mathbf{BC}}, 1^{1/\varepsilon}) \\ (\sigma_{\mathbf{BC}}, y_{\mathbf{B}}, y_{\mathbf{C}}) \leftarrow (\Pi_{\mathbf{B}} \otimes \Pi_{\mathbf{C}})(\rho_{\mathbf{BC}}^*) \\ \sigma_{\mathbf{BC}}^* \leftarrow \text{NLRepair}_{V, \mathcal{A}, \Pi_{\mathbf{B}}, \Pi_{\mathbf{C}}}(\sigma_{\mathbf{BC}}, y_{\mathbf{B}}, p_{\mathbf{B}}^*, y_{\mathbf{C}}, p_{\mathbf{C}}^*, 1^{1/\varepsilon}, 1^k) \\ (\rho_{\mathbf{BC}}^{**}, p_{\mathbf{B}}^{**}, p_{\mathbf{C}}^{**}) \leftarrow \text{NLValEst}_{V, \mathcal{A}}(\rho_{\mathbf{BC}}^*, 1^{1/\varepsilon}) \end{array} \right] \leq \varepsilon. \quad (6)$$

Proof. We start with value estimation. Let $\mathcal{M}_0^{\mathbf{B}} = (\Pi_A, I - \Pi_A)$ and $\mathcal{M}_1^{\mathbf{B}} = (\Pi_B, I - \Pi_B)$ be the projective measurements applied by $\text{ValEst}_{V, \mathcal{B}}(\cdot, 1^{\varepsilon/2})$. Using Lemma 5, we decompose the \mathbf{B} register into Jordan subspaces as $\mathcal{H}_{\mathbf{B}} = \bigoplus S_j$. Let $\{|v_{j,1}^{\mathbf{A}}\rangle, |v_{j,0}^{\mathbf{A}}\rangle\}$ and $\{|v_{j,1}^{\mathbf{B}}\rangle, |v_{j,0}^{\mathbf{B}}\rangle\}$ be the orthogonal eigenbases of S_j with respect to Π_A, Π_B .

Similarly for $\text{ValEst}_{V, \mathcal{C}}(\cdot, 1^{\varepsilon/2})$, we consider projective measurements $\mathcal{M}_0^{\mathbf{C}} = (\Gamma_A, I - \Gamma_A)$ and $\mathcal{M}_1^{\mathbf{C}} = (\Gamma_B, I - \Gamma_B)$, with the Jordan decomposition $\mathcal{H}_{\mathbf{C}} = \bigoplus T_{j'}$ and the eigenbases $\{|w_{j',1}^{\mathbf{A}}\rangle, |w_{j',0}^{\mathbf{A}}\rangle\}$ and $\{|w_{j',1}^{\mathbf{B}}\rangle, |w_{j',0}^{\mathbf{B}}\rangle\}$. For simplicity, we will assume $\dim S_j = \dim T_{j'} = 2$ for all j, j' . The degenerate cases can be handled similarly³¹.

We will need the following claim:

³⁰Here, by efficient we mean that the runtime of NLValEst is polynomial in the size of $\rho, 1/\varepsilon$, the runtime of V and the runtime of \mathcal{A} . Similarly for NLRepair.

³¹See [CMSZ21] Section 4.1 to see how to handle degenerate subspaces generically.

Claim 2. Let $\Pi_{\mathbf{B}}^{\text{Jor}} = \left\{ \Pi_{\mathbf{B},j}^{\text{Jor}} \right\}_j$ be the jordan subspace measurement, with $\Pi_{\mathbf{B},j}^{\text{Jor}} = |v_{j,0}^A\rangle\langle v_{j,0}^A| + |v_{j,1}^A\rangle\langle v_{j,1}^A|$. Similarly define $\Pi_{\mathbf{C}}^{\text{Jor}}$. Then, $\Pi_{\mathbf{B}}^{\text{Jor}}$ commutes with $\text{ValEst}_{V,\mathbf{B}}(\cdot, 1^{\varepsilon/2})$. Similarly, $\Pi_{\mathbf{C}}^{\text{Jor}}$ commutes with $\text{ValEst}_{V,\mathbf{C}}(\cdot, 1^{\varepsilon/2})$.

Proof. We will show that $\Pi_{\mathbf{B}}^{\text{Jor}}$ commutes with $\text{ValEst}_{V,\mathbf{B}}(\cdot, 1^{\varepsilon/2})$. Recall that $\Pi_{\mathbf{B}}^{\text{Jor}}$ commutes with both $\mathcal{M}_0^{\mathbf{B}}$ and $\mathcal{M}_1^{\mathbf{B}}$. We write the register \mathbf{B} as two registers, \mathbf{P} and \mathbf{Q} , where \mathbf{P} stores the input state and \mathbf{Q} contains all auxiliary registers used in the computation. In particular, this means that the measurements $\mathcal{M}_0^{\mathbf{B}}, \mathcal{M}_1^{\mathbf{B}}, \Pi_{\mathbf{B}}^{\text{Jor}}$ are all restricted to the \mathbf{P} register. By Lemma 10 bullet 3, every step of $\text{ValEst}_{V,\mathbf{B}}(\cdot, 1^{\varepsilon/2})$ falls into one of the following categories:

1. Apply $\mathcal{M}_b^{\mathbf{B}}$ on the \mathbf{P} register conditioned on a qubit located in the \mathbf{Q} register for some $b \in \{0, 1\}$. This captures all steps adaptively applying a projective measurement $\mathcal{M}_b^{\mathbf{B}}$.
2. Apply a local operation on the \mathbf{Q} register. This captures the steps computing the database of outcomes and the functions f, g described in Lemma 10.

Because $\Pi_{\mathbf{B}}^{\text{Jor}}$ is a projection applied to the \mathbf{P} register alone, it commutes with operations from both categories above. \square

Without loss of generality assume that ρ is a pure state, for the mixed state case follows by convexity. By assumption, the bipartite state after steps (a) and (b) described in the 2-Projection Implementation property in Lemma 10 are performed on both registers has the form

$$\sum_{j,j'} \alpha_{j,j'} |v_{j,1}^B\rangle |w_{j',1}^B\rangle$$

Let $\widetilde{\text{ValEst}}$ denote the execution of ValEst after steps (a) and (b) mentioned above. By Claim 2, the product Jordan subspace measurement $\Pi_{\mathbf{B}}^{\text{Jor}} \otimes \Pi_{\mathbf{C}}^{\text{Jor}}$ commutes with $\text{NLValEst}_{V,\mathbf{A}}(\cdot, 1^{1/\varepsilon})$, hence the values $p_{\mathbf{B}}^*, p_{\mathbf{C}}^*$ are distributed as follows:

- Sample (j, j') with probability $|\alpha_{j,j'}|^2$,
- Compute $p_{\mathbf{B}}^* \leftarrow \widetilde{\text{ValEst}}_{V,\mathbf{B}}(|v_{j,1}^B\rangle, 1^{1/(\varepsilon/2)})$ and $p_{\mathbf{C}}^* \leftarrow \widetilde{\text{ValEst}}_{V,\mathbf{C}}(|v_{j',1}^B\rangle, 1^{1/(\varepsilon/2)})$,
- Output $(p_{\mathbf{B}}^*, p_{\mathbf{C}}^*)$.

Let $p_{\mathbf{B}}^j$ be the value of state $|v_{j,1}^B\rangle$ as defined on the RHS of eq. (3). By the same equation, we know that $\mathbb{E}[p_{\mathbf{B}}^*|j] = p_{\mathbf{B}}^j$. Similarly define $p_{\mathbf{C}}^{j'}$ so that $\mathbb{E}[p_{\mathbf{C}}^*|j'] = p_{\mathbf{C}}^{j'}$. Let $p^{j,j'}$ denote the value of state $|v_{j,1}^B\rangle |v_{j',1}^C\rangle$ as defined on the RHS of eq. (5). Because this is pure product state, we have that $p^{j,j'} = p_{\mathbf{B}}^j p_{\mathbf{C}}^{j'}$, and also that the variables $p_{\mathbf{B}}^*, p_{\mathbf{C}}^*$ are independent conditioned on j, j' . On the other hand, since computing the value commutes with the Jordan measurements, the value p for the state ρ is distributed as follows:

- Sample (j, j') with probability $|\alpha_{j,j'}|^2$,
- Output $p^{j,j'}$.

Therefore, putting everything together, we have

$$\begin{aligned}\mathbb{E}[p_{\mathbf{B}}^* p_{\mathbf{C}}^*] &= \sum_{j,j'} |\alpha_{j,j'}|^2 \mathbb{E}[p_{\mathbf{B}}^* p_{\mathbf{C}}^* | j, j'] = \sum_{j,j'} |\alpha_{j,j'}|^2 \mathbb{E}[p_{\mathbf{B}}^* | j] \cdot \mathbb{E}[p_{\mathbf{C}}^* | j'] \\ &= \sum_{j,j'} |\alpha_{j,j'}|^2 p_{\mathbf{B}}^j p_{\mathbf{C}}^{j'} = \sum_{j,j'} |\alpha_{j,j'}|^2 p^{j,j'} = p\end{aligned}$$

as desired. Next, we show almost-projective estimation. Recall that $\text{NLValEst}_{V,\mathcal{A}}(\cdot, 1^{1/\varepsilon})$ applies

$$\left(\text{ValEst}_{V,\mathbf{B}}(\cdot, 1^{1/(\varepsilon/2)}) \otimes \text{ValEst}_{V,\mathbf{C}}(\cdot, 1^{1/(\varepsilon/2)}) \right),$$

and the operations on the \mathbf{B}, \mathbf{C} registers commute. Therefore, using condition (2) of Lemma 10 for the state $\text{Tr}_{\mathbf{C}}(\rho_{\mathbf{BC}})$, we have $\Pr[|p_{\mathbf{B}}^* - p_{\mathbf{B}}^{**}| \geq \varepsilon] \leq \varepsilon/2$, and similarly $\Pr[|p_{\mathbf{C}}^* - p_{\mathbf{C}}^{**}| \geq \varepsilon] \leq \varepsilon/2$. Using a union bound, we achieve the desired property. The repairing property follows by a similar argument, after observing that the procedure in eq. (6) involves two parallel executions of eq. (4) on registers \mathbf{B} and \mathbf{C} . \square

5.3.4 Achieving Non-Local Persistence

In Theorem 4.1 of [BBK22], the output p^* of the value estimation ValEst estimates the value of the solver in expectation, and can be used to achieve a distribution over (p^*, η) -persistent solvers. We observe that the estimation p^* could be stored non-locally $(p_{\mathbf{B}}^*, p_{\mathbf{C}}^*)$ such that $p_{\mathbf{B}}^* p_{\mathbf{C}}^* = p^*$. These local values are enough for non-local state repair by Lemma 19, and using this the proof of [BBK22] can be straightforwardly applied to the non-local setting. We state the formal theorem statement below.

Theorem 20 (Non-Local Persistence Theorem). *Let P be a non-interactive falsifiable assumption with a verifiably polynomial image. Let $\eta = \frac{1}{\text{poly}}$. There exist efficient non-local quantum algorithms $S = S_1 \otimes S_2, R = R_1 \otimes R_2$ with the following syntax and guarantee:*

- $S^{\mathcal{A}}(\text{state}_0) = (S_1^{\mathbf{B}} \otimes S_2^{\mathbf{C}})(\text{state}_0)$ takes as input a non-local algorithm $\mathcal{A} = (\mathbf{B}, \mathbf{C})$ and a bipartite state state_0 . It outputs a state state_0^* and values $(p_{\mathbf{B}}^*, p_{\mathbf{C}}^*)$, where $p_{\mathbf{B}}^*$ is output by $S_1^{\mathbf{B}}$ and $p_{\mathbf{C}}^*$ is output by $S_2^{\mathbf{C}}$.
- $R^{\mathcal{A}}(1^\lambda, 1^i, x_{\mathbf{B}}, x_{\mathbf{C}}, \text{state}_{i-1}^*) = (R_1^{\mathbf{B}}(1^\lambda, 1^i, x_{\mathbf{B}}, \cdot) \otimes R_2^{\mathbf{C}}(1^\lambda, 1^i, x_{\mathbf{C}}, \cdot))(\text{state}_{i-1}^*)$ takes as input a non-local algorithm \mathcal{A} , a security parameter λ , a step i , a pair of inputs $(x_{\mathbf{B}}, x_{\mathbf{C}})$, and a bipartite state state_{i-1}^* . It outputs a pair of solutions $(y_{\mathbf{B}}, y_{\mathbf{C}})$, where $y_{\mathbf{B}}$ is output by $R_1^{\mathbf{B}}$ and $y_{\mathbf{C}}$ is output by $R_2^{\mathbf{C}}$, and a bipartite state state_i^* .

For any non-local independent-challenge P -solver $\mathcal{A} = (\mathbf{B}, \mathbf{C}, \text{state}_0)$ with one-shot value $p = \text{val}_{\tilde{\mathcal{D}}}^{\mathcal{A}}[\mathcal{A}]$, with $\tilde{\mathcal{D}} = \mathcal{U}_{\{0,1\}^d \times \{0,1\}^d}$, considering the random variable $(\text{state}_0^*, p_{\mathbf{B}}^*, p_{\mathbf{C}}^*) \leftarrow S^{\mathcal{A}}(\text{state}_0)$, we have:

1. $\mathbb{E}[p_{\mathbf{B}}^* p_{\mathbf{C}}^*] = p$
2. $\mathcal{R}^* = (R^{\mathcal{A}}, \text{state}_0^*)$ sampled in this process is a distribution over efficient stateful non-local solvers that is $(p_{\mathbf{B}}^* p_{\mathbf{C}}^*, \eta)$ -persistent.

Moreover, if P does not have a verifiably polynomial image, the same holds, but S is not efficient.

Proof. The proof will closely follow the proof of Theorem 4.1 in [BBK22]. Let $P = (\text{GenC}, \text{Ver}, c)$ with polynomial-image verifier K and polynomial-image bound k . Let $\eta = 1/\text{poly}(\cdot)$. We will denote by \mathbf{B}, \mathbf{C} the two registers for the non-local algorithms S, R, \mathcal{A} . For $x \in \{0, 1\}^n$, we define the $(k+1)$ -outcome projective measurement $\Pi_x^{\mathcal{B}}$ as follows:

- Coherently run \mathcal{B} with input x .
- First, measure if the output of \mathcal{B} is in the polynomial image specified by K .
 - If not, output \perp
 - If yes, measure the output of \mathcal{B} and output the answer $y \in \{0, 1\}^m$.
- Uncompute.

We similarly define the projective measurement $\Pi_x^{\mathcal{C}}$. We also define the wrapper non-local solver $\tilde{\mathcal{A}} = \tilde{\mathcal{B}} \otimes \tilde{\mathcal{C}}$, where $\tilde{\mathcal{B}}$ (resp., $\tilde{\mathcal{C}}$) takes as input $r \in \{0, 1\}^d$, computes $x = \text{GenC}(1^\lambda; r)$ and runs \mathcal{B} (resp., \mathcal{C}) on input x , outputting the result $y \in \{0, 1\}^m$. Next, we move on to describe $S^{\mathcal{A}}$ and $R^{\mathcal{A}}$ next.

Description of $S^{\mathcal{A}}$: On input state_0 , compute $(\text{state}_0^*, p_{\mathcal{B}}^*, p_{\mathcal{C}}^*) \leftarrow \text{NLValEst}_{\text{Ver}, \tilde{\mathcal{A}}}(\text{state}_0, 1^{10/\eta})$. That is, $S_1^{\mathcal{B}}$ runs $\text{ValEst}_{\text{Ver}, \tilde{\mathcal{B}}}(\cdot, 1^{20/\eta})$ and $S_2^{\mathcal{C}}$ runs $\text{ValEst}_{\text{Ver}, \tilde{\mathcal{C}}}(\cdot, 1^{20/\eta})$. Output $(\text{state}_0^*, p_{\mathcal{B}}^*, p_{\mathcal{C}}^*)$.

Description of $R^{\mathcal{A}}$: On input $(1^\lambda, 1^i, x_{\mathcal{B}}, x_{\mathcal{C}}, \text{state}_{i-1}^*)$, where R_1 receives $1^\lambda, 1^i, x_{\mathcal{B}}$ and the register \mathbf{B} while R_2 receives $1^\lambda, 1^i, x_{\mathcal{C}}$ and \mathbf{C} , do the following:

- Define $\varepsilon_i = \eta/32i^2$, so that $(\varepsilon_i)_i$ is monotonically decreasing and $\sum_{i=1}^{\infty} \varepsilon_i < \eta/16$.
- Compute $(\sigma_{i-1}, q_{\mathcal{B}}^{i-1}, q_{\mathcal{C}}^{i-1}) \leftarrow \text{NLValEst}_{\text{Ver}, \tilde{\mathcal{A}}}(\text{state}_{i-1}^*, 1^{1/\varepsilon_i})$.
- Compute $(y_{\mathcal{B}}^i, y_{\mathcal{C}}^i, \sigma_i^*) \leftarrow (\Pi_{x_{\mathcal{B}}}^{\mathcal{B}} \otimes \Pi_{x_{\mathcal{C}}}^{\mathcal{C}})(\sigma_{i-1})$, where $y_{\mathcal{B}}^i, y_{\mathcal{C}}^i \in \{0, 1\}^m \cup \{\perp\}$ and σ_i^* is the post-measurement state.
- Compute $\rho_i \leftarrow \text{NLRepair}_{\text{Ver}, \tilde{\mathcal{A}}, \Pi_{x_{\mathcal{B}}}^{\mathcal{B}}, \Pi_{x_{\mathcal{C}}}^{\mathcal{C}}}(\sigma_i^*, y_{\mathcal{B}}^i, q_{\mathcal{B}}^{i-1}, y_{\mathcal{C}}^i, q_{\mathcal{C}}^{i-1}, 1^{1/\varepsilon_i, 1^{k+1}})$.
- Compute $(\text{state}_i^*, p_{\mathcal{B}}^i, p_{\mathcal{C}}^i) \leftarrow \text{NLValEst}_{\text{Ver}, \tilde{\mathcal{A}}}(\rho_i, 1^{1/\varepsilon_i})$.
- Output $(y_{\mathcal{B}}, y_{\mathcal{C}}, \text{state}_i^*)$, where $y_{\mathcal{B}}$, register \mathbf{B} is R_1 's output while $y_{\mathcal{C}}$, register \mathbf{C} is R_2 's output.

Note that R_1 and R_2 are defined implicitly above given that every step is a non-local algorithm; $R_1^{\mathcal{B}}$ performs the operations on the \mathbf{B} register, and $R_2^{\mathcal{C}}$ on the \mathbf{C} register.

Consider a solver-aided non-local algorithm $\mathcal{S}^{\mathcal{R}^*}$ with input z , and the random variables sampled by the process below:

- $(p_{\mathcal{B}}^*, p_{\mathcal{C}}^*, \text{state}_0^*) \leftarrow S^{\mathcal{A}}(\text{state}_0)$. Equivalently, $(\text{state}_0^*, p_{\mathcal{B}}^*, p_{\mathcal{C}}^*) \leftarrow \text{NLValEst}_{\text{Ver}, \tilde{\mathcal{A}}}(\text{state}_0, 1^{2/\eta})$.
- For $i \geq 1$, $(y_{\mathcal{B}}^i, y_{\mathcal{C}}^i, \text{state}_i^*) \leftarrow R^{\mathcal{A}}(1^\lambda, 1^i, x_{\mathcal{B}}^i, x_{\mathcal{C}}^i, \text{state}_{i-1}^*)$, where $(x_{\mathcal{B}}^i, x_{\mathcal{C}}^i)$ is the i th pair of inputs chosen by $\mathcal{S}_z^{\mathcal{R}^*}$. In more detail,

- $(\sigma_{i-1}, q_{\mathcal{B}}^{i-1}, q_{\mathcal{C}}^{i-1}) \leftarrow \text{NLValEst}_{\text{Ver}, \tilde{\mathcal{A}}}(\text{state}_{i-1}^*, 1^{1/\varepsilon_i})$.
- $(y_{\mathcal{B}}^i, y_{\mathcal{C}}^i, \sigma_i^*) \leftarrow (\Pi_{x_{\mathcal{B}}}^{\mathcal{B}} \otimes \Pi_{x_{\mathcal{C}}}^{\mathcal{C}})(\sigma_{i-1})$.
- $\rho_i \leftarrow \text{NLRepair}_{\text{Ver}, \tilde{\mathcal{A}}, \Pi_{x_{\mathcal{B}}}^{\mathcal{B}}, \Pi_{x_{\mathcal{C}}}^{\mathcal{C}}}(\sigma_i^*, y_{\mathcal{B}}^i, q_{\mathcal{B}}^{i-1}, y_{\mathcal{C}}^i, q_{\mathcal{C}}^{i-1}, 1^{1/\varepsilon_i}, 1^{k+1})$.
- $(\text{state}_i^*, p_{\mathcal{B}}^i, p_{\mathcal{C}}^i) \leftarrow \text{NLValEst}_{\text{Ver}, \tilde{\mathcal{A}}}(\rho_i, 1^{1/\varepsilon_i})$.

Over the randomness of this process, we have the following identities:

$$\mathbb{E}[p_{\mathcal{B}}^* p_{\mathcal{C}}^*] = p \tag{7}$$

$$\Pr[|p_{\mathcal{B}}^* - q_{\mathcal{B}}^0| \geq \eta/10 \vee |p_{\mathcal{C}}^* - q_{\mathcal{C}}^0| \geq \eta/10] \leq \eta/10 \tag{8}$$

$$\Pr[|p_{\mathcal{B}}^i - q_{\mathcal{B}}^i| \geq \varepsilon_i \vee |p_{\mathcal{C}}^i - q_{\mathcal{C}}^i| \geq \varepsilon_i] \leq \varepsilon_i, \quad \forall i \geq 1 \tag{9}$$

$$\Pr[|q_{\mathcal{B}}^{i-1} - p_{\mathcal{B}}^i| \geq \varepsilon_i \vee |q_{\mathcal{C}}^{i-1} - p_{\mathcal{C}}^i| \geq \varepsilon_i] \leq \varepsilon_i, \quad \forall i \geq 1 \tag{10}$$

$$\mathbb{E}[q_{\mathcal{B}}^{i-1} q_{\mathcal{C}}^{i-1}] = \text{val}_{\tilde{P}}^{\tilde{D}}[i-1, \mathcal{S}_z^{\mathcal{R}^*}], \quad \forall i \geq 1 \tag{11}$$

$$\Pr\left[\left|p_{\mathcal{B}}^{i-1} p_{\mathcal{C}}^{i-1} - \text{val}_{\tilde{P}}^{\tilde{D}}[i-1, \mathcal{S}_z^{\mathcal{R}^*}]\right| \geq \varepsilon_i\right] \leq \varepsilon_i, \quad \forall i \geq 1, \mathcal{S}, z \tag{12}$$

Equation (7) follows from the value estimation property in Lemma 19 and the fact that $\tilde{\mathcal{A}}$ is a wrapper solver for \mathcal{A} . This is the first bullet we need to show to prove the theorem.

Moving on to the second bullet (persistence), eqs. (8) and (9) follow from the almost-projective estimation property in Lemma 19 since $\varepsilon \leq \eta/2$ and $\varepsilon_{i+1} \leq \varepsilon_i$. Equation (10) follows from the repairing property. Finally, eq. (12) follows from the value estimation property and from the fact that³² $\text{val}_{\tilde{P}}^{\tilde{D}}[i-1, \mathcal{S}_z^{\mathcal{R}^*}] = \Pr[\text{Ver}(1^\lambda, r_{\mathcal{B}}^i, y_{\mathcal{B}}^i) = \text{Ver}(1^\lambda, r_{\mathcal{C}}^i, y_{\mathcal{C}}^i) = 1]$. The latter is true because $y_{\mathcal{B}}^i$ (resp., $y_{\mathcal{C}}^i$) is distributed the same as the output of \mathcal{B} (resp., \mathcal{C}) conditioned on $y_{\mathcal{B}}^i \neq \perp$, and otherwise the verification fails as expected.

Therefore, by union bound we have with probability at least $1 - \eta/10 - \sum_{i=1}^{\infty} 4\varepsilon_i \geq 1 - \eta$ that

1. $|p_{\mathcal{B}}^* - q_{\mathcal{B}}^0| \leq \eta/10$ and $|p_{\mathcal{C}}^* - q_{\mathcal{C}}^0| \leq \eta/10$,
2. $|p_{\mathcal{B}}^i - q_{\mathcal{B}}^i| \leq \varepsilon_i$ and $|p_{\mathcal{C}}^i - q_{\mathcal{C}}^i| \leq \varepsilon_i$,
3. $|p_{\mathcal{B}}^i - q_{\mathcal{B}}^{i-1}| \leq \varepsilon_i$ and $|p_{\mathcal{C}}^i - q_{\mathcal{C}}^{i-1}| \leq \varepsilon_i$,
4. $\mathbb{E}[q_{\mathcal{B}}^{i-1} q_{\mathcal{C}}^{i-1}] = \text{val}_{\tilde{P}}^{\tilde{D}}[i-1, \mathcal{S}_z^{\mathcal{R}^*}]$

for all $i \geq 1$. Conditioned on the inequalities 1-4 above being true, we will show that

$$\left|p_{\mathcal{B}}^* p_{\mathcal{C}}^* - \text{val}_{\tilde{P}}^{\tilde{D}}[i, \mathcal{S}_z^{\mathcal{R}^*}]\right| \leq \eta \tag{13}$$

to finish the proof. Indeed, by triangle inequality we have that $|p_{\mathcal{B}}^* - q_{\mathcal{B}}^i| \leq \eta/10 + \sum_{j=1}^i 2\varepsilon_j \leq \eta/3$, and similarly $|p_{\mathcal{C}}^* - q_{\mathcal{C}}^i| \leq \eta/3$. Since $p_{\mathcal{B}}^*, p_{\mathcal{C}}^* \in [0, 1]$, this implies that $|p_{\mathcal{B}}^* p_{\mathcal{C}}^* - q_{\mathcal{C}}^i q_{\mathcal{B}}^i| \leq 2(\eta/3) + (\eta/3)^2 \leq \eta$. Taking the expectation, we obtain eq. (13).

³²Here we define $\text{Ver}(1^\lambda, r, \perp) = 0$ for convenience.

If P does not have a verifiably polynomial image, note that we can always take $k = 2^n$, i.e. P always has a verifiably *exponential* image. Thus, the proof remains the same, except the runtime of R now depends exponentially on n . □

Remark 7. *We note that the transformation in Theorem 20 is not optimal in terms of concrete efficiency. For instance, it is redundant to apply value estimation once in S and twice in R . The reason we do this is to use previous work as closely to a black-box as possible, thus keeping the proof simple.*

5.3.5 Non-Local Classical-to-Quantum Lifting Theorem

The remaining steps to reach our non-local lifting theorem (Theorem 22) are essentially the same as those in [BBK22]. For this reason, we do not repeat the detailed proofs here. Nonetheless, we note that the proof has the following (informal) structure: given a P -solver,

1. Obtain a persistent P -solver.
2. Next, obtain a memoryless P -solver, i.e. one that only remembers the number of times it has interacted with the assumption.
3. Next, obtain a stateless solver.

Step 1 corresponds to Theorem 20, whereas steps 2,3 can be summarized below:

Theorem 21. *There exists an efficient non-local quantum algorithm $\text{Sim} = \text{Sim}_1 \otimes \text{Sim}_2$ with the following properties. Let $\mathcal{A} = (\mathcal{B}, \mathcal{C}, \text{state}_0)$ be a (p, η) -persistent $(\ell, \ell_{\mathcal{B}}, \ell_{\mathcal{C}})$ -stateful non-local independent-challenge P -solver for a falsifiable non-interactive assumption P . Let $\vec{x}_{\mathcal{B}}$ and $\vec{x}_{\mathcal{C}}$ each be (independently) sampled from an efficiently samplable distribution over k -tuples of P -instances and define $\vec{x} := (\vec{x}_{\mathcal{B}}, \vec{x}_{\mathcal{C}})$. Then, there exists a (p, η) -persistent distribution $\{\mathcal{A}^\alpha = (\mathcal{B}^\alpha, \mathcal{C}^\alpha)\}_\alpha$ over stateless solvers such that*

$$\text{Sim}^{\mathcal{A}}(1^\lambda, 1^\ell, 1^{1/\delta}, \vec{x}^*) = \text{Sim}_1^{\mathcal{B}}(1^\lambda, 1^{\ell_{\mathcal{B}}}, 1^{1/\delta}, \vec{x}_{\mathcal{B}}^*) \otimes \text{Sim}_2^{\mathcal{C}}(1^\lambda, 1^{\ell_{\mathcal{C}}}, 1^{1/\delta}, \vec{x}_{\mathcal{C}}^*)$$

makes non-adaptive black-box queries to \mathcal{B} and \mathcal{C} and produces a distribution within δ statistical distance from $\mathcal{A}^\alpha(1^\lambda, \vec{x})$.

Proof (sketch). Follows from Corollary 6.2 in [BBK22] after making the observation that the simulators in Theorem 5.1 and Theorem 6.1 both preserve non-locality, i.e. they are non-local simulators if given as input non-local solvers. □

The theorem below is a direct consequence of Theorems 20 and 21 as per Theorem 7.1 of [BBK22]. We refer the reader to [BBK22] for the proof.

Theorem 22 (Non-Local Classical-to-Quantum Lifting). *Let $P = (\text{GenC}_P, \text{Ver}_P, c_P)$ and $Q = (\text{GenC}_Q, \text{Ver}_Q, c_Q)$ be non-interactive assumptions. Assume there exists a non-adaptive non-local classical black-box $(\tilde{\mathcal{D}}_P, c_P, \tilde{\mathcal{D}}_Q, c_Q)$ -reduction from solving Q to solving P , where $\tilde{\mathcal{D}}_P = \mathcal{U}_{\{0,1\}^{d_P} \times \{0,1\}^{d_P}}$ is the uniform extension. Then, there exists an inefficient non-local quantum $(\tilde{\mathcal{D}}_P, c_P, \tilde{\mathcal{D}}_Q, c_Q)$ -reduction from solving Q to solving P . If P has verifiably polynomial image, then the reduction is efficient.*

5.4 Applications

In this section, we present two applications of our main theorem (Theorems 13 and 14). The first one is information theoretic single-decryptor encryption for one-bit messages, which relies on simultaneous quantum Goldreich-Levin extraction. The second application involves the effect of changing the challenge distribution in a cloning game to unclonable security. Informally, we show that independent-challenge unclonable security implies unclonable security for correlated distributions for primitives which have negligible trivial success probability.

5.4.1 Single-Decryptor Encryption

In this section, we analyze a corollary of Theorem 14, which is Simultaneous Quantum Goldreich-Levin Lemma. Then, we show a construction of information theoretic single-decryptor encryption as a corollary.

Definition 33 (Simultaneous Extraction). *We say that an adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ can simultaneously extract a classical function $f(k, x, r)$ given a quantum token $\rho_{k,x}$ using key $k = (k_B, k_C)$ with probability δ if $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in the following experiment with probability δ :*

- In phase 1, the challenger sends $\rho_{k,x}$ to \mathcal{A} , who applies a CPTP map to split the state into two registers \mathcal{B} and \mathcal{C} .
- In phase 2, \mathcal{B} and \mathcal{C} can no longer communicate. The challenger samples independent random coins $(r, r') \xleftarrow{\$} \mathcal{R} \times \mathcal{R}$, then sends (r, k_B) to \mathcal{B} and (r', k_C) to \mathcal{C} . Later, \mathcal{B} outputs a string z_B and \mathcal{C} outputs a string z_C .
- The adversary wins if $z_B = f(k, x, r)$ and $z_C = f(k, x, r')$.

It is understood that x above is a random variable.

Lemma 23 (Simultaneous Quantum Goldreich-Levin). *Suppose that $\mathcal{R} = \{0, 1\}^n$ and there exists a (possibly inefficient) adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which can simultaneously extract $\langle r, x \rangle$ from a quantum token $\rho_{k,x}$ using key k , with probability $1/2 + \varepsilon$ for a non-negligible function ε . Then, there exists an adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ which can simultaneously extract³³ x given the same token $\rho_{k,x}$ using key k with non-negligible probability ε' .*

Proof. We will interpret the key $k = (k_B, k_C) =: (\text{ch}_B, \text{ch}_C)$ as the challenge and $x =: m$ as the message in a cloning game. We will assume here that k_B, k_C are independently distributed, which is sufficient for the application to single-decryptor encryption. See Appendix A for a direct proof³⁴ of Lemma 23 which covers the general case. Let $\mathcal{G}' = (\text{Setup}, \text{GenT}, \text{GenC}', \text{Ver}')$ be a (stateful) cloning game, such that:

- $\rho_{k,x}$ is the token received by \mathcal{A} , x is the message, and k is the challenge received by \mathcal{B} and \mathcal{C} .
- $\text{Ver}'(\text{sk}, x, \text{ch}, \text{ans})$ accepts if and only if $\text{ans} = x$, i.e. \mathcal{G} is a search game.

³³We note that the values (r, r') can be ignored when extracting x .

³⁴The direct proof also shows that if the simultaneous extractor for $\langle r, x \rangle$ is efficient, then so is the simultaneous extractor for x .

Now define a new cloning game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$, where:

- $\text{GenC}(\text{sk}, x)$ computes $\text{ch} \leftarrow \text{GenC}(\text{sk}, x)$ and samples $r \leftarrow \mathcal{U}_{\mathcal{R}}$. It outputs (r, ch) .
- $\text{Ver}(\text{sk}, x, (r, \text{ch}), \text{ans})$ accepts if and only if $\text{ans} = \langle r, x \rangle$.

Suppose that the distribution of x , denoted by $\mathcal{D}_{\mathcal{M}}$, is non-trivial in the sense that $p^{\text{triv}}(\mathcal{G}, \mathcal{D}_{\mathcal{M}}) \leq 1/2 + \text{negl}(\lambda)$ and \mathcal{G}' is $\mathcal{D}_{\mathcal{M}}$ -evasive, for otherwise the proof is trivial.

We will use the contrapositive of Theorem 14, with $\widetilde{\text{GenC}} = \text{GenC}_{\text{ind}}$. Now if $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ exists, then \mathcal{G} does not have information theoretic $(\mathcal{D}_{\mathcal{M}}, \text{negl})$ independent-challenge unclonable security, where $\mathcal{M} = \{0, 1\}^n$ is the message space and $\mathcal{D}_{\mathcal{M}}$ denotes the distribution of x . Now we observe that the conditions of Theorem 14, i.e. conditions in items (1) to (3) and item (b) of Theorem 13, are all satisfied. Indeed, items (1) to (3) are satisfied by assumption. Item (b) follows from the well-known classical local Goldreich-Levin extraction³⁵ [GL89]. Therefore, it follows that \mathcal{G} does not have $(\mathcal{D}_{\mathcal{M}}, \text{negl})$ information theoretic independent-challenge unclonable security, meaning there exists $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ which can simultaneously extract x with non-negligible probability as desired. \square

Search to Decision Transformation.

Corollary 4 (Search to Decision). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning search game with extension $\widetilde{\text{GenC}}$ such that (1) $\text{GenC}(\text{sk}, m)$ and $\widetilde{\text{GenC}}(\text{sk}, m)$ both do not depend on m , (2) \mathcal{G} has statistical correctness, and (3) \mathcal{G} has $(\mathcal{D}, \varepsilon, \widetilde{\text{GenC}})$ unclonable security. Define a (stateful) cloning decision game $\mathcal{G}' = (\text{Setup}', \text{GenT}', \text{GenC}', \text{Ver}')$ as follows:*

- *There is no message, i.e. $\mathcal{M}' = \{\perp\}$ is the message space.*
- *$\text{Setup}'(1^\lambda)$ computes $\text{sk} \leftarrow \text{Setup}(1^\lambda)$. $\text{sk}' = \text{sk}$.*
- *$\text{GenT}'(\text{sk}', m')$ parses the input as $\text{sk}' = \text{sk}$, samples $m \leftarrow \mathcal{D}$, and computes $\rho \leftarrow \text{GenT}(\text{sk}, m)$. It outputs a token ρ and random coins $r_{\text{GenT}'} = m$.*
- *$\text{GenC}'(\text{sk}', m'; r_{\text{GenC}'})$ parses the input as $\text{sk}' = \text{sk}$. It interprets the random coins as $r_{\text{GenC}'} = (r, b, r_{\text{GenC}})$, where $r \in \mathcal{M}$, $b \in \{0, 1\}$, and r_{GenC} is randomness for GenC . Then it computes $\text{ch} \leftarrow \text{GenC}(\text{sk}, m; r_{\text{GenC}})$, and outputs $\text{ch}' = (r, \langle r, m \rangle \oplus b, \text{ch})$.*
- *$\text{Ver}'(\text{sk}', m', \text{ch}', \text{ans}', r_{\text{GenC}'})$ parses the input as $r_{\text{GenC}'} = (r, b, r_{\text{GenC}})$ and outputs $b' = [\text{ans}' == b]$.*

Then, \mathcal{G}' has statistical correctness and $(\mathcal{D}_{\perp}, \sqrt{\varepsilon}/2, \widetilde{\text{GenC}}')$ independent-challenge unclonable security, where the extension $\widetilde{\text{GenC}}'(\text{sk}, m')$ is defined as follows:

1. *Run $(r_B, r_C) \leftarrow \text{GenC}(\text{sk}, m)$*
2. *Sample $r'_B, r'_C \xleftarrow{\$} \mathcal{M}$ and $b_B, b_C \xleftarrow{\$} \{0, 1\}$*
3. *Output $((r'_B, b_B, r_B), (r'_C, b_C, r_C))$.*

³⁵Note that even though the success probability of the extractor may depend on x , its description does not.

In particular, if \mathcal{G} has $(\mathcal{D}, \varepsilon)$ independent-challenge unclonable security, then \mathcal{G}' has $(\mathcal{D}_\perp, \sqrt{\varepsilon}/2)$ independent-challenge unclonable security.

Proof. We construct the game $\mathcal{G}' = (\text{Setup}', \text{GenT}', \text{GenC}', \text{Ver}')$ as follows:

We define several hybrids:

- **Hybrid 0:** This is the cloning experiment $\mathfrak{CE}_{\mathcal{G}', \mathcal{D}_\perp}^{\widetilde{\text{GenC}'}}$. Since \mathcal{G}' is a decision game, we have $p^{\text{triv}}(\mathcal{G}', \mathcal{D}', \widetilde{\text{GenC}'}) \geq 1/2$. Assume for the sake of contradiction that the statement is false, then there exists a cloning adversary $(\mathcal{A}_0, \mathcal{B}_0, \mathcal{C}_0)$ that succeeds in this experiment with probability $p > \frac{1}{2} + \sqrt{\varepsilon}/2$.
- **Hybrid 1:** In this hybrid, we modify the success condition for the adversary. Instead of outputting the bits b_B and b_C , \mathcal{B} and \mathcal{C} are now required to output $\langle r, m \rangle$ and $\langle r', m \rangle$, respectively, where (r, r') are the random coins generated for $\text{ch}'_{\mathcal{B}}$ and $\text{ch}'_{\mathcal{C}}$, respectively. This hybrid is clearly equivalent to Hybrid 0, since $\langle r, m \rangle \oplus b_B$ and $\langle r', m \rangle \oplus b_C$ are known to \mathcal{B} and \mathcal{C} , respectively. Thus, there exists $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1)$ which succeeds in this hybrid with probability p . Specifically, \mathcal{B}_1 and \mathcal{C}_1 simply run \mathcal{B}_0 and \mathcal{C}_0 ; then they XOR the output with the value mentioned above.
- **Hybrid 2:** In this hybrid, we truncate the challenges given to \mathcal{B}' and \mathcal{C}' . Specifically, instead of getting $\text{ch}'_{\mathcal{B}} = (r, \langle r, m \rangle \oplus b_B, \text{ch}_{\mathcal{B}})$, \mathcal{B} will receive $(r, \text{ch}_{\mathcal{B}})$; similarly \mathcal{C} will receive $(r', \text{ch}_{\mathcal{C}})$. This hybrid is equivalent to Hybrid 1, so there exists an adversary $(\mathcal{A}_2, \mathcal{B}_2, \mathcal{C}_2)$ which succeeds in this hybrid with probability p . The reason is that b_B, b_C are uniformly random bits that information theoretically hide the inner products $\langle r, m \rangle$ and $\langle r', m \rangle$. In more detail, $(\mathcal{A}_2, \mathcal{B}_2, \mathcal{C}_2)$ can be constructed as follows:

- Upon receiving a token ρ , \mathcal{A}_2 runs $\rho_{BC} \leftarrow \mathcal{A}_1$ and samples random bits $b'_B, b'_C \xleftarrow{\$} \{0, 1\}$. Then it sends the bipartite state $\rho_{BC} \otimes |b'_B\rangle\langle b'_B|_B \otimes |b'_C\rangle\langle b'_C|_C$ to \mathcal{B}_2 and \mathcal{C}_2 .
- \mathcal{B}_2 , upon receiving $(r, \text{ch}_{\mathcal{B}})$ from the challenger and a state $\rho_B \otimes |b'_B\rangle\langle b'_B|_B$ from \mathcal{A}_2 , runs \mathcal{B}_1 with input $(r, b'_B, \text{ch}_{\mathcal{B}})$. \mathcal{C}_2 is defined similarly.

Since the view of $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1)$ exactly matches Hybrid 1, we conclude that $(\mathcal{A}_2, \mathcal{B}_2, \mathcal{C}_2)$ succeeds in Hybrid 2 with probability $p > \frac{1}{2} + \sqrt{\varepsilon}/2$.

By Lemma 23, this implies that there exists a cloning adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ which succeeds in $\mathfrak{CE}_{\mathcal{G}', \mathcal{D}}^{\widetilde{\text{GenC}'}}$ with probability greater than $4(\sqrt{\varepsilon}/2)^2 = \varepsilon$, a contradiction. □

By plugging in $\mathcal{G} = \mathcal{G}_{\text{BB84}}$ from Definition 21, we get the following corollary:

Corollary 5. *There exists a single-decryptor encryption scheme in the plain model with information-theoretic independent-challenge security.*

Proof. It could easily be checked that the resulting game \mathcal{G}' is the cloning game corresponding to the following single-decryptor encryption scheme $(\text{Gen}, \text{GenT}, \text{Enc}, \text{Dec})$:

1. $(\text{Gen}(1^\lambda))$: Sample $m, \theta \xleftarrow{\$} \{0, 1\}^\lambda$. Output $\text{sk} = (m, \theta)$

2. $\text{GenT}(\text{sk})$: Parse the input as $\text{sk} = (m, \theta)$ and output the decryption token $|m^\theta\rangle\langle m^\theta|$.
3. $\text{Enc}(\text{sk}, b)$: Sample $r \xleftarrow{\$} \{0, 1\}^\lambda$. Output $ct = (r, \langle r, m \rangle \oplus b, \theta)$
4. $\text{Dec}(\rho, ct)$: Parse $ct = (r, b', \theta)$. Measure ρ in basis H^θ to get m . Output $\langle r, m \rangle \oplus b'$.

Note that since $\mathcal{G}_{\text{BB84}}$ is a cloning encryption game, it is secure against any extension, including the independent-challenge case. Therefore, the single-decryptor scheme above is optimally secure against independent challenges by Corollary 4. \square

5.4.2 Relationship Between Challenge Distributions

As a second application, we show that when the trivial success probability of a cloning game is negligible, unclonable security for independent challenges implies unclonable security for any challenge distribution, up to a polynomial loss in the success probability.

Corollary 6. *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a \mathcal{D} -evasive cloning game with $(\mathcal{D}, \text{negl})$ independent-challenge unclonable security. Suppose that for any key-message pair (sk, m) , the non-interactive assumption induced by $(\mathcal{G}, \mathcal{D}, \text{sk}, m)$ has verifiably polynomial image. Then, \mathcal{G} has $(\mathcal{D}, \text{negl}, \widetilde{\text{GenC}})$ unclonable security for any extension $\widetilde{\text{GenC}}$. In particular, \mathcal{G} has $(\mathcal{D}, \text{negl})$ identical-challenge unclonable security.*

Proof. Follows directly from Theorem 13 by setting $\mathcal{G} = \mathcal{G}'$. The only non-trivial condition is (b), which follows by Corollary 3. Note that the classical reduction is the identity reduction, hence it does not depend on (sk, m) . \square

Remark 8. *The corollary above can be applied to any unclonable primitive with verifiably polynomial image, where p^{triv} is negligible, including copy-protection for functions with output-length $\omega(\log \lambda)$.*

Direct Proof with Concrete Bounds. We can in fact show a stronger statement without using the main theorem, namely that the identity reduction works in Corollary 6 with a cubic loss in success probability.

Alternate proof of Corollary 6. Follows directly from Lemma 18 after setting \mathcal{D} in the lemma statement to be the output of GenC' , setting $\widetilde{\mathcal{D}}$ to be the output of the extension $\widetilde{\text{GenC}}$, and setting B_r (C_r) to be a POVM element which tests whether \mathcal{B} (\mathcal{C}) passes the verification Ver on challenge r . \square

6 Relating Unclonable-Search and Unclonable-Indistinguishability

In this section, we will give a relationship between games satisfying unclonable-indistinguishability security Definition 11 and unclonable-search security Definition 10.

In Section 6.1, we show that games with unclonable-search property imply games satisfying unclonable-indistinguishability property. In Section 6.2, we show that games with unclonable-indistinguishability property imply games satisfying unclonable-search property.

We start with some simple observations before moving onto our results.

Trivial Success. We show upper and lower bounds for trivial success probability of search games.

Lemma 24 (Trivial Success Probability of Search Games). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning search game with correctness δ and let $\widetilde{\text{GenC}}$ be an extension. Then,*

$$(1 - \sqrt{1 - \delta}) \left(\text{OPT}(m \mid \text{ch}) - \sqrt{1 - \delta} \right) \leq p^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) \leq \text{OPT}(m \mid \text{ch}).$$

where the variables (m, ch) are sampled as in Definition 2.

Proof Sketch. Consider a \mathcal{B} -trivial attack. For the upper-bound, simply bound the success probability of \mathcal{B} . For the lower bound, note that for $(1 - \sqrt{1 - \sigma})$ fraction of the time, \mathcal{C} will succeed with probability at least $1 - \sqrt{1 - \sigma}$ by correctness. Observe that \mathcal{B} can succeed with probability $\text{OPT}(m \mid \text{ch}_{\mathcal{B}}, \text{aux})$ without the token. Use union bound to complete the proof. Similar for the \mathcal{C} -trivial attack. \square

Corollary 7. *If \mathcal{G} is a search game with perfect correctness, then $p^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) = \text{OPT}(m \mid \text{ch})$.*

Message Hiding. Clearly, a search game must have the property that the token hides the message, for otherwise cloning would be trivial. We formalize this in the lemma below.

Lemma 25 (Message Hiding). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning search game with $(\mathcal{D}, \varepsilon)$ unclonable security, then for any QPT adversary \mathcal{A} we have*

$$\Pr \left[m \leftarrow \mathcal{A}(1^\lambda, \rho) : \begin{array}{l} \text{sk} \leftarrow \text{Setup}(1^\lambda) \\ m \leftarrow \mathcal{D} \\ \rho \leftarrow \text{GenT}(\text{sk}, m) \end{array} \right] \leq p^{\text{triv}}(\mathcal{D}, \varepsilon) + \varepsilon(\lambda)$$

Proof. If the statement is false, then \mathcal{A} can send m to \mathcal{B} and \mathcal{C} in the splitting phase, both of which output m as their answer, hence $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ breaks the $(\mathcal{D}, \varepsilon)$ unclonable security of \mathcal{G} . \square

6.1 Search-to-Indistinguishability

We transform games satisfying unclonable-search property to games satisfying unclonable-indistinguishability property in the following steps.

1. In Section 6.1.1, we present a generic transformation that reduces the unclonable-search security of cloning games, where the message distribution comes from a high entropy distribution, to unclonable-search security, where the message distribution is uniform.
2. In Section 6.1.2, we consider a new notion of security called augmented unclonable security. In this security notion, all the adversaries have oracle access to a point function $P_m(\cdot)$, where m is such that the adversary receives a token generated using m . We show that we can generically transform any game satisfying unclonable-search security into one satisfying augmented unclonable-search security.
3. In Section 6.1.3, we show how augmented unclonable security implies unclonable-indistinguishability security.

6.1.1 Relationship Between Message Distributions

The following lemma is adapted from Theorem 9 in [BL20].

Lemma 26. *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning game. Let \mathcal{D} be a distribution over the message space \mathcal{M} with min-entropy h , then for any cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and any extension $\widetilde{\text{GenC}}$ of GenC , we have*

$$\Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq 2^{\log_2 |\mathcal{M}| - h} \cdot \Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{U}_{\mathcal{M}}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right].$$

Proof.

$$\begin{aligned} \Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] &= \sum_{m \in \mathcal{M}} \Pr [m \leftarrow \mathcal{D}] \cdot \Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \mid m \right] \\ &\leq \sum_{m \in \mathcal{M}} 2^{-h} \cdot \Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \mid m \right] \\ &= 2^{\log_2 |\mathcal{M}| - h} \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{M}|} \cdot \Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{U}_{\mathcal{M}}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \mid m \right] \\ &= 2^{\log_2 |\mathcal{M}| - h} \cdot \Pr \left[1 \leftarrow \text{cc}_{\mathcal{G}, \mathcal{U}_{\mathcal{M}}}^{\widetilde{\text{GenC}}} (1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \end{aligned}$$

□

6.1.2 Generically Augmenting Security

Definition 34 (Augmented Unclonable Security). *Let \mathcal{G} be a cloning search game and \mathcal{D} be a distribution over the message space \mathcal{M} . Let $\mathfrak{A}\text{cc}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$ be the following augmented cloning experiment, with the modification highlighted in blue:*

- **Setup Phase:**

- All parties get a security parameter 1^λ as input.
- Ref samples a message $m \leftarrow \mathcal{D}$.
- Ref computes $\text{sk} \leftarrow \text{Setup}(1^\lambda)$ and $\rho \leftarrow \text{GenT}(\text{sk}, m)$.
- \mathcal{R} sends ρ to \mathcal{A} .
- $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ all get oracle access to the point function $P_m(\cdot)$.

- **Splitting Phase:**

- \mathcal{A} computes a bipartite state ρ' over registers B, C .
- \mathcal{A} sends $\rho'[B]$ to \mathcal{B} and $\rho'[C]$ to \mathcal{C} .

- **Challenge Phase:**

- Ref samples $(r_B, r_C) \leftarrow \widetilde{\text{GenC}}(\text{sk}, m)$ and computes $\text{ch}_B = \widetilde{\text{GenC}}(\text{sk}, m; r_B)$, $\text{ch}_C = \widetilde{\text{GenC}}(\text{sk}, m; r_C)$
- Ref sends ch_B to \mathcal{B} and ch_C to \mathcal{C} .
- Ref sends ch_B to \mathcal{B} and ch_C to \mathcal{C} .

- \mathcal{B} and \mathcal{C} send back answers $\text{ans}_{\mathcal{B}}$ and $\text{ans}_{\mathcal{C}}$, respectively.
- Ref computes bits $b_{\mathcal{B}} \leftarrow \text{Ver}(\text{sk}, m, \text{ch}_{\mathcal{B}}, \text{ans}_{\mathcal{B}})$ and $b_{\mathcal{C}} \leftarrow \text{Ver}(\text{sk}, m, \text{ch}_{\mathcal{C}}, \text{ans}_{\mathcal{C}})$.
- The outcome of the game is denoted by $\mathfrak{ACE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$, which equals 1 if $b_{\mathcal{B}} = b_{\mathcal{C}} = 1$, indicating that the adversary has won, and 0 otherwise, indicating that the adversary has lost.

Note that $\mathfrak{ACE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$ defined the same as $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$, but $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ additionally get oracle access to $P_m(\cdot)$. We say that \mathcal{G} has $(\mathcal{D}, \varepsilon, \widetilde{\text{GenC}})$ augmented unclonable security if for all QPT cloning adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ we have

$$\Pr \left[1 \leftarrow \mathfrak{ACE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq p^{\text{triv}}(\mathcal{G}, \mathcal{D}, \widetilde{\text{GenC}}) + \varepsilon(\lambda).$$

Lemma 27. Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a cloning search game with message space \mathcal{M} such that $p^{\text{triv}}(\mathcal{G}, \mathcal{U}_{\mathcal{M}})$ is negligible. If \mathcal{G} has $(\mathcal{U}_{\mathcal{M}}, |\mathcal{M}|^{-\delta})$ unclonable search security for some $\delta > 0$, then \mathcal{G} has $(\mathcal{U}_{\mathcal{M}}, \text{negl})$ augmented unclonable security.

Proof. We will define a sequence of hybrids:

Hybrid 1: This is the original augmented cloning experiment $\mathfrak{ACE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$. Suppose for the sake of contradiction that there exists a QPT adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which succeeds with non-negligible probability p .

Hybrid 2: In this hybrid, we replace the oracles $P_m(\cdot)$ with $P_S(\cdot)$, where $S \subset \mathcal{M}$ is a random subset containing m of size $|\mathcal{M}|^{1-\delta/2}$. The fact that $p^{\text{triv}}(\mathcal{G}, \mathcal{U}_{\mathcal{M}})$ is negligible implies that $\frac{1}{|\mathcal{M}|}$ is negligible, hence $\frac{|S|}{|\mathcal{M}|} = \frac{1}{|\mathcal{M}|^{\delta/2}}$ is also negligible. We claim that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in this hybrid with probability at least $p - \text{negl}(\lambda)$.

Suppose not, we will construct an adversary \mathcal{A}' which will violate Corollary 2:

- \mathcal{A}' picks $m \leftarrow \mathcal{U}_{\mathcal{M}}$ and receives oracle access to \mathcal{O} , where either $\mathcal{O} = P_m$ or $\mathcal{O} = P_S$ for a random subset $S \subset \mathcal{M}$ of size $|\mathcal{M}|^{1-\delta/2}$ containing m .
- \mathcal{A}' then simulates $\mathfrak{ACE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}$ for $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, using m as the message and using \mathcal{O} as the oracle given to $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. Since $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ are QPT algorithms, \mathcal{A}' only makes polynomially many queries. \mathcal{A}' outputs the bit $\mathfrak{ACE}_{\mathcal{G}, \mathcal{D}}^{\widetilde{\text{GenC}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$.

If $\mathcal{O} = P_m$, then the view of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is exactly Hybrid 1, and otherwise it is exactly Hybrid 2. Thus, \mathcal{A}' has a non-negligible advantage.

Hybrid 3: In this hybrid, we change the order of sampling. A random subset $S \subset \mathcal{M}$ of size $|\mathcal{M}|^{1-\delta/2}$ is sampled at the beginning of the experiment, and the message later is sampled as $m \leftarrow \mathcal{U}_S$. This is perfectly indistinguishable from Hybrid 2, as the view of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is unchanged.

Hybrid 4: In this hybrid, we fix a particular S such that $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ succeeds with probability $p - \text{negl}(\lambda)$ in Hybrid 3, by an averaging argument.

Now we observe that Hybrid 4 is exactly the experiment $\mathfrak{CE}_{\mathcal{G}, \mathcal{U}_S}^{\widetilde{\text{GenC}}}$. Note that the distribution \mathcal{U}_S has min-entropy $(1 - \delta/2) \log_2 |\mathcal{M}|$. Thus, by Lemma 26, we have

$$p - \text{negl}(\lambda) \leq \Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{U}_S}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq |\mathcal{M}|^{\delta/2} \Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{U}_{\mathcal{M}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq |\mathcal{M}|^{-\delta/2},$$

which contradicts the assumption that p is non-negligible. \square

6.1.3 From Augmented Security to Unclonable-Indistinguishability

Now, we state our result that shows how to go from a search game in the plain model to a search game in QROM with unclonable indistinguishable security. We first invoke Lemma 27 to generically obtain augmented unclonable-search security and we then leverage this notion of security to obtain unclonable-indistinguishable security.

Theorem 28 (Search to Indistinguishability). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a statistically correct cloning search game with message space $\mathcal{M} = \{0, 1\}^\ell$, where $\ell = \text{poly}(\lambda)$, such that $p^{\text{triv}}(\mathcal{G}, \mathcal{U}_{\mathcal{M}})$ is negligible and $\text{GenC}(\text{sk}, m)$ does not depend³⁶ on m . Suppose that \mathcal{G} has $(\mathcal{U}_{\mathcal{M}}, |\mathcal{M}|^{-\delta})$ unclonable search security for some $\delta > 0$. Let $n = \text{poly}(\lambda)$, and define a cloning search game $\mathcal{G}' = (\text{Setup}', \text{GenT}', \text{GenC}', \text{Ver}')$ in QROM as follows:*

- \mathcal{G}' has message space $\mathcal{M}' = \{0, 1\}^n$.
- Let $H : \mathcal{M} \rightarrow \mathcal{M}'$ be a random oracle.
- $\text{Setup}'(1^\lambda)$ runs $\text{sk} \leftarrow \text{Setup}(1^\lambda)$. It outputs $\text{sk}' = \text{sk}$.
- $\text{GenT}'(\text{sk}', m')$ parses the input as $\text{sk}' = (\text{sk}, m)$. It samples $m \leftarrow \mathcal{U}_{\mathcal{M}}$ and computes $\rho \leftarrow \text{GenT}(\text{sk}, m)$, then it outputs the token $\rho' = \rho \otimes |m' \oplus H(m)\rangle\langle m' \oplus H(m)|$ and random coins $r_{\text{GenT}'} = m$.
- $\text{GenC}'(\text{sk}', m')$ parses the input as $\text{sk}' = \text{sk}$. It computes $\text{ch} \leftarrow \text{GenC}(\text{sk}, m)$ (recall that by assumption this does not require knowledge of m) and outputs ch .
- \mathcal{G}' is a search game, which defines Ver' .

Then, \mathcal{G}' has statistical correctness and negl unclonable indistinguishable security.

Proof. Statistical correctness of \mathcal{G}' follows easily from statistical correctness of \mathcal{G} , so it suffices to show negl unclonable indistinguishable security. Keep in mind that by Lemma 27, \mathcal{G} has negl augmented unclonable security. Suppose there exists a QPT cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which breaks the unclonable distinguishing security of \mathcal{G}' . Let (m'_0, m'_1) be the messages used by $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

³⁶This requirement can be lifted by extending the definition of stateful cloning games and having $\widetilde{\text{GenC}}$ know the random coins of GenT' (in this case m). We keep the syntax simple for there is no known application to the more general case.

Note that $p^{\text{triv}}(\mathcal{G}', \mathcal{D}_{m'_0, m'_1}) \geq 1/2 - \text{negl}(\lambda)$ by the statistical correctness of \mathcal{G}' and Lemma 24, so that we have

$$\Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}', \mathcal{D}_{m'_0, m'_1}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \geq \frac{1}{2} + \mu(\lambda)$$

for some non-negligible function μ .

We first define a sequence of hybrids:

- **Hybrid 0:** The original cloning experiment $\mathfrak{CE}_{\mathcal{G}', \mathcal{D}_{m'_0, m'_1}}(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ succeeds with probability $1/2 + \mu(\lambda)$ in this experiment.
- **Hybrid 1:** In this hybrid, we will replace the oracle H for \mathcal{A}' only with the reprogrammed oracle H_1 , where

$$H_1(x) := \begin{cases} u, & x = m \\ H(x), & x \neq m \end{cases},$$

where $u \in \{0, 1\}^n$ is an independent random string. We claim that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in this hybrid with probability at least $\frac{1}{2} + \mu(\lambda) - \text{negl}(\lambda)$.

Suppose not, we will construct an adversary \mathcal{A}' that breaks the message hiding property of \mathcal{G} (Lemma 25):

- \mathcal{A}' simulates Hybrid 1 for \mathcal{A} , using the token ρ it receives and a fresh random oracle H' it simulates on-the-fly.
- Then \mathcal{A}' measures a random query made by \mathcal{A} to H' .

Note that if H' was replaced by H , the view of \mathcal{A} would be exactly Hybrid 0. Thus, by Theorem 4, \mathcal{A}' outputs the message m with non-negligible probability, which is larger than $p^{\text{triv}}(\mathcal{G}, \mathcal{U}_{\mathcal{M}}) + |\mathcal{M}|^{-\delta}$, a contradiction.

- **Hybrid 2:** In this hybrid, we change the order of sampling with regard to $H(m)$ in the experiment without changing the view of $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$, so that the probability of success remains the same. More specifically, we sample $r \xleftarrow{\$} \{0, 1\}^n$ and send it to \mathcal{A} instead of $m' \oplus H(m)$. Then, we sample $b \xleftarrow{\$} \{0, 1\}$ and give both \mathcal{B} and \mathcal{C} the reprogrammed oracle H_2^b , where

$$H_2^b(x) := \begin{cases} m'_b \oplus r, & x = m \\ H(x), & x \neq m \end{cases}$$

For fixed $\text{sk}, m, \text{ch}_{\mathcal{B}}, \text{ch}_{\mathcal{C}}$ and fixed H_{-m} , which is the partial random oracle defined on inputs $x \neq m$, consider the following projectors:

- $\Pi_{\mathcal{B}}^b$: Run \mathcal{B}' on challenge $\text{ch}_{\mathcal{B}}$ with oracle H_2^b . Check if the output of \mathcal{B}' is m'_b . Undo the computation.
- Similarly define $\Pi_{\mathcal{C}}^b$ for $b \in \{0, 1\}$.

Without loss of generality assume that the bipartite state output by \mathcal{A} is a pure state $|\phi_{BC}\rangle$ in Hybrid 2. Let $\Pi_B = (\Pi_B^0 + \Pi_B^1)/2$ and $\Pi_C = (\Pi_C^0 + \Pi_C^1)/2$. We spectrally decompose the state as

$$|\phi_{BC}\rangle = \sum_{i,j} \alpha_{i,j} |\varphi_i\rangle_B \otimes |\psi_j\rangle_C.$$

where $\Pi_B |\varphi_i\rangle = \lambda_i |\varphi_i\rangle$ and $\Pi_C |\psi_j\rangle = \gamma_j |\psi_j\rangle$.

Claim 3. For any polynomial $p(\cdot)$, with overwhelming probability over $\text{sk}, m, \text{ch}_B, \text{ch}_C, H_{-m}$, we have

$$\sum_{\substack{i: |\lambda_i - 1/2| \geq 1/p(\lambda) \\ j: |\gamma_j - 1/2| \geq 1/p(\lambda)}} |\alpha_{i,j}|^2 \leq \text{negl}(\lambda) \quad (14)$$

Proof. Assume the quantity in eq. (14) is a non-negligible function $w(\lambda)$. We will construct an adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ which breaks the negl augmented security of \mathcal{G} :

- $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ get oracle access to $P_m(\cdot)$, where $m \leftarrow \mathcal{D}$.
- \mathcal{A}' gets a quantum token ρ . It samples a $2t$ -wise independent hash function $H : \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ to simulate a random oracle and a random string $r \xleftarrow{\$} \{0, 1\}^n$, where $t = \text{poly}(\lambda)$ is an upper-bound on the number of total random oracle queries made by $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. Then, it runs \mathcal{A} on $\rho \otimes |r\rangle\langle r|$ to get a bipartite state $|\phi_{BC}\rangle$. It sends this state to \mathcal{B}' and \mathcal{C}' . It also sends the description of H to both of them.
- \mathcal{B}' receives a challenge ch_B from the challenger. It implements the operator Π_B defined above. Note that \mathcal{B}' can reprogram the random oracle H_2 on input m using and its oracle access to $P_m(\cdot)$. Then, \mathcal{B}' applies the efficient symmetric approximate threshold measurement $\text{SATI}_{(P,Q),\gamma}^{\epsilon,\delta}$ in Theorem 9 with $P = (\Pi_0^B + \Pi_1^B)/2$, $Q = I - P$, $\gamma = 3/4p$, $\epsilon = 1/4p$ and $\delta = 2^{-\lambda}$. If the outcome is 0, \mathcal{B}' aborts. If the outcome is 1, \mathcal{B}' then runs \mathcal{B} on the leftover state with H_2^0 or H_2^1 picked uniformly at random. It measures and outputs a random query \mathcal{B} makes to the random oracle.
- \mathcal{C}' is defined the same way as \mathcal{B}' , in the end outputting a random query made by \mathcal{C} to the random oracle.

By Theorem 9 bullet (1), both \mathcal{B}' and \mathcal{C}' will get outcome 1 with non-negligible probability $w - 2\delta$, in which case by bullet (2) the leftover state will be 4δ -close to the the following state:

$$\sum_{\substack{i: |\lambda_i - 1/2| > 1/4p \\ j: |\gamma_j - 1/2| > 1/4p}} \alpha_{i,j} |\varphi_i\rangle_B \otimes |\psi_j\rangle_C.$$

Observe that when \mathcal{B} does not query m , it will succeed with probability exactly $1/2$. Therefore, by Theorem 4, the query weight of \mathcal{B} on m is non-negligible. Similarly, the query weight of \mathcal{C} on m is non-negligible. Therefore, $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ succeed with non-negligible probability, a contradiction. \square

Therefore, for any polynomial p , we have that $|\phi_{BC}\rangle$ is negligibly close to the state $|\phi_B\rangle + |\phi_C\rangle$, where

$$|\phi_B\rangle = \sum_{i:|\lambda_i-1/2|\leq 1/p} \alpha_{i,j} |\varphi_i\rangle_B \otimes |\psi_j\rangle_C, \quad |\phi_C\rangle = \sum_{\substack{i:|\lambda_i-1/2|>1/p \\ j:|\gamma_j-1/2|\leq 1/p}} \alpha_{i,j} |\varphi_i\rangle_B \otimes |\psi_j\rangle_C,$$

in which case we could bound the success probability of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ in Hybrid 2 as:

$$\begin{aligned} & \frac{1}{2} + \mu(\lambda) - \text{negl}(\lambda) \leq (|(\Pi_B^0 \otimes \Pi_C^0)(|\phi_B\rangle + |\phi_C\rangle)|^2 + |(\Pi_B^1 \otimes \Pi_C^1)(|\phi_B\rangle + |\phi_C\rangle)|^2)/2 \\ &= \frac{1}{2} \cdot (\langle \phi_B | (\Pi_B^0 \otimes \Pi_C^0) | \phi_B \rangle + \langle \phi_B | (\Pi_B^1 \otimes \Pi_C^1) | \phi_B \rangle + \langle \phi_C | (\Pi_B^0 \otimes \Pi_C^0) | \phi_C \rangle + \langle \phi_C | (\Pi_B^1 \otimes \Pi_C^1) | \phi_C \rangle) \\ &+ \text{Re} (\langle \phi_B | (\Pi_B^0 \otimes \Pi_C^0) | \phi_C \rangle + \langle \phi_B | (\Pi_B^1 \otimes \Pi_C^1) | \phi_C \rangle) \\ &\leq \frac{1}{2} \cdot (\langle \phi_B | (\Pi_B^0 \otimes I) | \phi_B \rangle + \langle \phi_B | (\Pi_B^1 \otimes I) | \phi_B \rangle + \langle \phi_C | (I \otimes \Pi_C^0) | \phi_C \rangle + \langle \phi_C | (I \otimes \Pi_C^1) | \phi_C \rangle) \\ &+ \text{Re} (\langle \phi_B | (\Pi_B^0 \otimes \Pi_C^0) | \phi_C \rangle + \langle \phi_B | (\Pi_B^1 \otimes \Pi_C^1) | \phi_C \rangle) \\ &= \langle \phi_B | (\Pi_B \otimes I) | \phi_B \rangle + \langle \phi_C | (I \otimes \Pi_C) | \phi_C \rangle + \text{Re} (\langle \phi_B | (\Pi_B^0 \otimes \Pi_C^0) | \phi_C \rangle + \langle \phi_B | (\Pi_B^1 \otimes \Pi_C^1) | \phi_C \rangle) \\ &\leq \frac{1}{2} + \frac{1}{p}, \end{aligned}$$

where in the last step we used Lemma 6. Since p is arbitrary, this is a contradiction. \square

As a corollary, we achieve unclonable encryption in QROM using BB84 states. This is an improvement over the main result of [AKL⁺22], as it can be more easily implemented on near-term quantum computers.

Corollary 8 (Existence of Unclonable Encryption in QROM Using Prepare-and-Measure States). *There exists a public-key unclonable encryption scheme with unclonable indistinguishable security in QROM, which uses only prepare-and-measure quantum operations.*

Proof. Set $\mathcal{G} = \mathcal{G}_{\text{BB84}}$ from Definition 21, then it satisfies the condition of Theorem 28, so there exists \mathcal{G}' as described in the theorem. Observe that since \mathcal{G} is a cloning encryption game, so is \mathcal{G}' . Thus, \mathcal{G}' gives a construction of unclonable encryption in QROM, and the negl unclonable distinguishing security implies unclonable security of this encryption scheme.

Finally, we invoke the generic compiler in [AK21] to achieve a public-key scheme³⁷. \square

6.2 Indistinguishability-to-Search

Theorem 29. *If $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ is a cloning search game with statistical correctness and ε unclonable indistinguishable security, then \mathcal{G} has $(\mathcal{D}, 2\varepsilon + \text{negl})$ unclonable security for any distribution \mathcal{D} over the message space \mathcal{M} such that \mathcal{G} is \mathcal{D} -evasive.*

Proof. We will give a generic proof that works for an arbitrary extension $\widetilde{\text{GenC}}$, which we will omit for simplicity. Suppose there exists a cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which breaks $(\mathcal{D}, \varepsilon)$ unclonable

³⁷Although [AK21] only showed that their transformation preserves weak unclonable security, it could easily be checked that it also preserves strong unclonable security.

security of \mathcal{G} . We will construct an adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$, which will break $(\mathcal{D}_{m_0, m}, \varepsilon')$ unclonable security of \mathcal{G} , where $\varepsilon' = \varepsilon/2 - \text{negl}$, $m_0 \in \mathcal{M}$ is a fixed message, and m is a message sampled as $m \leftarrow \mathcal{D}$. By convexity, this would imply that there exists a fixed value of m for which the security is broken, hence finishing the proof. We describe $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ below:

- \mathcal{A}' is the same as \mathcal{A} .
- \mathcal{B}' runs \mathcal{B} with the input it receives. If the output is m , \mathcal{B}' outputs m ; otherwise \mathcal{B}' outputs m_0 .
- \mathcal{C}' is defined similarly to \mathcal{B}' .

Note that if the message in the experiment $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}_{m_0, m}}$ above is m , then by assumption $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ wins with probability $p > \varepsilon$. On the other hand, if the message is m_0 , then the probability that \mathcal{B}' outputs m is at most $\text{OPT}(m \mid \text{ch}_{\mathcal{B}}) \leq \text{negl}(\lambda)$ by Lemma 24. Hence, by union bound, $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ succeeds with overwhelming probability in this case, and we have

$$\mathbb{E}_{m \leftarrow \mathcal{D}} \Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}_{m_0, m}}(1^\lambda, (\mathcal{A}', \mathcal{B}', \mathcal{C}')) \right] = \frac{1}{2} (p + 1 - \text{negl}) \geq \frac{1}{2} + \frac{\varepsilon(\lambda)}{2} - \text{negl}(\lambda)$$

□

as desired.

7 From Search to Decision Games

We present a transformation from search games to decision games. We remark that the transformation is tailored to application of copy-protection and thus, the resulting decision game has a specific form.

Theorem 30 (Search to Decision). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be a statistically correct cloning encryption game with message space $\mathcal{M} = \{0, 1\}^\ell$, where $\ell = \text{poly}(\lambda)$, such that*

- $\text{Setup}(1^\lambda)$ outputs a uniformly random key $\text{sk} \xleftarrow{\$} \mathcal{K}$, where $|\mathcal{K}|^{-1}$ is negligible in λ .
- \mathcal{G} is $\mathcal{U}_{\mathcal{M}}$ -evasive.
- \mathcal{G} has $(\mathcal{U}_{\mathcal{M}}, |\mathcal{M}|^{-\delta})$ unclonable search security for some $\delta > 0$

Define a cloning decision game $\mathcal{G}' = (\text{Setup}', \text{GenT}', \text{GenC}', \text{Ver}')$ in QROM as follows:

- Let \mathcal{M}' be an arbitrary message space.
- Let $H : \mathcal{M} \rightarrow \{0, 1\}^n$ and $G : \mathcal{M}' \rightarrow \mathcal{K}$ be random oracles, where $n = \Omega(\lambda)$.
- $\text{Setup}'(1^\lambda)$ outputs $\text{sk}' = \perp$
- $\text{GenT}'(\text{sk}', m')$ samples $m \leftarrow \mathcal{U}_{\mathcal{M}}$. It computes $\rho \leftarrow \text{GenT}(G(m'), m)$. It outputs $\rho' = \rho \otimes |H(m)\rangle\langle H(m)|$.
- $\text{GenC}'(\text{sk}', m')$ is an algorithm which does not depend on sk' , and outputs a value $\text{ch}' \in \mathcal{M}'$.

- $\text{Ver}'(\text{sk}', m', \text{ch}', \text{ans}')$ accepts if $\text{ans}' = [\text{ch}' == m']$.

Then, \mathcal{G}' has statistical correctness and $(\mathcal{D}', \text{negl})$ independent-challenge unclonable security for any unlearnable distribution \mathcal{D}' over \mathcal{M} .

Proof. We begin with correctness. Following Definition 3, we define a QPT algorithm $\mathcal{A}_{\mathcal{G}'}$ as follows:

- Parse the token as $\rho' = \rho \otimes |y\rangle\langle y|$, i.e. measure the last register to obtain $y \in \{0, 1\}^n$.
- Compute $\text{ch} \leftarrow \text{GenC}(G(\text{ch}'), m)$, recall that this does not require the knowledge of m .
- Compute $m \leftarrow \mathcal{A}_{\mathcal{G}}(\rho, \text{ch})$.
- Output $b = [H(m) == y]$.

To analyze correctness, we consider two cases: (1) $\text{ch}' = m'$ and (2) $\text{ch}' \neq m'$.

- If $\text{ch}' = m'$, then by statistical correctness of \mathcal{G} , $\mathcal{A}_{\mathcal{G}'}$ receives the correct message $m = \text{sk}'$ from $\mathcal{A}_{\mathcal{G}}$ above and outputs 1 with overwhelming probability.
- If $\text{ch}' \neq m'$, then $\mathcal{A}_{\mathcal{G}}$ sees a challenge generated by a random key $\tilde{\text{sk}} = G(\text{ch}')$ that is independent from $G(m')$, which is the key used to generate the token $\mathcal{A}_{\mathcal{G}}$ receives. We will show that the probability that $\mathcal{A}_{\mathcal{G}}$ outputs the correct message m in this case is negligible. This will then imply that $\mathcal{A}_{\mathcal{G}'}$ outputs 0 with overwhelming probability, since the output length of H is $\Omega(\lambda)$.

Now suppose that $\mathcal{A}_{\mathcal{G}}$ outputs the correct message above with non-negligible probability p , we will construct an adversary \mathcal{A}' that breaks the message hiding property of \mathcal{G} using $\mathcal{A}_{\mathcal{G}}$:

- \mathcal{A}' receives a token $\rho \leftarrow \text{GenT}(\text{sk}, m)$, where $\text{sk} \xleftarrow{\$} \mathcal{K}$ and $m \leftarrow \mathcal{D}$.
- \mathcal{A}' then samples $\tilde{\text{sk}} \xleftarrow{\$} \mathcal{K}$ and computes $\text{ch} \leftarrow \text{GenC}(\tilde{\text{sk}}, m)$. Recall that this does not require the knowledge of m .
- Next, \mathcal{A}' computes and outputs $\tilde{m} \leftarrow \mathcal{A}_{\mathcal{G}}(\rho, \text{ch})$.

By assumption, $\tilde{m} = m$ with non-negligible probability, since for $\text{ch}' \neq m'$ the value $G(\text{ch}')$ is identically distributed as the value $\tilde{\text{sk}}$ sampled by \mathcal{A}' above. Therefore, \mathcal{A}' breaks the message-hiding property of \mathcal{G} given in Lemma 25, a contradiction.

Next, we show $(\mathcal{D}, \text{negl})$ unclonable security. Note that by Lemma 27, \mathcal{G} has negl augmented unclonable security. We will define a sequence of hybrids:

- **Hybrid 0:** This is the original cloning experiment $\mathfrak{CE}_{\mathcal{G}', \mathcal{D}'}^{\text{ind}}$. Suppose for the sake of contradiction that there exists a cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which succeeds in this hybrid with probability $p = p^{\text{triv}}(\mathcal{G}', \mathcal{D}') + \mu(\lambda)$ for a non-negligible function μ .

- **Hybrid 1:** In this hybrid, we will change the oracle G for \mathcal{A} only to the punctured oracle G_m , defined as

$$G_m(x) := \begin{cases} u, & x = m' \\ G(x), & x \neq m' \end{cases},$$

where $u \in \mathcal{K}$ is an independent uniformly random value. We claim that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeed in this experiment with probability $p - \text{negl}(\lambda)$. Suppose this is false. Let ρ_{BC}^j be the bipartite state outputs by \mathcal{A} in Hybrid j , then $\text{TD}(\rho_{BC}^1, \rho_{BC}^2)$ must be non-negligible, since the only difference between Hybrids 0-1 is on the random oracle G for \mathcal{A} . Using \mathcal{A} , we will construct an adversary that breaks the unlearnability of \mathcal{D}' :

- \mathcal{A}' gets oracle access to $P_{m'}(\cdot)$ (which will not be used), where $m' \leftarrow \mathcal{D}'$. It samples random oracles $G' : \mathcal{M}' \rightarrow \mathcal{K}$ and $H' : \mathcal{M} \rightarrow \{0, 1\}^n$, as well as a message $m \leftarrow \mathcal{D}$.
- Then, \mathcal{A}' samples $\text{sk} \xleftarrow{\$} \mathcal{K}$, computes $\rho \leftarrow \text{GenT}(\text{sk}, m)$, and runs $\mathcal{A}(\rho, H'(m))$.
- During the last step, it measures and outputs a random query made by \mathcal{A} to the oracle G' .

As the view of \mathcal{A} in Hybrid 1 is perfectly simulated, by Theorem 4, the query weight of \mathcal{A} on m' is non-negligible, hence \mathcal{A}' correctly outputs m' with non-negligible probability, breaking unlearnability.

- **Hybrid 2:** In this hybrid, we change the order of sampling. A random value $k \leftarrow \mathcal{U}_{\mathcal{K}}$ is sampled at the beginning of the experiment and the quantum part of the token received by \mathcal{A} is calculated as $\rho \leftarrow \text{GenT}(k, m)$. In turn, the random oracle G for \mathcal{B} and \mathcal{C} is replaced by the reprogrammed oracle $G_{m'}^2$ which is defined as

$$G_{m'}^2(x) := \begin{cases} k, & x = m' \\ G(x), & x \neq m' \end{cases},$$

This hybrid is perfectly indistinguishable from Hybrid 1, hence $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in this hybrid with probability at least $p - \text{negl}(\lambda)$.

- **Hybrid 3:** In this hybrid, we replace the oracle H for \mathcal{A} with the punctured oracle H_m defined as

$$H_m(x) := \begin{cases} w, & x = m \\ G(x), & x \neq m \end{cases},$$

where $w \in \{0, 1\}^n$ is an independent uniformly random value. We claim that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in this hybrid with probability at least $p - \text{negl}(\lambda)$. Suppose not, then we have $\text{TD}(\rho_{BC}^2, \rho_{BC}^3)$ is non-negligible. We will use $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ to construct an adversary \mathcal{A}' to break the message hiding property of \mathcal{G} :

- \mathcal{A}' receives $\rho \leftarrow \text{GenT}(\text{sk}, m)$ from the challenger, where $\text{sk} \xleftarrow{\$} \mathcal{K}$ and $m \leftarrow \mathcal{D}$.

- Then \mathcal{A}' samples $w \xleftarrow{\$} \{0,1\}^n$ and simulates random oracles $H' : \mathcal{M} \rightarrow \{0,1\}^n$, $G' : \mathcal{M}' \rightarrow \mathcal{K}$ and runs \mathcal{A} on input $\rho \otimes |w\rangle\langle w|$ with oracles H', G' . It measures and outputs a random query made by \mathcal{A} to the oracle H' .

As the view of \mathcal{A} in Hybrid 3 is perfectly simulated, by Theorem 4, the query weight of \mathcal{A} on m is non-negligible, hence \mathcal{A}' outputs m with non-negligible probability, contradicting the message-hiding property of \mathcal{G} given in Lemma 25.

Therefore, we have established that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in Hybrid 3 with probability $p^{\text{triv}}(\mathcal{G}', \mathcal{D}') + \eta(\lambda)$ for some non-negligible function η . Without loss of generality, assume that the bipartite state created by \mathcal{A} is a pure state, i.e. $\rho_{BC}^3 = |\phi_{BC}\rangle\langle\phi_{BC}|$. Consider the following binary POVM elements:

- Π_B : Sample $\text{ch}' \leftarrow \text{GenC}'(sk', m')$. Run \mathcal{B} with oracles $H, G_{m'}^2$ and input ch' . Measure if the output of \mathcal{B} equals $[\text{ch}' == m']$, in which case we will say that \mathcal{B} passed Π_B .
- Similarly define Π_C for \mathcal{C} .

We write the state received by \mathcal{B} and \mathcal{C} in spectral decomposition as

$$|\phi_{BC}\rangle\langle\phi_{BC}| = \sum_{i,j} \alpha_{i,j} |\varphi_i\rangle_B |\psi_j\rangle_C,$$

where $\Pi_B |\varphi_i\rangle_B = \lambda_i |\varphi_i\rangle_B$ and $\Pi_C |\psi_j\rangle_C = \gamma_j |\psi_j\rangle_C$. Let q be a polynomial. Let $p^{\text{triv}} := p^{\text{triv}}(\mathcal{G}', \mathcal{D}')$. We can bound the probability that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in Hybrid 3 as

$$\begin{aligned} p - \text{negl}(\lambda) &\leq \langle\phi_{BC}|(\Pi_B \otimes \Pi_C)|\phi_{BC}\rangle = \sum_{i,j} |\alpha_{i,j}|^2 \lambda_i \gamma_j \\ &\leq \sum_{\substack{i: \lambda_i > p^{\text{triv}} + 1/q \\ j: \gamma_j > p^{\text{triv}} + 1/q}} |\alpha_{i,j}|^2 \lambda_i \gamma_j \\ &\quad + \left(p^{\text{triv}} + \frac{1}{q}\right) \sum_{\substack{i: \lambda_i \leq p^{\text{triv}} + 1/q \\ j: \gamma_j > p^{\text{triv}} + 1/q}} |\alpha_{i,j}|^2 + \left(p^{\text{triv}} + \frac{1}{q}\right) \sum_{i,j: \mu_j \leq p^{\text{triv}} + 1/q} |\alpha_{i,j}|^2 \\ &\leq p^{\text{triv}} + \frac{1}{q} + \sum_{\substack{i: \lambda_i > p^{\text{triv}} + 1/q \\ j: \gamma_j > p^{\text{triv}} + 1/q}} |\alpha_{i,j}|^2 \lambda_i \gamma_j. \end{aligned}$$

Thus, it suffices to show the following claim to reach a contradiction:

Claim 4. *For any polynomial q , with overwhelming probability over $\text{sk}' = m, m', H, G, u, k, w$, we have*

$$\sum_{\substack{i: \lambda_i > p^{\text{triv}} + 1/q \\ j: \gamma_j > p^{\text{triv}} + 1/q}} |\alpha_{i,j}|^2 \lambda_i \gamma_j \leq \text{negl}(\lambda).$$

Proof. Suppose that there exists $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which violates Claim 4. Consider the following cloning adversary $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1)$ against the $(\mathcal{D}, \text{negl})$ augmented security of \mathcal{G} :

- $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1)$ get a quantum token $\rho \leftarrow \text{GenT}(k, m)$ from the challenger, where $k \leftarrow \mathcal{U}_K$ and $m \leftarrow \mathcal{D}$. Also, $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1)$ get oracle access to $P_m(\cdot)$.
- \mathcal{A}_1 samples³⁸ random oracles $G' : \mathcal{M}' \rightarrow \mathcal{K}$ and $H' : \mathcal{M} \rightarrow \{0, 1\}^n$, as well as random strings $m' \leftarrow \mathcal{D}'$, $w \xleftarrow{\$} \{0, 1\}^n$. It runs \mathcal{A} on input $\rho \otimes |w\rangle\langle w|$ with oracles G', H' to obtain a bipartite state ρ_{BC} , which it sends to \mathcal{B}_1 and \mathcal{C}_1 . In addition, \mathcal{A}_1 sends (G', H', m', w) to both \mathcal{B}_1 and \mathcal{C}_1 .
- In the challenge phase, \mathcal{B}_1 receives $\text{ch}_B = k \leftarrow \text{GenC}(k, m)$ from the challenger and $(G', H', m', w, \rho_{BC}[B])$ from \mathcal{A}_1 . Observe that \mathcal{B}_1 can implement Π_B using these values. In particular, it can reprogram H' so that it outputs w on input m using its oracle access to $P_m(\cdot)$, and it can reprogram G' to output k . With this in mind, \mathcal{B}_1 applies the efficient approximated threshold measurement $\text{ATI}_{(P,Q),\gamma_1}^{\epsilon,\delta}$ in Theorem 8 with $P = \Pi_B, Q = I - \Pi_B, \gamma_1 = p^{\text{triv}} + 3/4q, \epsilon = 1/4q$, and $\delta = 2^{-\lambda}$, with outcome b_B . If $b_B = 0$, \mathcal{B}_1 aborts. If $b_B = 1$, then \mathcal{B}_1 runs Π_B and measures and outputs a random query made by \mathcal{B} to the reprogrammed oracle H' . \mathcal{C}_1 is defined similarly to \mathcal{B}_1 .

By Theorem 8 bullets (1) and (2), with non-negligible probability the bipartite state obtained by $(\mathcal{B}_1, \mathcal{C}_1)$ right before measuring the queries is negligibly close to a state of the form

$$\sum_{\substack{i: \lambda_i > p^{\text{triv}} + 1/4q \\ j: \gamma_j > p^{\text{triv}} + 1/4q}} \beta_{i,j} |\varphi_i\rangle_B \otimes |\psi_j\rangle_C. \quad (15)$$

Before we analyze the success probability of $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1)$, we define two algorithms $\mathcal{B}_2, \mathcal{C}_2$, where \mathcal{B}_2 is defined the same as \mathcal{B}_1 except it does not reprogram the oracle H' when implementing Π_B , and similarly for \mathcal{C}_2 . We will refer to this different implementation of Π_B (hence a different operator) as Π'_B . We emphasize that the only difference between (\mathcal{B}_1, Π_B) and (\mathcal{B}_2, Π'_B) (similarly between (\mathcal{C}_1, Π_C) and (\mathcal{C}_2, Π'_C)) is the output of the oracle H' given to \mathcal{B} (\mathcal{C}) on input m .

In contrast to eq. (15), if we consider $(\mathcal{A}_1, \mathcal{B}_2, \mathcal{C}_1)$ instead, the state shared by $(\mathcal{B}, \mathcal{C})$ at the same stage will have negligible weight on eigenstates of Π'_B with eigenvalue $\lambda' > p^{\text{triv}} + 1/8q$, for otherwise one could violate the message hiding property (Lemma 25) using \mathcal{A} and \mathcal{B} by simulating the view of \mathcal{B} without the knowledge of m , which is information theoretically hidden from \mathcal{B} . Similarly, the state will have negligible weight on eigenstates of Π'_C with eigenvalue $\gamma' > p^{\text{triv}} + 1/8q$ if we consider $(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_2)$ instead. Therefore, By Theorem 4, conditioned on eq. (15), i.e. conditioned on $b_B = b_C = 1$, \mathcal{B}_1 and \mathcal{C}_1 will both output m with non-negligible probability, hence breaking $(\mathcal{D}, \text{negl})$ augmented security of \mathcal{G} .

□

□

Corollary 9. *There exists a copy-protection scheme for an arbitrary class of point functions secure in QROM.*

³⁸Recall that this can be efficiently done via $2t$ -wise independent hash functions where $t = \text{poly}(\lambda)$ is a query-bound for $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

Proof. Follows easily from Theorem 30 after setting $\mathcal{G} = \mathcal{G}_{\text{BB84}}$ and \mathcal{M}' to be the class of point functions represented by the special input m' , that is, m' is interpreted as the description of the point function $P_{m'}(\cdot)$. \square

8 Asymmetric Cloning Games

Recall that we defined cloning games (e.g. see Definition 9) such that \mathcal{B} and \mathcal{C} are required to pass the same verification phase, i.e. with respect to the same algorithms $(\text{GenC}, \text{Ver})$. This does not capture all unclonable primitives, in particular cryptography with certified deletion. In this section, we extend further the cloning game syntax to include such primitives and give a simpler proof of an existing feasibility result as an application of our framework in the asymmetric setting.

8.1 Definitions

We start with the formal definition of asymmetric cloning games.

Definition 35 (Asymmetric Cloning Game). *An asymmetric cloning game consists of a tuple of efficient algorithms $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$:*

- **Key Generation:** $\text{Setup}(1^\lambda)$ is a PPT algorithm which takes as input a security parameter 1^λ in unary. It outputs a secret key $\text{sk} \in \{0, 1\}^*$. We will assume without loss of generality that sk always contains the security parameter 1^λ in order to simplify the notation below.
- **Token Generation:** $\text{GenT}(\text{sk}, m)$ is a QPT algorithm takes as input a secret key sk and a message $m \in \{0, 1\}^*$. It outputs a quantum token ρ .
- **Challenge Generation:** $\text{GenC}_{\mathcal{B}}(\text{sk}, m)$ takes as input a secret key sk and a message m . It outputs a classical challenge $\text{ch} \in \{0, 1\}^*$. $\text{GenC}_{\mathcal{C}}$ has the same syntax.
- **Verification:** $\text{Ver}_{\mathcal{B}}(\text{sk}, m, \text{ch}, \text{ans})$ takes as input a secret key sk , a message m , a challenge ch , and an answer ans . It outputs either 0 (reject) or 1 (accept). $\text{Ver}_{\mathcal{C}}$ has the same syntax.

Correctness. For correctness, we require that the verifications for \mathcal{B} and \mathcal{C} are both individually doable given the entire quantum token.

Definition 36 (Correctness of Asymmetric Cloning Games). *Let $\delta_{\mathcal{B}}, \delta_{\mathcal{C}} : \mathbb{Z}^+ \rightarrow [0, 1]$ and $\delta = (\delta_{\mathcal{B}}, \delta_{\mathcal{C}})$. We say that \mathcal{G} has δ -correctness if there exist efficient quantum algorithms $\mathcal{B}_{\mathcal{G}}, \mathcal{C}_{\mathcal{G}}$ such that for all messages $m \in \mathcal{M}$:*

$$\Pr \left[\text{Ver}_{\mathcal{B}}(\text{sk}, m, \text{ch}, \text{ans})=1 : \begin{array}{l} \text{sk} \leftarrow \text{Setup}(1^\lambda) \\ \rho \leftarrow \text{GenT}(\text{sk}, m) \\ \text{ch} \leftarrow \text{GenC}_{\mathcal{B}}(\text{sk}, m) \\ \text{ans} \leftarrow \mathcal{B}_{\mathcal{G}}(\rho, \text{ch}) \end{array} \right] \geq \delta_{\mathcal{B}}(\lambda), \quad \Pr \left[\text{Ver}_{\mathcal{C}}(\text{sk}, m, \text{ch}, \text{ans})=1 : \begin{array}{l} \text{sk} \leftarrow \text{Setup}(1^\lambda) \\ \rho \leftarrow \text{GenT}(\text{sk}, m) \\ \text{ch} \leftarrow \text{GenC}_{\mathcal{C}}(\text{sk}, m) \\ \text{ans} \leftarrow \mathcal{C}_{\mathcal{G}}(\rho, \text{ch}) \end{array} \right] \geq \delta_{\mathcal{C}}(\lambda).$$

If $\delta = (1, 1)$ (or $\delta(\lambda) = (1 - \text{negl}(\lambda), 1 - \text{negl}(\lambda))$), we say \mathcal{G} has perfect (or statistical) correctness.

Security. We define security formally so that \mathcal{B} and \mathcal{C} are asked to pass their corresponding verifications. We highlight the part different from Definition 6 in blue.

Definition 37 (Asymmetric Cloning Experiment). An asymmetric cloning experiment, denoted by $\mathfrak{CE}_{\mathcal{G},\mathcal{D}}$, is a security game played between a referee Ref and a cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. It is parameterized by an asymmetric cloning game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$ and a distribution \mathcal{D} over the message space \mathcal{M} . The experiment is described as follows:

- **Setup Phase:**

- All parties get a security parameter 1^λ as input.
- Ref samples a message $m \leftarrow \mathcal{D}$.
- Ref computes $\text{sk} \leftarrow \text{Setup}(1^\lambda)$ and $\rho \leftarrow \text{GenT}(\text{sk}, m)$.
- Ref sends ρ to \mathcal{A} .

- **Splitting Phase:**

- \mathcal{A} computes a bipartite state ρ' over registers B, C .
- \mathcal{A} sends $\rho'[B]$ to \mathcal{B} and $\rho'[C]$ to \mathcal{C} .

- **Challenge Phase:**

- Ref samples $\text{ch}_{\mathcal{B}} \leftarrow \text{GenC}_{\mathcal{B}}(\text{sk}, m)$ and $\text{ch}_{\mathcal{C}} \leftarrow \text{GenC}_{\mathcal{C}}(\text{sk}, m)$.
- Ref sends $\text{ch}_{\mathcal{B}}$ to \mathcal{B} and $\text{ch}_{\mathcal{C}}$ to \mathcal{C} .
- \mathcal{B} and \mathcal{C} send back answers $\text{ans}_{\mathcal{B}}$ and $\text{ans}_{\mathcal{C}}$, respectively.
- Ref computes bits $b_{\mathcal{B}} \leftarrow \text{Ver}_{\mathcal{B}}(\text{sk}, m, \text{ch}_{\mathcal{B}}, \text{ans}_{\mathcal{B}})$ and $b_{\mathcal{C}} \leftarrow \text{Ver}_{\mathcal{C}}(\text{sk}, m, \text{ch}_{\mathcal{C}}, \text{ans}_{\mathcal{C}})$.
- The outcome of the game is denoted by $\mathfrak{CE}_{\mathcal{G},\mathcal{D}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$, which equals 1 if $b_{\mathcal{B}} = b_{\mathcal{C}} = 1$, indicating that the adversary has won, and 0 otherwise, indicating that the adversary has lost.

Definition 38 (Asymmetric Trivial Cloning Attack). We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is a \mathcal{B} -trivial cloning attack against an asymmetric cloning experiment $\mathfrak{CE}_{\mathcal{G},\mathcal{D}}$ if \mathcal{A} upon receiving a token ρ , sends the product state $|\perp\rangle\langle\perp| \otimes \rho$ to \mathcal{B} and \mathcal{C} . In other words, only \mathcal{C} gets the token ρ . We denote by $\text{TRIV}_{\mathcal{B}}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}})$ the set of \mathcal{B} -trivial attacks against $\mathfrak{CE}_{\mathcal{G},\mathcal{D}}$. We similarly define $\text{TRIV}_{\mathcal{C}}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}})$ as the set of \mathcal{C} -trivial attacks.

We denote by $\text{TRIV}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}})$ the set of trivial attacks against $\mathfrak{CE}_{\mathcal{G},\mathcal{D}}$.

Finally, we define

$$\text{TRIV}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}}) := \text{TRIV}_{\mathcal{B}}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}}) \cup \text{TRIV}_{\mathcal{C}}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}})$$

As pointed out in Remark 2, the definition above captures mixtures of \mathcal{B} -trivial and \mathcal{C} -trivial attacks by convexity.

Definition 39 (Asymmetric Trivial Success Probability for Cloning Games). We define the \mathcal{B} -trivial success probability of an asymmetric cloning experiment $\mathfrak{CE}_{\mathcal{G},\mathcal{D}}$ as

$$p_{\mathcal{B}}^{\text{triv}}(\mathcal{G}, \mathcal{D}) := \sup_{(\mathcal{A}, \mathcal{B}, \mathcal{C}) \in \text{TRIV}_{\mathcal{B}}(\mathfrak{CE}_{\mathcal{G},\mathcal{D}})} \Pr [1 \leftarrow \mathfrak{CE}_{\mathcal{G},\mathcal{D}}((\mathcal{A}, \mathcal{B}, \mathcal{C}))].$$

We similarly define $p_C^{\text{triv}}(\mathcal{G}, \mathcal{D})$ as the \mathcal{C} -trivial success probability of $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}$. Accordingly, we define the trivial success probability of $\mathfrak{CE}_{\mathcal{G}, \mathcal{D}}$ as

$$\begin{aligned} p^{\text{triv}}(\mathcal{G}, \mathcal{D}) &:= \sup_{(\mathcal{A}, \mathcal{B}, \mathcal{C}) \in \text{TRIV}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}})} \Pr [1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}}((\mathcal{A}, \mathcal{B}, \mathcal{C}))] \\ &= \max(p_B^{\text{triv}}(\mathcal{G}, \mathcal{D}), p_C^{\text{triv}}(\mathcal{G}, \mathcal{D})), \end{aligned}$$

where the last equality follows from the fact that any $(\mathcal{A}, \mathcal{B}, \mathcal{C}) \in \text{TRIV}(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}})$ is a convex combination of attacks from $\text{TRIV}_B(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}})$ and $\text{TRIV}_C(\mathfrak{CE}_{\mathcal{G}, \mathcal{D}})$.

Definition 40 (Asymmetric Cloning Search Game). *Let $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}, \text{Ver})$ be an asymmetric cloning game such that $\text{Ver}_C(\text{sk}, m, \text{ch}, \text{ans})$ accepts if and only if $\text{ans} = m$. Then, \mathcal{G} is called an asymmetric cloning search game.*

Remark 9. *In the definition above, the "search" restriction only applies to \mathcal{C} . As such, the definition complements the asymmetry between \mathcal{B} and \mathcal{C} , while being consistent with Definition 4.*

With the modified definitions of trivial success, cloning experiment and search game, the notions of unclonable security for asymmetric games are defined similarly to Definitions 9 to 11.

Definition 41 ((Asymmetric) Unclonable Security). *Let \mathcal{G} be an asymmetric cloning game, \mathcal{D} be a distribution over the message space \mathcal{M} , and $\varepsilon : \mathbb{Z}^+ \rightarrow [0, 1]$. We say that \mathcal{G} has $(\mathcal{D}, \varepsilon)$ unclonable security if for all QPT cloning adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ we have:*

$$\Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}, \mathcal{D}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq p^{\text{triv}}(\mathcal{G}, \mathcal{D}) + \varepsilon(\lambda).$$

If $|\mathcal{M}| = 1$, we will simply write ε unclonable security.

Definition 42 ((Asymmetric) Unclonable Search Security). *If \mathcal{G} is an asymmetric cloning search game with $(\mathcal{D}, \varepsilon)$ unclonable security, we additionally say that \mathcal{G} has $(\mathcal{D}, \varepsilon)$ unclonable search security.*

Definition 43 ((Asymmetric) Unclonable Indistinguishable Security). *Let \mathcal{D}_{m_0, m_1} denote the distribution that outputs messages m_0 and m_1 with probability $1/2$ each. We say that an asymmetric cloning search game \mathcal{G} has ε unclonable indistinguishable security if it has $(\mathcal{D}_{m_0, m_1}, \varepsilon)$ unclonable search security for any pair of messages $m_0, m_1 \in \mathcal{M}$.*

Remark 10. *Note that Definition 43 could have significantly different flavor compared to its symmetric counterpart (Definition 11), given that the distinguishing between m_0 and m_1 need not be simultaneous.*

Remark 11 (Extended Asymmetric Cloning Games). *One could consider the extended definition for asymmetric cloning games with correlated challenge distributions as in Section 4.4, but it is unnecessary for the certified deletion setting which we will focus on next.*

Applications. Asymmetric cloning games provide a framework to analyze unclonable primitives in which the goal of the adversary is to perform two different tasks (as \mathcal{B} and \mathcal{C}), each of which require a quantum token, using only one copy of the token. Examples include primitives with certified deletion property, analyzed below in Section 8.2, where \mathcal{C} is asked to perform the intended use of the primitive and \mathcal{B} is asked to generate a classical certificate of deletion for the quantum token. Yet another example is secure software leasing [AL20], where \mathcal{B} is asked to return a quantum (software) token and \mathcal{C} is asked to achieve the functionality of the software.

Tokenized Signatures. Another class of unclonable primitives is *one-time primitives*, in which an honest user consumes the quantum token after using it once. While it is possible to model a one-time primitive as a (regular) cloning game, it is more natural to cast it as an asymmetric cloning game. We will demonstrate this using the example of *tokenized signatures* for single-bit messages. Informally, in a tokenized signature scheme, a quantum signing token ρ can be used to sign one (and only one) bit $b \in \{0, 1\}$. Formally, a tokenized signature scheme [BS16, CLLZ21, Shm22] is a tuple of efficient algorithms $(\text{Gen}, \text{GenT}', \text{Sign}, \text{Ver}')$:

- $\text{Gen}(1^\lambda)$ takes as input a security parameter and outputs a pair of classical keys (vk, sk) .
- $\text{GenT}'(sk)$ takes as input a secret key and outputs a quantum signature token ρ .
- $\text{Sign}(\rho, x)$ takes as input a token ρ and a classical message $x \in \{0, 1\}$. It outputs a classical signature σ_x .
- $\text{Ver}'(vk, x, \sigma)$ takes as input a verification key, a classical message, and a classical signature. It outputs 0 (reject) or 1 (accept).

$(\text{Gen}, \text{GenT}', \text{Sign}, \text{Ver}')$ defines an asymmetric cloning game $\mathcal{G}_{\text{TS}} = (\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$, as follows:

- Similar to quantum money, there is no message, i.e. $m = \perp$.
- $\text{Setup}(1^\lambda)$ runs $(vk, sk) \leftarrow \text{Gen}(1^\lambda)$ and outputs $\text{sk} = (vk, sk)$.
- $\text{GenT}(\text{sk}, m)$ parses the input as $\text{sk} = (vk, sk)$, computes $\rho \leftarrow \text{GenT}'(sk)$ and outputs $\rho \otimes |vk\rangle\langle vk|$.
- There is no challenge for either \mathcal{B} or \mathcal{C} , i.e. $\text{GenC}_{\mathcal{B}}$ and $\text{GenC}_{\mathcal{C}}$ always output \perp .
- $\text{Ver}_{\mathcal{B}}(\text{sk}, \perp, \perp, \text{ans})$ parses the input as $\text{sk} = (vk, sk)$. It computes $b \leftarrow \text{Ver}'(vk, 0, \text{ans})$ and accepts if $b = 1$.
- $\text{Ver}_{\mathcal{C}}(\text{sk}, \perp, \perp, \text{ans})$ parses the input as $\text{sk} = (vk, sk)$. It computes $b \leftarrow \text{Ver}'(vk, 1, \text{ans})$ and accepts if $b = 1$.

Informally, \mathcal{B} is signing the bit 0 and \mathcal{C} is signing the bit 1 using the signing token.

Correctness. We say that the tokenized signature scheme $(\text{Gen}, \text{GenT}', \text{Sign}, \text{Ver}')$ satisfies correctness if \mathcal{G}_{TS} has perfect correctness.

Security. We say that the scheme has unclonable security if \mathcal{G}_{TS} has $(\mathcal{D}_{\perp}, \varepsilon)$ unclonable security, where $\varepsilon = \text{negl}$ yields optimal security.

8.2 Deletion Games

Certified deletion can be defined as a special case of asymmetric cloning games, where \mathcal{B} gets no challenge.

Definition 44 (Deletion Game). *A deletion game is an asymmetric cloning game $\mathcal{G} = (\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$ such that $\text{GenC}_{\mathcal{B}}$ always outputs \perp .*

Note that in the definition above, \mathcal{B} represents \mathcal{A} generating a classical certificate for deleting ρ . Since \mathcal{B} gets no challenge, \mathcal{A} and \mathcal{B} can be effectively considered as one party.

Unclonable Encryption with Certified Deletion. A notable primitive in this category is unclonable encryption with certified deletion, which we can define as a deletion game $\mathcal{G}_{\text{UE-CD}} = (\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$ with the following properties:

- $\mathcal{G}_{\text{UE-CD}}$ is an asymmetric cloning search game.
- $\mathcal{G}_{\text{UE-CD}}$ has statistical correctness.
- $\text{GenC}_{\mathcal{C}}(\text{sk}, m)$ outputs sk with probability 1.

For this primitive, like regular unclonable encryption, we consider two types of security: (1) $(\mathcal{U}_{\mathcal{M}}, \varepsilon)$ unclonable security and ε unclonable indistinguishable security, the latter of which is stronger.

8.3 Construction of Unclonable Encryption with Certified Deletion

8.3.1 Preliminaries

We cite two lemmas from literature that we will need in our construction. The first lemma is commonly used to bound the value of monogamy-of-entanglement games.

Lemma 31 (Lemma 2 in [TFKW13]). *Let A_1, \dots, A_N be positive-semidefinite operators over a Hilbert space \mathcal{H} and let $\{\pi_k\}_{k \in [N]}$ be N mutually orthogonal permutations over $[N]$. Then,*

$$\left\| \sum_{i \in [N]} A_i \right\|_{\text{op}} \leq \sum_{k \in [N]} \max_{i \in [N]} \left\| \sqrt{A_i} \sqrt{A_{\pi^k(i)}} \right\|_{\text{op}},$$

where $\|\cdot\|_{\text{op}}$ denotes the operator norm, also known as the Schatten- ∞ norm.

The second lemma we will need is the local version of quantum Goldreich-Levin (Lemma 23, which was known in previous work [AC02, CLLZ21]).

Lemma 32 (Quantum Goldreich-Levin). *Suppose a quantum algorithm \mathcal{A} , given a quantum state ρ , a key k , and a random string $r \in \{0, 1\}^n$ can output $\langle r, x \rangle \pmod{2}$ with probability $1/2 + \varepsilon$. Then, there exists a quantum algorithm (extractor) \mathcal{A}' , which, given the same quantum state ρ and the key k , can output $x \in \{0, 1\}^n$ with probability $4\varepsilon^2$.*

8.3.2 Achieving Unclonable Search Security

We first describe a known construction³⁹ based on BB84 states, denoted as $\mathcal{G}_{\text{BB84-CD}} = (\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}})$:

- The message space is $\mathcal{M} = \{0, 1\}^\lambda$
- $\text{Setup}(1^\lambda)$ outputs $\text{sk} = \theta \xleftarrow{\$} \{0, 1\}^\lambda$
- $\text{GenT}(\theta, m)$ takes as input $\theta, m \in \{0, 1\}^\lambda$ and outputs $\rho = |m^\theta\rangle\langle m^\theta|$, where $|m^\theta\rangle = H^\theta |m\rangle$

³⁹This is a simplified version of the construction of [BI20].

- $\text{GenC}_{\mathcal{B}}(\theta, m)$ outputs \perp as required by a deletion game.
- $\text{Ver}_{\mathcal{B}}(\theta, m, \perp, \text{ans}_{\mathcal{B}})$ accepts if and only if $\text{ans}_{\mathcal{B}} \in \{0, 1\}^\lambda$ satisfies $\text{ans}_{\mathcal{B},i} = m_i$ for all $i \in \theta[1]$, where $\theta[b] := \{i \in [n] : \theta_i = b\}$ for $b \in \{0, 1\}$.
- $\text{GenC}_{\mathcal{C}}(\theta, m)$ outputs θ as required by unclonable encryption with certified deletion.
- $\text{Ver}_{\mathcal{C}}(\theta, m, \text{ch}_{\mathcal{C}}, \text{ans}_{\mathcal{C}})$ checks if $\text{ans}_{\mathcal{C}} = m$ as required by an unclonable search game.

Correctness. It is easy to see that $\mathcal{G}_{\text{BB84-CD}}$ satisfies perfect correctness. In order to generate a certificate, $\mathcal{B}_{\mathcal{G}_{\text{BB84-CD}}}(\rho, \perp)$ measures ρ in the Fourier basis and outputs the result, whereas in order to decrypt, $\mathcal{C}_{\mathcal{G}_{\text{BB84-CD}}}(\rho, \theta)$ computes $H^\theta \rho H^\theta$, then measures in the computational basis and outputs the result.

Security. While it is possible to show that $\mathcal{G}_{\text{BB84-CD}}$ above satisfies negl unclonable indistinguishable security, the proofs involve either entropic arguments [BI20] or other advanced techniques [BK22]. We will instead follow a different approach which we believe is simpler in many aspects. First, we will show that $\mathcal{G}_{\text{BB84-CD}}$ satisfies $(\mathcal{U}_{\mathcal{M}}, \text{negl})$ unclonable security by reduction to a monogamy-of-entanglement game following the techniques of [TFKW13], similar to the first construction of unclonable encryption [BL20]. Then, we will modify the scheme and apply Quantum Goldreich-Levin to achieve negl unclonable indistinguishable security. We give the formal details below.

Theorem 33. $\mathcal{G}_{\text{BB84-CD}}$ defined above has $(\mathcal{U}_{\mathcal{M}}, \text{negl})$ information theoretic unclonable search security.

Proof. We first define a monogamy-of-entanglement game $\text{MOE} = \text{MOE}(\lambda)$ for certified deletion which is closely related to $\mathcal{G}_{\text{BB84-CD}}$. Let $\theta[b]$ be defined as above.

MOE is a game between a referee Ref and an adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:

- $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and Ref get a security parameter 1^λ as input.
- $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ prepare a bipartite state ρ_{XA} and send $\rho[X]$ (the X register) to ch, where $X = \{0, 1\}^\lambda$.
- \mathcal{A} computes a bipartite state ρ' over registers B, C , then sends $\rho'[B]$ to \mathcal{B} and $\rho'[C]$ to \mathcal{C} .
- Ref samples $\theta \xleftarrow{\$} \{0, 1\}^\lambda$ and measures the X register in the basis $\{|x^\theta\rangle\}_{x \in \{0,1\}^\lambda}$, obtaining outcome x .
- \mathcal{B} outputs $x_B \in \{0, 1\}^\lambda$
- \mathcal{C} gets θ as input and outputs $x_C \in \{0, 1\}^\lambda$.
- The outcome of the game is denoted by $\text{MOE}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$, which equals 1 if $x_C = x$ and $x_{B,i} = x_i$ for all $i \in \theta[1]$, indicating that the adversary has won, and 0 otherwise, indicating that the adversary has lost.

We will begin with showing that the success probability of any adversary in this game is exponentially small. The proof leverages the widely used techniques of [TFKW13].

Claim 5. For any (unbounded) adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, we have

$$\Pr \left[1 \leftarrow \text{MOE}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] \leq \left(\frac{1}{2} + \frac{1}{2\sqrt[4]{2}} \right)^\lambda$$

Proof. We can write the winning probability of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ as follows:

$$\begin{aligned} p_{\text{win}} &:= \Pr \left[1 \leftarrow \text{MOE}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] = \mathbb{E}_\theta \sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta[1]}} \text{Tr} \left[\left(|x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right) \rho \right] \\ &\leq \frac{1}{2^\lambda} \left\| \sum_{\substack{\theta, x, x': \\ x'_i = x_i, \forall i \in \theta[1]}} |x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right\|_{\text{op}}, \end{aligned}$$

where $\{B_{x'}\}_{x' \in \{0,1\}^\lambda}$, as well as $\{C_x^\theta\}_{x \in \{0,1\}^\lambda}$ for any $\theta \in \{0,1\}^\lambda$, is a POVM. By a standard purification argument, we can w.l.o.g. assume that the POVM's are projective measurements. Next, we can apply Lemma 31 to get

$$p_{\text{win}} \leq \frac{1}{2^\lambda} \sum_{k \in \{0,1\}^\lambda} \max_{\theta \in \{0,1\}^\lambda} \left\| \Pi^\theta \Pi^{\pi_k(\theta)} \right\|_{\text{op}}, \quad (16)$$

where

$$\Pi^\theta := \sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta[1]}} |x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta,$$

and $\pi_k : \{0,1\}^\lambda \rightarrow \{0,1\}^\lambda$ are 2^λ mutually orthogonal permutations to be determined later. For convenience, we define the index sets $S[\theta, \theta'] := \{i \in [\lambda] : \theta_i = \theta'_i\}$ ('same' indices) and $D[\theta, \theta'] := \{i \in [\lambda] : \theta_i \neq \theta'_i\}$ ('different' indices). Also let $\theta' = \pi_j(\theta)$ and $s = |S[\theta, \theta']|$ for short-hand notation. Now, we have

$$\Pi^{\theta'} \leq \bar{Q} := \sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta'[1]}} |x^{\theta'}\rangle\langle x^{\theta'}| \otimes B_{x'} \otimes I,$$

hence

$$\begin{aligned}
& \left\| \Pi^\theta \Pi^{\theta'} \right\|_{\text{op}}^2 \leq \left\| \Pi^\theta \bar{Q} \right\|_{\text{op}}^2 = \left\| \Pi^\theta \bar{Q} \Pi^\theta \right\|_{\text{op}} \\
& = \left\| \sum_{\substack{x,y,z,x',y',z': \\ x'_i=x_i, z'_i=z_i, \forall i \in \theta[1] \\ y'_i=y_i, \forall i \in \theta'[1]}} |x^\theta \rangle \langle x^\theta| \cdot |y^{\theta'} \rangle \langle y^{\theta'}| \cdot |z^\theta \rangle \langle z^\theta| \otimes B_{x'} B_{y'} B_{z'} \otimes C_x^\theta C_z^\theta \right\|_{\text{op}} \\
& = \left\| \sum_{\substack{x,y,x': \\ x'_i=x_i, \forall i \in \theta[1] \\ x'_i=y_i, \forall i \in \theta'[1]}} |x^\theta \rangle \langle x^\theta| \cdot |y^{\theta'} \rangle \langle y^{\theta'}| \cdot |x^\theta \rangle \langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right\|_{\text{op}} \\
& = \left\| \sum_{\substack{x,y,x': \\ x'_i=x_i, \forall i \in \theta[1] \\ x'_i=y_i, \forall i \in \theta'[1]}} \left| \langle x^\theta | y^{\theta'} \rangle \right|^2 |x^\theta \rangle \langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right\|_{\text{op}} \tag{17}
\end{aligned}$$

$$= 2^{s-\lambda} \left\| \sum_{\substack{x,y,x': \\ x'_i=x_i, \forall i \in \theta[1] \\ x'_i=y_i, \forall i \in \theta'[1] \\ x_i=y_i, \forall i \in S[\theta, \theta']}} |x^\theta \rangle \langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right\|_{\text{op}} \tag{18}$$

Above in eq. (17) we used the fact that the inner product $|\langle x^\theta | y^{\theta'} \rangle|$ vanishes unless $x_i = y_i$ for all $i \in S[\theta, \theta']$, in which case it is contributed a factor of $2^{-1/2}$ for every index $i \notin S[\theta, \theta']$. Now in eq. (18), every term in the sum with distinct (x, x') is orthogonal, so we only need to count the number of y values for given (x, x') . Specifically, we need the number of $y \in \{0, 1\}^\lambda$ such that $y_i = x'_i$ for $i \in \theta'[1]$ and $y_i = x_i$ for $i \in S[\theta, \theta']$. Note that these two conditions never contradict because $x'_i = x_i$ for all $i \in \theta[1]$, and $S[\theta, \theta'] \cap \theta'[1] \subseteq \theta[1]$. Hence, y has $|S[\theta, \theta'] \cup \theta'[1]|$ coordinates fixed. The number of free coordinates, then is given by $\lambda - |S[\theta, \theta'] \cup \theta'[1]| = |\theta[1] \setminus \theta'[1]|$, which is the number of indices where $\theta'_i = 0, \theta_i = 1$. Thus, eq. (18) equals 2^{-t} , where t is the number of indices i such that $\theta_i = 0, \theta'_i = 1$, then we can bound eq. (16) as

$$p_{\text{win}} \leq \frac{1}{2^\lambda} \sum_{k \in \{0,1\}^\lambda} \max_{\theta \in \{0,1\}^\lambda} \left\| \Pi^\theta \Pi^{\pi_k(\theta)} \right\|_{\text{op}} \leq \frac{1}{2^\lambda} \sum_{k \in \{0,1\}^\lambda} \max_{\theta \in \{0,1\}^\lambda} 2^{-t/2}$$

Furthermore, observe that by symmetry, we can achieve the same bound for t representing the number of indices i with $\theta_i = 1, \theta'_i = 0$ as well for any θ, θ' . Thus, without loss of generality we

could assume $t \geq |D[\theta, \theta']|/2$ by choosing the optimal order in the analysis. Finally, we choose $\pi_k(\theta) = \theta \oplus k$ to be the cyclic permutations, which yields $\binom{\lambda}{k}$ permutations with $D[\theta, \theta'] = k$. Thus,

$$p_{\text{win}} \leq \frac{1}{2^\lambda} \sum_{k=0}^{\lambda} \binom{\lambda}{k} 2^{-k/4} = \left(\frac{1}{2} + \frac{1}{2\sqrt[4]{2}} \right)^\lambda \approx 0.920^\lambda$$

as desired. □

Remark 12. [CV21] bounds an easier version of MOE by $0.85^{\lambda/2} \approx 0.924^\lambda$, where \mathcal{B} and \mathcal{C} need to guess indices in $\theta[0]$ and $\theta[1]$ after learning θ , respectively. Because θ is unknown, \mathcal{A} cannot trivially succeed by splitting the qubits between them.

We finish the proof of the theorem by giving a reduction from unclonable security of $\mathcal{G}_{\text{BB84-CD}}$ to hardness of MOE. Since 0.920^λ is negligible in λ , this suffices.

Claim 6. Suppose there exists an adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ breaking $(\mathcal{U}_{\mathcal{M}}, \varepsilon)$ unclonable search security of $\mathcal{G}_{\text{BB84-CD}}$, then there exists $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ such that

$$\Pr \left[1 \leftarrow \text{MOE}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right] > \varepsilon.$$

Proof. Define $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ as follows:

- $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ creates λ EPR pairs, i.e. the bipartite state $|\psi\rangle = 2^{-\lambda/2} \sum_{x \in \{0,1\}^\lambda} |x\rangle |x\rangle$ over registers XA , and send the X register to the referee Ref.
- \mathcal{A}' applies the same splitting channel as \mathcal{A} . Similarly, \mathcal{B}' and \mathcal{C}' apply the same measurements as \mathcal{B} and \mathcal{C} .

We will show that $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ defined above succeeds in MOE with probability equal to the success probability of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. Recall that for any $\theta \in \{0,1\}^\lambda$, we have $|\psi\rangle = 2^{-\lambda/2} \sum_{x \in \{0,1\}^\lambda} |x^\theta\rangle |x^\theta\rangle$.

Hence,

$$\begin{aligned}
\Pr \left[1 \leftarrow \text{MOE}(1^\lambda, (\mathcal{A}', \mathcal{B}', \mathcal{C}')) \right] &= \mathbb{E}_\theta \text{Tr} \left[\sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta'[1]}} \left(|x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right) (I_X \otimes \mathcal{A}) (|\psi\rangle\langle\psi|) \right] \\
&= 2^{-\lambda} \mathbb{E}_\theta \text{Tr} \left[\sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta'[1]}} \sum_{y, z} \left(|x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right) (I_X \otimes \mathcal{A}) \left(|y^\theta\rangle\langle z^\theta| \otimes |y^\theta\rangle\langle z^\theta| \right) \right] \\
&= 2^{-\lambda} \mathbb{E}_\theta \text{Tr} \left[\sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta'[1]}} \sum_{y, z} \left(|x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right) \left(|y^\theta\rangle\langle z^\theta| \otimes \mathcal{A} \left(|y^\theta\rangle\langle z^\theta| \right) \right) \right] \\
&= 2^{-\lambda} \mathbb{E}_\theta \text{Tr} \left[\sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta'[1]}} \left(|x^\theta\rangle\langle x^\theta| \otimes B_{x'} \otimes C_x^\theta \right) \left(|x^\theta\rangle\langle x^\theta| \otimes \mathcal{A} \left(|x^\theta\rangle\langle x^\theta| \right) \right) \right] \\
&= \mathbb{E}_{\theta, x} \text{Tr} \left[\sum_{\substack{x, x': \\ x'_i = x_i, \forall i \in \theta'[1]}} \left(B_{x'} \otimes C_x^\theta \right) \left(\mathcal{A} \left(|x^\theta\rangle\langle x^\theta| \right) \right) \right] \\
&= \Pr \left[1 \leftarrow \mathfrak{CE}_{\mathcal{G}_{\text{BB84-CD}}, \mathcal{U}_{\mathcal{M}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) \right],
\end{aligned}$$

which completes the proof. □

□

□

8.3.3 Achieving Unclonable-Indistinguishable Security

Now, we give the construction of unclonable-indistinguishable secure unclonable encryption with certified deletion, denoted as $\mathcal{G}_{\text{BB84-CD}}^{\text{UI}} = ((\text{Setup}, \text{GenT}, \text{GenC}_{\mathcal{B}}, \text{Ver}_{\mathcal{B}}, \text{GenC}_{\mathcal{C}}, \text{Ver}_{\mathcal{C}}))$ below:

- We consider single-bit messages, i.e. $\mathcal{M} = \{0, 1\}$
- $\text{Setup}(1^\lambda)$ samples $\theta \xleftarrow{\$} \{0, 1\}^\lambda$ and $r \xleftarrow{\$} \{0, 1\}^\lambda \setminus \{0^\lambda\}$ independently. It outputs $\text{sk} = (\theta, r)$.
- $\text{GenT}(\text{sk}, m)$ parses the input as $\text{sk} = (\theta, r)$. It samples $x \xleftarrow{\$} \{x' \in \{0, 1\}^\lambda : \langle r, x' \rangle = m\}$, where we treat $\{0, 1\}^\lambda$ as \mathbb{F}_2^λ for the inner product $\langle \cdot, \cdot \rangle$. It outputs $\rho = |x^\theta\rangle\langle x^\theta|$.

Theorem 34. $\mathcal{G}_{\text{BB84-CD}}^{\text{UI}}$ above is an unclonable encryption scheme for single-bit messages with certified deletion, which has negl information theoretic unclonable indistinguishable security.

Proof. Correctness is easy to see. For security, we define a sequence of hybrids:

- **Hybrid 0:** The asymmetric cloning experiment $\mathfrak{CE}_{\mathcal{G}_{\text{BB84-CD}}, \mathcal{U}_{\{0,1\}}^{\text{UI}}}$.

Suppose for the sake of contradiction that there exists an adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ which succeeds in this experiment with probability $1/2 + \varepsilon$, where ε is non-negligible in λ .

- **Hybrid 2:** In this hybrid, we sample $r \xleftarrow{\$} \{0, 1\}^\lambda$, i.e. we allow $r = 0^\lambda$.

Since the probability of this is negligibly small, the success probability of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ in this Hybrid is $\varepsilon - \text{negl}(\lambda)$.

- **Hybrid 2:** In this hybrid, instead of sampling m uniformly and sampling x conditioned on $\langle r, x \rangle = m$, we sample $x \xleftarrow{\$} \{0, 1\}^\lambda$ uniformly and set $m = \langle r, x \rangle$. In other words, we remove m from the experiment and ask \mathcal{C} to output $\langle r, x \rangle$ in order to pass verification.

This hybrid is statistically indistinguishable from Hybrid 0, since $|\Pr[\langle r, x \rangle = 0] - 1/2| \leq \text{negl}(\lambda)$ for $r, x \xleftarrow{\$} \{0, 1\}^\lambda$. Thus, $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ succeeds in this Hybrid with probability $\varepsilon - \text{negl}(\lambda)$. Let $p_{\mathcal{B}}$ be the probability that \mathcal{B} passes verification, and let $p_{\mathcal{C}}$ be the probability that \mathcal{C} passes verification conditioned on \mathcal{B} passing verification. Then, $p_{\mathcal{B}}p_{\mathcal{C}} \geq \varepsilon - \text{negl}(\lambda)$, so that $p_{\mathcal{B}}$ and $p_{\mathcal{C}}$ are both non-negligible in λ .

- **Hybrid 3:** This is the asymmetric cloning experiment $\mathfrak{CE}_{\mathcal{G}_{\text{BB84-CD}}, \mathcal{U}_{\{0,1\}^\lambda}}$.

Observe that the only difference between Hybrids 2 and 3 is the verification phase for \mathcal{C} . Accordingly, we define an adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C}')$ for this Hybrid, where \mathcal{C}' is the extractor guaranteed by Lemma 32, applied with respect to the mixed state received by \mathcal{C} in Hybrid 1 conditioned on \mathcal{B} passing verification. The guarantee of Lemma 32 states that if $p'_{\mathcal{C}}$ is the probability of \mathcal{C}' passing verification conditioned on \mathcal{B} passing verification, then $p'_{\mathcal{C}} \geq 4p_{\mathcal{C}}^2$, which is non-negligible. Therefore, $(\mathcal{A}, \mathcal{B}, \mathcal{C}')$ succeeds in Hybrid 3 with non-negligible probability $p_{\mathcal{B}}p'_{\mathcal{C}}$, contradicting Theorem 33. □

Remark 13. *For simplicity, we give a construction for single-bit messages, but a standard hybrid argument can be used to show that bitwise encryption works for multi-bit messages in the certified deletion setting.*

References

- [Aar09] Scott Aaronson. “Quantum copy-protection and quantum money”. In: *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE, 2009, pp. 229–242 (cit. on pp. 3, 4).
- [AC02] Mark Adcock and Richard Cleve. “A Quantum Goldreich-Levin Theorem with Cryptographic Applications”. In: *STACS 2002*. Ed. by Helmut Alt and Afonso Ferreira. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 323–334. ISBN: 978-3-540-45841-8 (cit. on pp. 20, 73, 84).

- [AC12] Scott Aaronson and Paul Christiano. *Quantum Money from Hidden Subspaces*. 2012. DOI: [10.48550/ARXIV.1203.4740](https://doi.org/10.48550/ARXIV.1203.4740). URL: <https://arxiv.org/abs/1203.4740> (cit. on pp. 3, 6, 35, 36).
- [AK21] Prabhanjan Ananth and Fatih Kaleoglu. “Unclonable Encryption, Revisited”. In: *Theory of Cryptography Conference*. Springer. 2021, pp. 299–329 (cit. on pp. 4, 5, 8, 34, 63).
- [AKL⁺22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. *On the Feasibility of Unclonable Encryption, and More*. Cryptology ePrint Archive, Paper 2022/884. <https://eprint.iacr.org/2022/884>. 2022. URL: <https://eprint.iacr.org/2022/884> (cit. on pp. 4, 6–8, 18–20, 24, 34, 63).
- [AL20] Prabhanjan Ananth and Rolando L. La Placa. *Secure Software Leasing*. 2020. DOI: [10.48550/ARXIV.2005.05289](https://doi.org/10.48550/ARXIV.2005.05289). URL: <https://arxiv.org/abs/2005.05289> (cit. on pp. 3, 71).
- [ALL⁺20] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. *New Approaches for Quantum Copy-Protection*. 2020. DOI: [10.48550/ARXIV.2004.09674](https://doi.org/10.48550/ARXIV.2004.09674). URL: <https://arxiv.org/abs/2004.09674> (cit. on pp. 3, 8).
- [AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. “Cryptography from pseudorandom quantum states”. In: *Annual International Cryptology Conference*. Springer. 2022, pp. 208–236 (cit. on p. 5).
- [BB20] Charles H Bennett and Gilles Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In: *arXiv preprint arXiv:2003.06557* (2020) (cit. on p. 6).
- [BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. “Strengths and weaknesses of quantum computing”. In: *SIAM journal on Computing* 26.5 (1997), pp. 1510–1523 (cit. on p. 23).
- [BBK22] Nir Bitansky, Zvika Brakerski, and Yael Tauman Kalai. *Constructive Post-Quantum Reductions*. 2022. DOI: [10.48550/ARXIV.2203.02314](https://doi.org/10.48550/ARXIV.2203.02314). URL: <https://arxiv.org/abs/2203.02314> (cit. on pp. 14, 21, 25, 36–40, 43, 49, 50, 52).
- [BCQ22] Zvika Brakerski, Ran Canetti, and Luowen Qian. “On the computational hardness needed for quantum cryptography”. In: *arXiv preprint arXiv:2209.04101* (2022) (cit. on p. 5).
- [BDF⁺11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. “Random oracles in a quantum world”. In: *International conference on the theory and application of cryptology and information security*. Springer. 2011, pp. 41–69 (cit. on p. 18).
- [BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. “Quantum one-time programs”. In: *Annual Cryptology Conference*. Springer. 2013, pp. 344–360 (cit. on p. 3).
- [BI20] Anne Broadbent and Rabib Islam. “Quantum Encryption with Certified Deletion”. In: *Theory of Cryptography*. Springer International Publishing, 2020, pp. 92–122. DOI: [10.1007/978-3-030-64381-2_4](https://doi.org/10.1007/978-3-030-64381-2_4). URL: https://doi.org/10.1007/978-3-030-64381-2_4 (cit. on pp. 3, 6, 9, 20, 73, 74).

- [BJL⁺21] Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram. “Secure software leasing without assumptions”. In: *Theory of Cryptography Conference*. Springer. 2021, pp. 90–120 (cit. on pp. 3, 6, 33).
- [BK22] James Bartusek and Dakshita Khurana. *Cryptography with Certified Deletion*. 2022. DOI: [10.48550/ARXIV.2207.01754](https://doi.org/10.48550/ARXIV.2207.01754). URL: <https://arxiv.org/abs/2207.01754> (cit. on pp. 9, 20, 74).
- [BL20] Anne Broadbent and Sébastien Lord. “Uncloneable Quantum Encryption via Oracles”. en. In: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. DOI: [10.4230/LIPICS.TQC.2020.4](https://doi.org/10.4230/LIPICS.TQC.2020.4). URL: <https://drops.dagstuhl.de/opus/volltexte/2020/12063/> (cit. on pp. 3, 6, 8, 18, 20, 34, 58, 74).
- [BS16] Shalev Ben-David and Or Sattath. *Quantum Tokens for Digital Signatures*. 2016. DOI: [10.48550/ARXIV.1609.09047](https://doi.org/10.48550/ARXIV.1609.09047). URL: <https://arxiv.org/abs/1609.09047> (cit. on pp. 3, 72).
- [CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. “Hidden Cosets and Applications to Unclonable Cryptography”. In: *Advances in Cryptology – CRYPTO 2021*. Ed. by Tal Malkin and Chris Peikert. Cham: Springer International Publishing, 2021, pp. 556–584. ISBN: 978-3-030-84242-0 (cit. on pp. 3, 4, 6–8, 20, 72, 73, 84).
- [CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. *Quantum copy-protection of compute-and-compare programs in the quantum random oracle model*. 2020. DOI: [10.48550/ARXIV.2009.13865](https://doi.org/10.48550/ARXIV.2009.13865). URL: <https://arxiv.org/abs/2009.13865> (cit. on pp. 4–6, 8, 33).
- [CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. *Post-Quantum Succinct Arguments: Breaking the Quantum Rewinding Barrier*. Cryptology ePrint Archive, Paper 2021/334. <https://eprint.iacr.org/2021/334>. 2021. URL: <https://eprint.iacr.org/2021/334> (cit. on pp. 15, 16, 23, 25, 43, 47).
- [CV21] Eric Culf and Thomas Vidick. “A monogamy-of-entanglement game for subspace coset states”. In: (July 2021). arXiv: [2107.13324](https://arxiv.org/abs/2107.13324) [cs.CR] (cit. on pp. 20, 77).
- [Die82] DGBJ Dieks. “Communication by EPR devices”. In: *Physics Letters A* 92.6 (1982), pp. 271–272 (cit. on p. 3).
- [GL89] O. Goldreich and L. A. Levin. “A Hard-Core Predicate for All One-Way Functions”. In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. STOC ’89. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 25–32. ISBN: 0897913078. DOI: [10.1145/73007.73010](https://doi.org/10.1145/73007.73010) (cit. on pp. 6, 13, 54).
- [Got02] Daniel Gottesman. “Uncloneable Encryption”. In: (2002). DOI: [10.48550/ARXIV.QUANT-PH/0210062](https://doi.org/10.48550/ARXIV.QUANT-PH/0210062). URL: <https://arxiv.org/abs/quant-ph/0210062> (cit. on p. 3).
- [GZ20] Marios Georgiou and Mark Zhandry. “Unclonable Decryption Keys”. In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 877 (cit. on pp. 4, 7, 34).
- [HMNY22] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. “Certified everlasting zero-knowledge proof for QMA”. In: *Annual International Cryptology Conference*. Springer. 2022, pp. 239–268 (cit. on p. 3).

- [Imp95] Russell Impagliazzo. “A personal view of average-case complexity”. In: *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*. IEEE. 1995, pp. 134–147 (cit. on p. 5).
- [IR90] Russell Impagliazzo and Steven Rudich. “Limits on the provable consequences of one-way permutations”. In: *Advances in cryptology—CRYPTO ’88 (Santa Barbara, CA, 1988)*. Vol. 403. Lecture Notes in Comput. Sci. Springer, Berlin, 1990, pp. 8–26. DOI: [10.1007/0-387-34799-2_2](https://doi.org/10.1007/0-387-34799-2_2). URL: https://doi.org/10.1007/0-387-34799-2_2 (cit. on p. 5).
- [Jor75] Camille Jordan. “Essai sur la géométrie à n dimensions”. fre. In: *Bulletin de la Société Mathématique de France* 3 (1875), pp. 103–174. URL: <http://eudml.org/doc/85325> (cit. on pp. 16, 24).
- [KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. “Secure software leasing from standard assumptions”. In: *Theory of Cryptography Conference*. Springer. 2021, pp. 31–61 (cit. on p. 3).
- [KT22] Srijita Kundu and Ernest Y. -Z. Tan. *Device-independent uncloneable encryption*. 2022. DOI: [10.48550/ARXIV.2210.01058](https://arxiv.org/abs/2210.01058). URL: <https://arxiv.org/abs/2210.01058> (cit. on p. 84).
- [MW05] Chris Marriott and John Watrous. “Quantum Arthur-Merlin Games”. In: (2005). DOI: [10.48550/ARXIV.CS/0506068](https://arxiv.org/abs/cs/0506068). URL: <https://arxiv.org/abs/cs/0506068> (cit. on p. 24).
- [MY22] Tomoyuki Morimae and Takashi Yamakawa. “Quantum commitments and signatures without one-way functions”. In: *Annual International Cryptology Conference*. Springer. 2022, pp. 269–295 (cit. on p. 5).
- [Shm22] Omri Shmueli. “Public-key Quantum money with a classical bank”. In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 2022, pp. 790–803 (cit. on pp. 3, 72).
- [SW22] Or Sattath and Shai Wyborski. “Uncloneable Decryption from Quantum Copy Protection”. In: *arXiv preprint arXiv:2203.05866* (2022) (cit. on p. 5).
- [TFKW13] Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. “A monogamy-of-entanglement game with applications to device-independent quantum cryptography”. In: *New Journal of Physics* 15.10 (Oct. 2013), p. 103002. DOI: [10.1088/1367-2630/15/10/103002](https://doi.org/10.1088/1367-2630/15/10/103002). URL: <https://doi.org/10.1088/1367-2630/15/10/103002> (cit. on pp. 8, 9, 20, 73, 74).
- [VZ20] Thomas Vidick and Tina Zhang. *Classical proofs of quantum knowledge*. 2020. DOI: [10.48550/ARXIV.2005.01691](https://arxiv.org/abs/2005.01691). URL: <https://arxiv.org/abs/2005.01691> (cit. on p. 6).
- [Wie83] Stephen Wiesner. “Conjugate coding”. In: *ACM Sigact News* 15.1 (1983), pp. 78–88 (cit. on pp. 3, 6, 35).
- [WZ82] William K Wootters and Wojciech H Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (1982), pp. 802–803 (cit. on p. 3).

- [Zha12] Mark Zhandry. “Secure Identity-Based Encryption in the Quantum Random Oracle Model”. In: *Proceedings of the 32nd Annual Cryptology Conference on Advances in Cryptology — CRYPTO 2012 - Volume 7417*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 758–775. ISBN: 9783642320088. DOI: [10.1007/978-3-642-32009-5_44](https://doi.org/10.1007/978-3-642-32009-5_44). URL: https://doi.org/10.1007/978-3-642-32009-5_44 (cit. on p. 23).
- [Zha17] Mark Zhandry. “Quantum Lightning Never Strikes the Same State Twice”. In: *CoRR* abs/1711.02276 (2017). arXiv: [1711.02276](https://arxiv.org/abs/1711.02276). URL: <http://arxiv.org/abs/1711.02276> (cit. on pp. 3, 6, 35).
- [Zha19] Mark Zhandry. “How to Record Quantum Queries, and Applications to Quantum Indifferentiability”. In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 239–268. ISBN: 978-3-030-26951-7 (cit. on p. 23).
- [Zha20] Mark Zhandry. “Schrödinger’s pirate: How to trace a quantum decoder”. In: *Theory of Cryptography Conference*. Springer. 2020, pp. 61–91 (cit. on p. 24).

A Alternate Proof of Simultaneous Quantum Goldreich-Levin

Below we give an alternate⁴⁰ proof of Lemma 23 which does not use the lifting theorem.

Direct proof of Lemma 23. We will adapt the proof⁴¹ of [CLLZ21], originally due to [AC02], to the simultaneous case. We can assume that the token $\rho_{k,x} = |\psi_{k,x}\rangle\langle\psi_{k,x}|$ is a pure state, for the mixed state case follows by convexity. Observe that we can defer any measurements made by $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ until the very end. Accordingly, we can model \mathcal{A} as a unitary map Φ which acts as:

$$\Phi |\psi_{k,x}\rangle |0^m\rangle_{aux} = |\varphi_{k,x}\rangle_{BC},$$

where $|\varphi_{k,x}\rangle_{BC}$ is a bipartite state shared by \mathcal{B} and \mathcal{C} . In phase 2, the key k as well as the random coins (r, r') are revealed and unitary maps $(U_B^{k,r}, U_C^{k,r'})$ are applied by \mathcal{B} and \mathcal{C} , respectively. The resulting state then is given by

$$\begin{aligned} \left(U_B^{k_B,r} \otimes U_C^{k_C,r'} \right) |\varphi_{k,x}\rangle_{BC} &= \left(\alpha_{k,x,r,r'} |\langle r, x \rangle\rangle_B |\langle r', x \rangle\rangle_C |\phi_{k,x,r,r'}^0\rangle_{BC} + \beta_{k,x,r,r'} |\langle r, x \rangle\rangle_B |\overline{\langle r', x \rangle}\rangle_C |\phi_{k,x,r,r'}^1\rangle_{BC} \right. \\ &\quad \left. + \theta_{k,x,r,r'} |\overline{\langle r, x \rangle}\rangle_B |\langle r', x \rangle\rangle_C |\phi_{k,x,r,r'}^2\rangle_{BC} + \gamma_{k,x,r,r'} |\overline{\langle r, x \rangle}\rangle_B |\overline{\langle r', x \rangle}\rangle_C |\phi_{k,x,r,r'}^3\rangle_{BC} \right) \\ &=: |\Gamma_{k,x,r,r'}\rangle, \end{aligned}$$

where $|\phi_{k,x,r,r'}^j\rangle$ is a normalized state for $j \in \{0, 1, 2, 3\}$ and $\alpha_{k,x,r,r'}$ is the coefficient corresponding to the case of the adversary succeeding, so that we can express the assumption as

$$\mathbb{E}_{k,x,r,r'} |\alpha_{k,x,r,r'}|^2 \geq \frac{1}{2} + \varepsilon$$

and hence

$$\mathbb{E}_{k,x,r,r'} |\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2 \geq \mathbb{E}_{k,x,r,r'} \left(2|\alpha_{k,x,r,r'}|^2 - 1 \right) \geq 2\varepsilon \quad (19)$$

We now describe the new adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$:

- Given $|\psi_{k,x}\rangle$ in phase 1, \mathcal{A}' acts the same as \mathcal{A} , i.e. it applies Φ , obtaining the state $|\varphi_{k,x}\rangle_{BC}$.
- After receiving the key k from the challenger (ignoring the random coins received), \mathcal{B}' prepares a uniform superposition over $r \in \mathcal{R}$ and applies the unitary $U_B^{k_B}$, where we define $U_E^{k_E}$ as $U_E^{k_E} |r\rangle |\varphi\rangle = |r\rangle U_E^{k_E,r} |\varphi\rangle$ for $E \in \{B, C\}$. Then, \mathcal{B}' applies a Z gate to the register storing the inner product $\langle r, x \rangle$, and applies $(U_B^{k_B})^\dagger$ to its state. Finally, \mathcal{B}' measures the register storing the random coins r in the Fourier basis and outputs the result.
- \mathcal{C}' is defined in a similar fashion.

⁴⁰Kundu and Tan have independently generalized the Goldreich-Levin technique to the non-local (simultaneous) setting [KT22]. The authors apply this technique to achieve a weaker form of unclonable encryption in the plain model, whereas we apply it to achieve single-decryptor encryption with unclonable security against independently generated ciphertexts.

⁴¹See Lemma B.12 in [CLLZ21].

Next, we will analyze the evolution of the state shared by \mathcal{B}' and \mathcal{C}' step by step. Since the actions of \mathcal{B}' and \mathcal{C}' commute, we can synchronously track their operations. After the first step, the state is given by

$$\left(U_B^{k_B} \otimes U_C^{k_C}\right) \frac{1}{|\mathcal{R}|} \sum_{r,r' \in \mathcal{R}} |r\rangle_B |r'\rangle_C |\varphi_{k,x}\rangle_{BC} = \frac{1}{|\mathcal{R}|} \sum_{r,r' \in \mathcal{R}} |r\rangle_B |r'\rangle_C |\Gamma_{k,x,r,r'}\rangle.$$

Next, \mathcal{B}' and \mathcal{C}' each apply a Z gate to their register storing the inner product, which results in the state

$$\begin{aligned} & \frac{1}{|\mathcal{R}|} \sum_{r,r' \in \mathcal{R}} |r\rangle_B |r'\rangle_C (-1)^{\langle r,x \rangle \oplus \langle r',x \rangle} \left(\alpha_{k,x,r,r'} |\langle r,x \rangle\rangle_B |\langle r',x \rangle\rangle_C |\phi_{k,x,r,r'}^0\rangle_{BC} \right. \\ & - \beta_{k,x,r,r'} |\langle r,x \rangle\rangle_B |\overline{\langle r',x \rangle}\rangle_C |\phi_{k,x,r,r'}^1\rangle_{BC} - \theta_{k,x,r,r'} |\overline{\langle r,x \rangle}\rangle_B |\langle r',x \rangle\rangle_C |\phi_{k,x,r,r'}^2\rangle_{BC} \\ & \left. + \gamma_{k,x,r,r'} |\overline{\langle r,x \rangle}\rangle_B |\overline{\langle r',x \rangle}\rangle_C |\phi_{k,x,r,r'}^3\rangle_{BC} \right) \\ & =: \frac{1}{|\mathcal{R}|} \sum_{r,r' \in \mathcal{R}} |r\rangle_B |r'\rangle_C |\Gamma'_{k,x,r,r'}\rangle, \end{aligned}$$

with

$$\langle \Gamma_{k,x,r,r'}, \Gamma'_{k,x,r,r'} \rangle = (-1)^{\langle r,x \rangle \oplus \langle r',x \rangle} (|\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2).$$

Now \mathcal{B}' and \mathcal{C}' uncompute the unitary $U_B^{k_B} \otimes U_C^{k_C}$, and the state becomes

$$\begin{aligned} & \left(U_B^{k_B} \otimes U_C^{k_C}\right)^\dagger \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} |r\rangle_B |r'\rangle_C |\Gamma'_{k,x,r,r'}\rangle = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} |r\rangle_B |r'\rangle_C \left(U_B^{k_B,r} \otimes U_C^{k_C,r'}\right)^\dagger |\Gamma'_{k,x,r,r'}\rangle \\ & = \frac{1}{|\mathcal{R}|} \sum_{r,r' \in \mathcal{R}} |r\rangle_B |r'\rangle_C \left((-1)^{\langle r,x \rangle \oplus \langle r',x \rangle} (|\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2) |\varphi_{k,x}\rangle_{BC} + |\text{err}_{k,x,r,r'}\rangle \right), \end{aligned}$$

where $|\text{err}_{k,x,r,r'}\rangle$ is a subnormalized state orthogonal to $|\varphi_{k,x}\rangle_{BC}$.

Next, \mathcal{B}' and \mathcal{C}' each apply a Quantum Fourier Transform (QFT) on their random coins, resulting in the state

$$\begin{aligned} & \frac{1}{|\mathcal{R}|^2} \sum_{r,r' \in \mathcal{R}} \sum_{y,z \in \mathcal{R}} (-1)^{\langle r,y \rangle \oplus \langle r',z \rangle} |y\rangle_B |z\rangle_C \left((-1)^{\langle r,x \rangle \oplus \langle r',x \rangle} (|\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 \right. \\ & \left. - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2) |\varphi_{k,x}\rangle_{BC} + |\text{err}_{k,x,r,r'}\rangle \right). \end{aligned}$$

Note that the coefficient of $|x\rangle_B |x\rangle_C |\varphi_{k,x}\rangle$ equals

$$\frac{1}{|\mathcal{R}|^2} \sum_{r,r' \in \mathcal{R}} |\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2,$$

so the probability that \mathcal{B}' and \mathcal{C}' both output x is lower bounded by

$$\begin{aligned}
\Pr [y = z = x] &\geq \mathbb{E}_{k,x} \left| \frac{1}{|\mathcal{R}|^2} \sum_{r,r' \in \mathcal{R}} |\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2 \right|^2 \\
&= \mathbb{E}_{k,x} \left| \mathbb{E}_{r,r'} |\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2 \right|^2 \\
&\geq \left| \mathbb{E}_{k,x,r,r'} |\alpha_{k,x,r,r'}|^2 - |\beta_{k,x,r,r'}|^2 - |\theta_{k,x,r,r'}|^2 + |\gamma_{k,x,r,r'}|^2 \right|^2 \\
&\geq 4\varepsilon^2,
\end{aligned}$$

where we used Cauchy-Schwartz Inequality and eq. (19). □