Collusion Resistant Copy-Protection for Watermarkable Functionalities

Jiahui Liu[1][0000−0003−4380−8168], Qipeng Liu[2][0000−0002−3994−7061], Luowen Qian[3][0000−0002−1112−8822], and Mark Zhandry[4][0000−0001−7071−6272]

[1] University of Texas at Austin, Austin, USA jiahui@cs.utexas.edu
[2] Simons Institute for the Theory of Computing, Berkeley, USA qipengliu0@gmail.com
[3] Boston University, Boston, USA luowenq@bu.edu
[4] Princeton University & NTT Research, Princeton, USA mzhandry@gmail.com

Abstract. Copy-protection is the task of encoding a program into a quantum state to prevent illegal duplications. A line of recent works studied copy-protection schemes under "$1 \to 2$ attacks": the adversary receiving one program copy can not produce two valid copies. However, under most circumstances, vendors need to sell more than one copy of a program and still ensure that no duplicates can be generated. In this work, we initiate the study of collusion resistant copy-protection in the plain model. Our results are twofold:

- The feasibility of copy-protecting all watermarkable functionalities is an open question raised by Aaronson et al. (CRYPTO' 21). In the literature, watermarking decryption, digital signature schemes and PRFs have been extensively studied.
  For the first time, we show that digital signature schemes can be copy-protected. Together with the previous work on copy-protection of decryption and PRFs by Coladangelo et al. (CRYPTO' 21), it suggests that many watermarkable functionalities can be copy-protected, partially answering the above open question by Aaronson et al.
- We make all the above schemes (copy-protection of decryption, digital signatures and PRFs) k bounded collusion resistant for any polynomial k, giving the first bounded collusion resistant copy-protection for various functionalities in the plain model.

1 Introduction

The idea of exploiting the quantum no-cloning principle for building cryptography was pioneered by Wiesner. In his seminal work [27], he proposed the notion of quantum banknotes that cannot be counterfeited due to the unclonability of quantum information. This idea has profoundly influenced quantum cryptography, for example, inspiring the famous work on secure quantum key exchange [10]. Since all classical information is inherently clonable, unclonable cryptography is only achievable through the power of quantum information.

Aaronson [2] further leveraged the capability of no-cloning to achieve copy-protection. The idea of copy-protection is the following. A software vendor

wants to sell a piece of software, abstracted as a classical function f. It prepares a quantum state ρ_f so that anyone with a copy of ρ_f can evaluate f on a polynomial number of inputs. However, no efficient pirate receiving a single copy of ρ_f, could produce two programs that compute f correctly.

The notion above intuitively captures the security of a copy-protection scheme under what we call an "$1 \to 2$ attack": the adversary receives 1 program copy, and attempts to produce 2 copies with the correct functionality. A recent line of works [4,13,14,6] achieve secure copy-protection for various functionalities under $1 \to 2$ attacks.

However, such a security notion is extremely limiting: in most circumstances, we cannot expect the software vendor to issue only one copy of the program. When the vendor gives out multiple copies, all users can collude and generate pirate copies together. Therefore, a useful copy-protection scheme should be secure against any "$k \to k+1$ attack" for any polynomial k. Such security is usually referred to as collusion resistance in the literature.

Prior Works on Copy-Protection We first recall on a high level how most existing copy-protection schemes work: a copy-protection program consists of a quantum state as an "unclonable token", and a classical part containing an obfuscated program (either as an oracle or the output coming out of some obfuscation functionality). The obfuscated program takes in a token and an input one requests to evaluate on; it verifies the validity of the token and if the verification passes, it outputs the evaluation on the requested input. [5]

Until now, collusion resistant copy-protection has essentially been wide open. The only work that considers issuing more than a single program is Aaronson's original work [2], which is proven to be secure in the $k \to k+r$ setting for $r \geq k$ in some structured quantum oracle model. This is undesirable in two ways: (a) it is unclear whether the scheme allows an adversary to double the copies of programs (Aaronson leaves improving r as a challenging open question), which is not a complete break but still potentially devastating to applications; but more importantly, (b) unlike a classical oracle which could be heuristically instantiated using indistinguishability obfuscation, we do not even know how to heuristically instantiate a quantum oracle. Moreover, we believe that any extension of Aaronson's scheme would very likely still require some obfuscation of quantum circuits, since we have evidence that Haar random states, which is the core of Aaronson's scheme, lack the structure that can be verified by a classical circuit [22].

If we turn to the other works constructing copy-protection without using quantum oracles, one naïve idea is to take any such scheme that is $1 \to 2$ secure, and simply generate and hand out multiple copies of ρ_f. It turns out that while this satisfies correctness, they are all trivially broken once two copies are given.

[5] The *general functionality* copy protection schemes in [2,4] and the schemes in [13,7] all satisfy this format. The copy-protection schemes for *point/compute-and-compare functions* in [2,14,6,11] are not necessarily of such a format.

This is because they are all based on quantum states that are unclonable for one copy, but trivially clonable as soon as two copies are given.

To get around this issue, another idea is to instead employ a quantum state that already bears a "$(k \to k+1)$-unclonable" property. However, the only known such states are Haar random states and its computationally (or statistically) close neighbors, such as pseudorandom states (or t-designs), which leads us back to the verification issue without a quantum oracle from before.

Therefore, we raise the natural question: *Is collusion resistant copy-protection feasible, either resisting $k \to k+1$ attacks, or without using a quantum oracle? (Ideally both?)*

Copy-Protection in the Plain Model In this work, we restrict our attention to investigate the question above in the plain model, i.e. we want provably secure protocols without any oracle or heuristics. Unfortunately, it has been known that copy-protection in the plain model even for unlearnable functions is impossible in general [7], and thus we have to further restrict ourselves to construct copy-protection for specific classes of functions that evade the impossibility.

Secure software leasing (SSL) [7] is a weakened notion for copy-protection: in (infinite-term) SSL, the malicious pirate may attempt to make pirate copies as it wants. However, the freeloaders are restricted to running a fixed public quantum circuit on some quantum state produced by the pirate. On the other hand, in copy-protection, the freeloaders are free to execute any quantum circuit that the pirate asks them to. Despite facing the same impossibility as copy-protection, secure software leasing has also been built for various functionalities [7,14,11,4,21]. [6]

Especially, [4,21] showed that secure software leasing for watermarkable functions could be obtained from watermarking and public key quantum money in a black-box way. Watermarking [8] is a primitive that embeds a watermark into a program so that any attempt to remove the watermark would destroy the program's functionality. Observing this, Aaronson et al. [4] raised the following open question: *Can all watermarkable functions also be copy-protected in the plain model?*

In this work, we will use the word "major watermarkable functions" to denote (decrypting) public key encryption, (signing) signatures, and (evaluating) PRFs and only focus on copy-protecting those functionalities. Starting from the work by Cohen et al. [12], a line of works [19,18,20,28,29] focuses on watermarking these three functionalities. Copy-protecting these cryptographic functionalities also has a natural and strong motivation: the ability to evaluate these functions is supposedly private in many circumstances. If owners of a decryption key, signing key, or PRF key can share their key with others, it will trigger severe security concerns. Furthermore, copy-protecting a cryptographic function can lead to copy-protecting a software entity of which this cryptographic function is a component.

[6] The formal security definitions for SSL in [7,14,11,4,21] vary slightly from one to another. We will discuss them in 1.2.

We observe that collusion resistant secure software leasing for watermarkable functions can be achieved as long as the underlying watermarking scheme and quantum money scheme are both collusion resistant, by looking into the construction in [4,21]. (Bounded) collusion resistant watermarking for PRFs, public-key encryptions, etc. are constructed in the plain model [18,28,29, ...] and quantum money can be made collusion resistant with a digital signature on its serial number [3]. This observation seems to suggest that collusion resistant copy-protection could be a much more challenging goal.

1.1 Our Results

In this work, we (partially) answer all of the questions above. In particular, we show how, in the plain model, to construct collusion resistant copy-protection for (decrypting) public-key encryption, (signing) signatures, and (evaluating) PRFs. Our results, together with the prior work on copy-protection of decryption and PRFs (Coladangelo et al. [13]), show that major watermarkable cryptographic functionalities can be copy-protected against even colluding adversaries, in the plain model. We now explain this in more detail.

Collusion Resistant Unclonable Decryption Our first result is collusion resistant copy-protection for decryption keys in a public-key encryption scheme. We refer to such copy-protection scheme as *unclonable decryption* by convention, as first proposed by Georgiou and Zhandry [17].

Theorem 1. *Assuming post-quantum subexponentially secure indistinguishability obfuscation and subexponentially secure LWE, there exists k-bounded collusion resistant unclonable decryption for any polynomial k.*

Our collusion resistant unclonable decryption scheme is based on the construction from the prior work of Coladangelo et al. [13] that achieves the same except with only $1 \to 2$ security. Note that while we require subexponential security, these assumptions match those already required in the prior work. In particular, here, we invoke subexponential security only for a compute-and-compare obfuscation scheme with certain properties as our building block. All the reductions in this work are polynomial.

While we do achieve $k \to k+1$ security, a caveat is that we only achieve "k-bounded collusion resistance", by which we mean that we need a preset number of users k to generate the public key. Still, we consider *all* users as potentially malicious and colluding. We note that this is similar to watermarking decryption circuits of public-key encryption schemes, where to the best of our knowledge, unbounded collusion resistance is also unknown [28,18]. Furthermore, it is foreseeable that bounded collusion resistance suffices in certain enterprise use cases where the number of (partially) authorized parties is a priori known and fixed; furthermore, such tokens can be transferred to a new employee irrevocably.

The main challenges are in the anti-piracy security proof. The prior proof idea for $1 \to 2$ anti-piracy does not translate to the $k \to k+1$ setting. We

present a new view on security reductions to handle a polynomial number of possibly entangled quantum adversaries, which we will elaborate in the technical overview.

Copy-Protecting Watermarkable Functionalities We complement the previous theorem regarding public-key encryption, with the following result on collusion resistant copy-protection for signatures and PRFs:

Theorem 2. *Assuming post-quantum subexponentially secure indistinguishability obfuscation and subexponentially secure LWE, there exists k-bounded collusion resistant copy-protection for digital signatures and PRFs, for any polynomial k.*

We base our construction on the signature token scheme and unclonable PRF in the plain model built in [13] (with $1 \to 2$ anti-piracy). However, our signature scheme is significantly different in two aspects: (a) the signing key in [13] will be consumed after one use whereas our scheme is reusable, and (b) unforgeability breaks down when multiple signature queries can be issued, whereas ours satisfies standard existential unforgeability.

1.2 Related Works

[2] first built copy-protection for all unlearnable functions based on a quantum oracle, with weak collusion resistance. Besides [13] which we have discussed, [4] showed a construction for all unlearnable functions based on a classical oracle. [14,6] constructed copy-protection for point functions and compute-and-compare functions in QROM, the latter improving the security of the former. [7]

Regarding the negative results: [7] demonstrated that it is impossible to have a copy-protection scheme for all unlearnable circuits in the plain model, assuming LWE and quantum FHE. [5] extended this impossibility result to the setting where we allow approximate correctess of the copy-protection program and working in the classical-accessible random oracle model.

[7] put forward secure software leasing (SSL). In the finite-term case, a software vendor would lease a quantum state as the software to a user; later, the user needs to return a part of a bipartite state to the vendor, and the vendor will use its own secret key to verify if this returned state is the one issued in the authentic program. The security guarantees that while passing the above verification, the user should not be able to evaluate the functionality correctly using the other part of its bipartite state executed under a public, fixed quantum circuit eval (specified by the vendor). In the infinite-term case, the user does not need to return the state to the vendor; the security guarantees that it should not produce two states that can both evaluate the function correctly when executed under eval. [7] also built an (infinite-term) SSL scheme for searchable compute-and-compare circuits under iO and LWE.

[7] All constructions discussed in this section are not proved under collusion resistant security unless otherwise specified.

[4] observed that under a definition essentially equivalent to infinite-term SSL, namely copy-detection, one could obtain a black-box construction for infinite-term SSL from watermarking and public-key quantum money. [21] constructed finite-term SSL for PRFs and compute-and-compare functions from (subexponential) LWE, with similar observations.

[11,14] constructed secure software leasing for point functions and compute-and-compare functions; [11] is information-theoretically secure and [14] is secure under QROM. They both used a stronger version of finite-term SSL security: while the vendor will honestly check the returned state from the adversary, the adversary can execute the leftover half of its bipartite state maliciously, i.e., not following the instructions in eval. SSL security of this stronger finite-term variant is only known for point/compute-and-compare functions up till now.

1.3 Technical Overview

We start by showing how to overcome the aforementioned barriers and construct Collusion Resistant Unclonable Decryption (CRUD). As briefly discussed in the introduction, there are challenges to constructing collusion resistant copy-protection based on the so-called "$k \to (k+1)$ no-cloning theorem". Instead, we take a different approach by constructing collusion resistant unclonable decryption CRUD from unclonable decryption UD whose security only holds for "$1 \to 2$ attacks". The construction uses UD in a black-box manner:

- For every $i \in [k]$, sample $(|\mathsf{sk}_i\rangle, \mathsf{pk}_i) \leftarrow \mathsf{UD.KeyGen}$; $|\mathsf{sk}_i\rangle$ will be the i-th copy of the quantum unclonable decryption key; the public key will be $\mathsf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_k)$.
- The encryption algorithm takes a single bit message m and outputs a classical ciphertext ct that consists of k copies of ciphertext, among which the i-th copy ct_i is the ciphertext of m under pk_i.
- To decrypt $\mathsf{ct} = (\mathsf{ct}_1, \cdots, \mathsf{ct}_k)$ with $|\mathsf{sk}_i\rangle$, one can decrypt the i-th ciphertext ct_i.

Intuitively in the above encryption scheme, one can decrypt only if it knows the decryption key for at least one of the public keys. Note that our k decryption keys are sampled independently at random and each state satisfies $1 \to 2$ unclonability. To establish anti-piracy, we want to prove a security reduction from a $k \to k+1$ quantum pirate decryptors to the $1 \to 2$ unclonability of one of the decryption keys.

Unfortunately, we do not know how to prove the security of this scheme generically. As we will elaborate in Section 1.4, we need to open up the construction of the underlying unclonable encryption in order to establish the security.

More importantly, in the following section, we demonstrate that even if we open up the construction and the proof, the proof technique in [13] seems not sufficient for CRUD and we thereby work on a new technique that subsumes that in [13] to complete the proof. We start by recalling the definition of regular UD and the proof in [13].

Regular Unclonable Decryption. Let UD be a regular $(1 \to 2)$ unclonable decryption scheme. For the sake of convenience, we assume the message space is $\{0,1\}$. A pair of a classical public key pk and a quantum unclonable secret key $|\mathsf{sk}\rangle$ is generated by KeyGen.

The anti-piracy security guarantees that no efficient adversary with $|\mathsf{sk}\rangle$ can produce two "working" keys by a CPA indistinguishability standard: if one estimates the success probabilities of both decryption keys on distinguishing a ciphertext of 0 from a ciphertext of 1, their success probabilities cannot be *simultaneously* significantly greater than $1/2$, except with negligible probability. This security notion has been previously studied by Aaronson et al. [4] and Coladangelo et al. [13]

Before we delve into the security proof, it is enlightening to see how this security guarantee is efficiently "falsifiable". Estimating the success probability of a *classical* decryptor is easy. One can generate a ciphertext for a random message using the public key and check whether the classical decryptor is correct on that ciphertext; then, a simple counting estimates its success probability within any inverse polynomial error. Unfortunately, this method does not naturally work in the quantum setting since a single execution of the decryption key (produced by the adversary) may disturb the state and prevent further execution of the same key.

Nevertheless, Zhandry [30] shows that such estimation can be done analogous to the classical setting, inspired by the famous work of Marriot and Watrous [23] for witness-preserving error reduction for quantum Arthur–Merlin game. Informally, the work of Zhandry utilizes a measurement procedure called "projective implementation" (abbreviated as PI)[8] to estimate the success probability of a quantum adversary (see Figure 1).

1. Let $\mathcal{D}$ be a ciphertext distribution we define the procedure with respect to.
2. For any quantum decryptor σ with success probability p over $\mathcal{D}$, running $\mathsf{PI}_{\mathcal{D}}$ on the decryptor produces a probability p' and σ collapses to σ';
3. σ' as a decryptor, has success probability p' over $\mathcal{D}$;
4. Applying $\mathsf{PI}_{\mathcal{D}}$ on σ' always produces p' and σ' remains intact;
5. The expectation of p' is p.

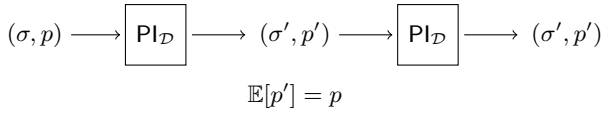

Fig. 1: PI: measure success probability of a decryptor.

[8] For simplicity, we only use the inefficient estimation procedure. The same argument in the technical overview holds using an efficient and approximated version. Similarly for TI.

Put shortly, this measurement procedure will output an estimation of the success probability p' for a quantum decryptor σ. After the measurement, the decryptor collapsed to another decryptor σ', whose success probability is still p'. We will intuitively call PI as "probability estimation' instead of its original name in the scope of the overview.

In the anti-piracy security definition, we care about whether both decryptors have the success probability significantly greater than $1/2$. [13] defines the following "threshold measurement" or "goodness measurement" $\mathsf{TI}_{\mathcal{D},\epsilon}$ for deciding if a quantum decryptor σ is good, for some inverse-polynomial ϵ:

1. Let $\mathcal{D}$ be a ciphertext distribution we define the procedure with respect to.
2. Run $\mathsf{PI}_{\mathcal{D}}$ coherently on σ and measure if the outcome register (containing the resulting probability p') is greater than $1/2+\epsilon$, which produces a single bit outcome b. The quantum decryptor collapses to σ'.
3. If $b = 1$, σ' lies in the span of good decryptors, whose success probability is at least $1/2+\epsilon$; otherwise, σ' is in the subspace with the basis being quantum decryptors whose winning probability is strictly less than $1/2+\epsilon$.

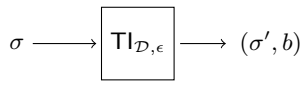

Fig. 2: TI: measure goodness of a decryptor.

We note that $\mathsf{TI}_{\mathcal{D},\epsilon}$ is a projection, which says if σ' is the collapsed decryptor for outcome b, applying $\mathsf{TI}_{\mathcal{D},\epsilon}$ will always produce b and σ' does not change.

We are now ready to formally define the anti-piracy security in [13]. Let $\mathcal{D}$ be the ciphertext distribution for honestly generated ciphertext, which encodes a uniformly random message. No efficient adversary can turn $|\mathsf{sk}\rangle$ into a possibly entangled decryptors σ over two registers, such that applying the threshold measurement $\mathsf{TI}_{\mathcal{D},\epsilon}$ on both decryptors $\sigma[1], \sigma[2]$ will produce two outcomes 1s with non-negligible probability. To put it another way, no efficient adversary can produce two decryptors such that they jointly have non-negligible weight on good decryptors.

Security Proof for "$1 \to 2$ Attacks". Before scoping the proof of our collusion resistant unclonable decryption, we recall the security proof in [13] for "$1 \to 2$ unclonability". In this following section, we will highlight the difficulties of applying the same ideas to CRUD and introduce a new approach to resolve this issue.

The proof works as follows:

– A reduction applies $\mathsf{TI}_{\mathcal{D},\epsilon}$ on both decryptors $\sigma[1], \sigma[2]$. With some non-negligible probability, it will produce two outcomes 1s and the two decryptors become $\sigma'[1], \sigma'[2]$.

- **Extraction on the first register.** Let $\mathcal{D}'$ be the ciphertext distribution for "junk" ciphertext which only encrypts an empty symbol $\perp$. Applying $\mathsf{TI}_{\mathcal{D}',\epsilon}$ on $\sigma'[1]$ always result in outcome 0, whereas the outcome of applying $\mathsf{TI}_{\mathcal{D},\epsilon}$ on $\sigma'[1]$ is always 1.
  We can thereby conclude that $\sigma'[1]$ must contain some secret information about the secret key $|\mathsf{sk}\rangle$. In fact, we can use an extraction algorithm to extract the classical information about the secret key. Note that the algorithm may be destructive that, for example, may measure $\sigma'[1]$ completely.
- **Extraction on the second register.** Conditioned on the successful extraction on $\sigma'[1]$, we want to argue that a similar extraction on the second register works. If so, we can *simultaneously* extract secret information about $|\mathsf{sk}\rangle$ from two non-communicating parties. This will violate the underlying quantum information guarantee[9].

The remaining is to show such an extraction is feasible on the second decryptor, even conditioned on the successful extraction on $\sigma'[1]$. This is because $\mathsf{TI}_{\mathcal{D},\epsilon}$ is a projection, conditioned on the outcome being 1, $\sigma'[2]$ will be in the span of good decryptors (see bullet (3) of the description of TI). Regardless of what event is conditioned on $\sigma'[1]$, the second decryptor is still in the span of good decryptors. Thus, an extraction algorithm would extract the classical information about the secret key from $\sigma'[2]$ with non-negligible probability. This concludes the proof idea in [13].

To conclude, the core idea in the proof is that, a "$1 \to 2$ attack" produces two quantum registers that

1. they have a non-negligible probability $w_1 = \gamma$ on both registers being good decryptors on $\mathcal{D}$ (with success probabilities at least $1/2 + \epsilon$);
2. they have a negligible probability w_2 on both being good decryptors on $\mathcal{D}'$.

If both 1 and 2 are satisfied, a simultaneous extraction succeeds with a non-negligible probability.

In the next few paragraphs, we still denote w_1 as the joint probability of both decryptors being good on distribution $\mathcal{D}$; w_2 as the joint probability of both decryptors being good on distribution $\mathcal{D}'$.

In the above proof for $1 \to 2$ attack, we crucially require w_1 is non-negligible and w_2 is negligible or zero, in order to argue that extraction would succeed even after conditioned on successful extraction on one side.

We can also observe that for the $1 \to 2$ proof, w_2 is automatically zero. As $\mathcal{D}'$ does not encode a real message, no quantum decryptor can achieve any advantage over random guessing. But this is *not* always the case when it turns to our CRUD security proof: for which, $\mathcal{D} = (\mathsf{ct}_\perp, \cdots, \mathsf{ct}_j, \mathsf{ct}_{j+1}, \cdots)$ has the first $(j-1)$ ciphertexts being junk and the rest being real; whereas $\mathcal{D}' = (\mathsf{ct}_\perp, \cdots, \mathsf{ct}_\perp, \mathsf{ct}_{j+1}, \cdots)$ has the first j ciphertexts being junk.

[9] In the actual proof, two non-communicating parties will extract two vectors, one in the primal coset and the other in the dual coset of a coset state. This will violate the strong computational monogamy-of-entanglement property of coset states.

As we will see in the following section, for CRUD, the condition "$w_1 - w_2$ is non-negligible" is the best we can hope for. Therefore, we attempted to see if a proof similar to the above exists, when we can only condition on "$w_1 - w_2$ is non-negligible". Unfortunately, the answer to this attempt is negative, as we will provide some intuition in the immediate next paragraph. We thereby conclude that the proof technique in [13] cannot extend to collusion resistant anti-piracy security proof in a generic way.

To see why the condition "$w_1 - w_2$ is non-negligible" does not necessarily give a simultaneous extraction, we consider the time when a successful extraction has already been done on the first decryptor $\sigma'[1]$. If w_2 is negligible, the leftover state of the second decryptor $\sigma'[2]$ has at most w_2/ζ weight lying in the span of bad decryptors. Here ζ is the probability of a successful extraction on the first decryptor and conditioned on this extraction, the weight w_2 will be amplied by at most $1/\zeta$. Since w_2/ζ is still negligible, this allows an extraction from $\sigma'[2]$ happens with a non-negligible chance. However, if w_2 is not negligible but only satisfies $w_1 - w_2$ is non-negligible, $\sigma'[2]$ can lie in the span of bad decryptors: the extreme case will be the event of successful extraction on $\sigma'[1]$ has "positive correlation" with $\sigma'[2]$ being bad; in this case, the weight can be as large as $w_2/\zeta \approx 1$.

Obstacles for Extraction from Quantum Decryptors. The high-level intuition for why such a construction would satisfy $k \to k+1$ is comprehensible. Assume an adversary uses $|\mathsf{sk}_1\rangle, \cdots, |\mathsf{sk}_k\rangle$ to produce $(k+1)$ (possibly entangled) malicious decryptors σ. Let $\sigma[i]$ denote the i-th pirate decryptor. Since each $\sigma[i]$ is a "working" pirate decryptor, it should at least decrypt one of $\mathsf{ct}_1, \cdots, \mathsf{ct}_k$ (say ct_j). Applying pigeonhole principle, there are two decryptors that decrypts the same ciphertext slot, which would violate $1 \to 2$ unclonability. However, such an intuition is nontrivial to formalize since a quantum adversary could distribute these secret keys in multiple ways in superposition.

A straightforward idea is to extract secret information for the j-th private key $|\mathsf{sk}_j\rangle$ from $\sigma[i]$. Let $\mathcal{D}'$ be the ciphertext distribution $(\mathsf{ct}_\perp, \mathsf{ct}_\perp, \cdots, \mathsf{ct}_\perp)$ containing all junk ciphertext. Clearly, if we apply $\mathsf{TI}_{\mathcal{D}',\epsilon}$ on any quantum decryptor, the result is always 0 (meaning "bad"). If we can find an index j such that $\mathcal{D}_j$ is the distribution $(\mathsf{ct}_\perp, \mathsf{ct}_\perp, \cdots, \mathsf{ct}_j, \cdots, \mathsf{ct}_\perp)$ and applying $\mathsf{TI}_{\mathcal{D}_j,\epsilon}$ on $\sigma[i]$ gives 1 with non-negligible chance, we can extract secrets for $|\mathsf{sk}_j\rangle$ from $\sigma[i]$. If one can extract from every $\sigma[i]$, by the pigeonhole principle, it breaks the underlying quantum information guarantee for one of the unclonable decryption keys. Unfortunately, this idea does not go through, considering the following bad situation.

Even if $\sigma[i]$ has success probability 1, such j may not exist. Consider a quantum program that knows all the decryption keys $|\mathsf{sk}_1\rangle, \cdots, |\mathsf{sk}_k\rangle$ but only decrypts ct if and only if every $|\mathsf{sk}_j\rangle$ can successfully decrypt ct_j; if any decryption fails to decrypt, it outputs a random guess. Feeding $(\cdots, \mathsf{ct}_\perp, \mathsf{ct}_j, \mathsf{ct}_\perp, \cdots,)$ to the decryptor will always result in a random guessing.

Note that this is not only an issue for quantum decryptors but also presents if decryptors are classical.

A natural fix of the above idea is to consider the following hybrid distributions. We define $\mathcal{D}_j$ for every $j \in \{0, 1, \cdots, k\}$:

- $\mathcal{D}_j$: = $(\mathsf{ct}_\perp, \cdots, \mathsf{ct}_\perp, \mathsf{ct}_j, \mathsf{ct}_{j+1} \cdots)$. In other words, only the last $k - j$ ciphertexts encode the same random message $m \in \{0, 1\}$, the first j ciphertexts are junk ciphertexts .
- $\mathsf{TI}_j := \mathsf{TI}_{\mathcal{D}_j,\epsilon}$: the goodness estimation with respect to the ciphertext distribution $\mathcal{D}_j$ and threshold $1/2 + \epsilon$.

That is, each $\mathcal{D}_j$ will replace the first non-junk ciphertext from $\mathcal{D}_{j-1}$ with a junk ciphertext. Note that $\mathcal{D} := \mathcal{D}_0$. By the definition of $\sigma[i]$ is a working decryptor, applying TI_0 on $\sigma[i]$ will produce 1 with a non-negligible probability. On the flip side, applying TI_k on $\sigma[i]$ will always produce 0.

We denote w_j as the probability of applying $\mathsf{TI}_{\mathcal{D}_j,\epsilon}$ on the decryptor $\sigma[i]$ and getting outcome 1. By a standard hybrid argument, we can conclude that there must exist an index $j \in [k]$ such that,

$$w_{j-1} - w_j \text{ is non-negligible.}$$

The gap allows extraction on $\sigma[i]$. However, as we discussed in the last section, it does not satisfy the condition "w_{j-1} is non-negligible and w_j is negligible", which can not guarantee a simultaneous extraction when we consider two decryptors.

A bad example looks like the following: $w_0 = \gamma$ for some inverse polynomial γ and $w_j = \gamma/2^j$ for all $j \neq k$ and $w_k = 0$. There does not exists a j such that w_{j-1} is non-negligible but w_j is negligible.

We now elaborate on our approaches to resolve these obstacles. Our approach directly takes advantage of the probability measure PI instead of TI. This also gives an alternative security proof for the construction in [13].

Extract a Single Decryption Key: Detect a Large Jump in Success Probability Let us start with attempts to extract from a single "working" decryptor σ, using the probability estimation PI. Recall that by the definition of "working", we mean applying $\mathsf{PI}_{\mathcal{D}}$ on σ yields some probability p significantly larger than the trivial guessing probability $1/2$.

We first recall the following ciphertext distributions $\mathcal{D}_j$ and define probability estimation procedure PI_j for every $j \in \{0, 1, \cdots, k\}$:

- $\mathcal{D}_j$: = $(\mathsf{ct}_\perp, \cdots, \mathsf{ct}_\perp, \mathsf{ct}_j, \mathsf{ct}_{j+1} \cdots)$.
- $\mathsf{PI}_j := \mathsf{PI}_{\mathcal{D}_j}$: the probability estimation with respect to the ciphertext distribution $\mathcal{D}_j$.

Now we give the following attempted extraction, which almost works but has one caveat. We call this extraction procedure a "repeated probability estimation/measurement":

1. We first apply PI_0 to σ and obtain p_0 and a collapsed decryption key σ_0.
2. We then apply PI_1 to the collapsed σ_0 to obtain p_1 and σ_1.

   Now if $p_1 - p_0$ is at least $\frac{p_0 - \frac{1}{2}}{k}$, we perform an extraction procedure to extract secrets for $|\mathsf{sk}_1\rangle$ from σ_0. Intuitively, since we observe a noticeable probability decrease when ct_1 is replaced with junk ciphertext, there must be some part of $\sigma[i]$ that uses ct_1 to recover the original plaintext. We then abort the procedure.
3. Otherwise, p_0 and p_1 should be negligibly close. We again apply PI_2 on σ_1 and obtain p_2, σ_2. If $p_2 - p_1$ is at least $\frac{p_0 - \frac{1}{2}}{k}$, we perform extraction on σ_1 and abort.
4. We continue this process for all $j = 3, ..., k$.

We claim that the above repeated measurement procedure will always terminate at some $j \in [k]$. To see this, think of $p_1, ..., p_k$ as a sequence of random variables, whose values are only observed when the corresponding measurement is applied. Note that $p_k = 1/2$ always, because the underlying ciphertext distribution $\mathcal{D}_k$ encodes all junk ciphertexts, so no adversary can achieve better advantage than guessing. Therefore, the claim follows from triangle inequality.

$$(\sigma_0, p_0) \xrightarrow{\mathsf{PI}_1} (\sigma_1, p_1) \xrightarrow{\mathsf{PI}_2} (\sigma_2, p_2) \xrightarrow{\mathsf{PI}_3} \cdots \xrightarrow{\mathsf{PI}_{k-1}} (\sigma_k, p_k)$$

The above extraction procedure almost works. But it is actually not physically executable: we need σ_{j-1} in order to perform extraction as that is the state with a "working" component for ciphertext ct_j, but by the time that we decide to extract, we already get to state σ_j because we have to obtain measurement outcome p_j to claim a jump in probability happens. It is generally infeasible to rewind a quantum state, in this case from σ_j to σ_{j-1}.[10]

Fortunately, it is plausible for a single decryptor: we guess j (denoting the first index having a probability jump) and stop the procedure when we have done $\mathsf{PI}_0, \cdots, \mathsf{PI}_{j-1}$. With probability at least $1/k$, we can extract for $|\mathsf{sk}_j\rangle$ from the current decryptor σ_{j-1}. We will get to why this procedure avoids the rewinding issue and preserves our success probability, when it comes to the $(k+1)$ decryptors case in the next paragraph.

Extending to $(k+1)$ decryptors. Finally, we show how to generalize the above extraction strategy to extracting secrets from the same key $|\mathsf{sk}_j\rangle$.

We apply the repeated measurement individually to every decryptor: that is, for the i-th decryptor, we apply $\mathsf{PI}_0, \mathsf{PI}_1, \cdots, \mathsf{PI}_k$, one upon another. The procedure will yield $p_{i,0}, p_{i,1}, \cdots, p_{i,k}$. Since $p_{i,0}$ is always greater than $1/2 + \gamma$ and $p_{i,k} = 1/2$, there must exist a large probability gap between p_{i,j_i-1} and p_{i,j_i} for some $j_i \in [k]$. By the pigeonhole principle, for some $x \neq y$, $j := j_x = j_y$. We hope to stop at the x-th and y-th decryptors before applying PI_j and simultaneously turn them into two keys for ct_j.

[10] The probability estimation PI_j will preserve the success probability of the state but nothing else. Applying PI_j will likely change σ_{j-1}.

Since there will always be two decryptors having large probability gaps for the same index, the chance of having such gaps for randomly guessed x, y and j is at least $\frac{1}{\binom{k+1}{2}k} \geq 1/k^3$. But the success probability of this guess is not immediately guaranteed, because we need to stop before the j-th probability estimation for states $\sigma[x], \sigma[y]$ otherwise we can't rewind to this state needed for extraction. We are still two unpredictable measurements away from the event we guess for. Fortunately, guessing and stopping before the j-th PI will indeed work with probability at least $1/(2k^3)$, through a trick for randomized algorithms.

Now we can apply repeated measurement and stop before applying PI_j on any of these two decryptors. Let the leftover decryptors be $\sigma^*[x, y]$ and the last probability outcomes be $p_{x,j-1}$ and $p_{y,j-1}$. With probability at least $1/(2k^3)$, $(\sigma^*[x, y], p_{x,j-1}, p_{y,j-1})$ satisfy the following conditions (*) and (**):

(*) Applying PI_{j-1} on both $\sigma^*[x]$ and $\sigma^*[y]$ always produces $p_{x,j-1}$ and $p_{y,j-1}$.
(**) Applying PI_j on both $\sigma^*[x]$ and $\sigma^*[y]$, with probability at least $1/(2k^3)$, produces large probabilities gaps for both $p_{x,j-1}$ and $p_{y,j-1}$.

It seems that we have come to the right "spot" for extraction. However, we still face a challenge. How do we guarantee that we can simultaneously extract from two possibly entangled states? A possible malicious behavior is that measuring one decryptor's key will collapse the other decryptor to a "not working" state.

We can clearly extract secrets for $|\mathsf{sk}_j\rangle$ from either $\sigma^*[x]$ or $\sigma^*[y]$: since there is a probability gap, it must mean $\sigma^*[x]$ (or $\sigma^*[y]$) use ct_j for decryption at some point. From the probability point of view, we then argue why simultaneous extraction is feasible.

Define $\mathbf{E}_x$ ($\mathbf{E}_y$, here $\mathbf{E}$ stands for "(E)xtraction") be the event of a successful extraction on the x-th decryptor (or on the y-th decryptor respectively). Define $\mathbf{G}_x$ ($\mathbf{G}_y$, here $\mathbf{G}$ stands for "(G)ap") be the event that applying PI_j on the x-th decryptor (or on the y-th decryptor respectively) yields a large probability gap. We will prove $\Pr[\mathbf{E}_x \wedge \mathbf{E}_y]$ is non-negligible by contradiction.

It is clear that $\Pr[\mathbf{E}_x]$ is non-negligible. To show $\Pr[\mathbf{E}_y|\mathbf{E}_x]$ is non-negligible, it is sufficient to show that $\Pr[\mathbf{G}_y|\mathbf{E}_x]$ is non-negligible, since a large gap implies a large chance of extraction.

We can intuitively think of $\Pr[\mathbf{E}_x] = 0.1 \Pr[\mathbf{G}_x]$ and $\Pr[\mathbf{E}_y] = 0.1 \Pr[\mathbf{G}_y]$[11]. We may expect that $\Pr[\mathbf{E}_x \wedge \mathbf{E}_y] = 0.1 \Pr[\mathbf{G}_x \wedge \mathbf{G}_y]$, which would conclude the proof. However, this does not follow immediately from above as it could be the case that $\mathbf{G}_x \wedge \mathbf{G}_y$ occurs with non-negligible probability, but $\mathbf{E}_x \wedge \mathbf{E}_y$ never occurs. The main insight here is that we can instead show that $\Pr[\mathbf{E}_x|\mathbf{G}_y] = 0.1 \Pr[\mathbf{G}_x|\mathbf{G}_y]$, as finding the gap for y does not impact the extraction for x. Invoking Bayes' rule, this shows that $\Pr[\mathbf{G}_y|\mathbf{E}_x] = \Pr[\mathbf{E}_x|\mathbf{G}_y] \Pr[\mathbf{G}_y] / \Pr[\mathbf{E}_x]$ is non-negligible as well. As a consequence, $\Pr[\mathbf{E}_y|\mathbf{E}_x]$ and thus $\Pr[\mathbf{E}_x \wedge \mathbf{E}_y]$ (simultaneous extraction) are both large.

[11] The choice of 0.1 is arbitrary here. Indeed, they are polynomially related. For the sake of simplicity, we assume they are linearly related.

Collusion Resistant Copy-Protection for Signatures and PRFs Now with the building block of collusion resistant unclonable decryption, we come to copy-protect more cryptographic functions.

As briefly discussed in the introduction, even though [13] presented the first unclonable signature scheme without oracles, its scheme is a signature token that will be consumed after one use. One-time signature is a security notion interesting under many circumstances [9,17], but it's crucial that we investigate the possibility of copy-protecting a standard digital signature. Moreover, once achieved, this construction helps us get closer to the goal of copy-protecting all watermarkable functionalities.

The [13] signature token is one-time because when signing a message, the signer simply measures the quantum key and the measurement outcome is a signature. It is not existentially unforgeable for the same reason: if an adversary gets a few random measurement results of quantum keys, he is granted the power to sign, without the need of an intact quantum key.

To resolve the problem, we resort to the classic picture of generic copy-protection: the signing program first verifies if a quantum key is a valid "token" and then outputs a signature (computed independently of the quantum key) as well as the almost unharmed key. In particular, we observe that the unclonable decryption scheme in [13] will pave the way for such a construction. Their scheme can be extended to a copy-protection for evaluating puncturable PRFs with the " hidden trigger" technique from [25]. Meanwhile, such PRF evaluation functionality can be used as a signing program after obfuscation.

We thereby give a copy-protection for existentially unforgeable, publicly-verifiable signature scheme, based on the above ideas. Along the way, we deal with a few subtleties that emerge because we need public verification and generalization to collusion resistance. More specifically, we present a k-party version of the [25] hidden trigger technique to obtain both collusion resistant copy-protection for signatures and for PRFs.

1.4 Discussions and Open Problems

Comparisons to [13]. An informed reader may claim that one main obstacle (namely simultaneous extraction) for proving anti-piracy security in this paper resembles the obstacle in the $1 \to 2$ anti-piracy schemes of [4,13]. We emphasize that while this issue may be bumped into in all quantum copy-protection proofs, our approach of resolving the issue is different from previous works, especially to identify gaps in a repeated probability estimation procedure (see more details in the technical overview). In particular, our approach can be used to prove security for the schemes in [4] and [13], but as we have discussed in the technical overview, their techniques will *not* work for the $k \to k+1$ setting [12]

[12] The approach for simultaneous extraction when showing $1 \to 2$ anti-piracy in [4] bears a high-level similarity with [13]. We have discussed [13] in the overview since we focus on unclonable decryption.

On Non-Black-Box Reduction. In the technical overview, we describe a black-box way of reducing "$k \to (k+1)$ security" to "$1 \to 2$ security". As mentioned earlier, we cheat in the technical overview and the approach is not entirely black-box.

A high-level summary for the reason is: a black-box reduction algorithm (i.e. an adversary for a $1 \to 2$ unclonable decryption scheme) is not able to generate the correct distribution for the ciphertext to feed to the k collusion resistant adversary. Elaborated as follows:

First, recall that in a k collusion resistant scheme, an encryption for a message m is an ensemble of ciphertexts $\mathsf{ct} = (\mathsf{ct}_1, ..., \mathsf{ct}_k)$ where $\mathsf{ct}_i = \mathsf{Enc}(\mathsf{pk}_i, m)$ for all $i \in [k]$.

In the reduction, we want to apply $\mathsf{PI}_{\mathcal{D}_j}$ on a malicious decryptor to extract secrets from $|\mathsf{sk}_j\rangle$ for some $j \in [k]$:

$\mathcal{D}_j$: the first j ciphertexts (that is, $\mathsf{ct}_1, \cdots$ up to ct_j) are simulated ciphertexts, the rest of them encrypt the same message.

The problem is the following: the reduction only gets a single ciphertext ct_{j+1}, whereas the malicious decryptor takes input of the form in $\mathcal{D}_j$. The reduction needs to generate other ciphertext on its own: including those simulated and those encrypting the same message as c_{j+1}. Since the reduction does not know which message is encrypted in ct_{j+1} (otherwise, the reduction itself already breaks the security of the underlying $1 \to 2$ unclonable decryption), it cannot generate a valid ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \cdots, \mathsf{ct}_k)$ from the distribution $\mathcal{D}_j$.

Therefore, we need to open this proof up in a non-black-box way: it's based on the security of coset states. When we break the security of coset states, the message (encrypted in ct_{j+1}) is known by the reduction. In fact, it is even sampled by the reduction R.

Open Problems. The main limitation of our constructions is that the number of collusions is bounded to a polynomial specified during setup, and the parameters grow with the collusion bound. Because of this collusion bound, our results are technically incomparable to [2], which, despite having a much weaker copy-protection guarantee and using a strong oracle, required no prefixed user number. We leave achieving unbounded $k \to k+1$ collusion resistance as an interesting open question.

1.5 Organization

The rest of the paper is organized as follows. In Section 2, we recall the definitions and properties of coset states and how to measure success probabilities of quantum adversaries. In Section 3, we present the definition, construction, and security proof of collusion resistant unclonable decryption. Our constructions and security proofs for (collusion resistant) copy-protection for signature schemes and PRFs are covered in the full version.

2 Preliminaries

In this paper, λ denotes the security parameter. $\mathsf{poly}(\cdot)$ denotes a polynomial function. We say a function $f(\cdot) : \mathbb{N} \to \mathbb{R}^{\geq 0}$ is negligible if for all constant $c > 0$, $f(n) \leq \frac{1}{n^c}$ for all sufficiently large n. $\mathsf{negl}(\cdot)$ denotes a negligible function. Similarly, we say a function $f(\cdot) : \mathbb{N} \to \mathbb{R}^{\geq 0}$ is sub-exponential if there exists a constant $c < 1$, such that $f(n) \leq 2^{n^c}$ for all sufficiently large n. $\mathsf{subexp}(\cdot)$ denotes a sub-exponential function. For an integer k, We denote $\{1, 2, \cdots, k\}$ by $[k]$. We denote $\mathbb{F}_2$ to be the binary field.

We refer the reader to [24] for a reference of basic quantum information and computation concepts. We also leave the definition for indistinguishability obfuscation in the full version. Readers can also find the definition in [8,16].

2.1 Coset States

We recall the notion of coset states, introduced by [26] and later studied by [13] in the setting of quantum copy-protection. We then present a property of coset states: a strong computational monogamy-of-entanglement (MOE) property. This property is used to obtain an unclonable decryption scheme and other copy-protection of watermarkable cryptographic primitives in this work. Some part of this section is taken verbatim from [13].

Definitions For any subspace A, its complement is $A^\perp = \{b \in \mathbb{F}_2^n \,|\, \langle a, b\rangle = 0\,,\, \forall a \in A\}$. It satisfies $\dim(A) + \dim(A^\perp) = n$. We also let $|A| = 2^{\dim(A)}$ denote the number of elements in the subspace A.

Definition 1 (Coset States). *For any subspace $A \subseteq \mathbb{F}_2^n$ and vectors $s, s' \in \mathbb{F}_2^n$, the coset state $|A_{s,s'}\rangle$ is defined as:*

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

By applying $H^{\otimes n}$ (Hadamard on every qubit) on the state $|A_{s,s'}\rangle$, one obtains exactly $|A^\perp_{s',s}\rangle$. Given A, s and s', there is an efficient quantum algorithm that generates $|A_{s,s'}\rangle$, by [13].

For a subspace A and vectors s, s', we define cosets $A + s = \{v + s : v \in A\}$, and $A^\perp + s' = \{v + s' : v \in A^\perp\}$. It is also convenient for later sections to define a canonical representative, with respect to subspace A, of the coset $A + s$.

Definition 2 (Canonical Representative of a Coset). *For a subspace A, we define the function $\mathsf{Can}_A(\cdot)$ such that $\mathsf{Can}_A(s)$ is the lexicographically smallest vector contained in $A + s$ (we call this the canonical representative of coset $A + s$).*

[13] showed that, Can_A and $\mathsf{Can}_{A^\perp}$ are efficiently computable given the classical description of A.

When it is clear from the context, we will write $A + s$ to denote the *program* that checks membership in $A + s$. The following equivalences, which follow

straightforwardly from the security of iO, will be useful in our security proofs later on.

Proposition 1. *For any subspace* $A \subseteq \mathbb{F}_2^n$, $\mathsf{iO}(A+s) \approx_c \mathsf{iO}(\mathsf{CC}[\mathsf{Can}_A, \mathsf{Can}_A(s)])$. *Recall that* $\mathsf{CC}[\mathsf{Can}_A, \mathsf{Can}_A(s)]$ *refers to the compute-and-compare program which on input* x *outputs* 1 *if and only if* $\mathsf{Can}_A(x) = \mathsf{Can}_A(s)$.

This is due to the fact that $A+s$ has the same functionality as $\mathsf{CC}[\mathsf{Can}_A, \mathsf{Can}_A(s)]$. The lemma then follows the security of iO.

Strong Monogamy-of-Entanglement Property Consider a game between a challenger and an adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$:

- The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $\frac{n}{2}$, and two uniformly random elements $s, s' \in \mathbb{F}_2^n$. It sends $|A_{s,s'}\rangle$, $\mathsf{iO}(A+s)$, and $\mathsf{iO}(A^\perp + s')$ to $\mathcal{A}_0$.
- $\mathcal{A}_0$ creates a bipartite state on registers B and C. Then, $\mathcal{A}_0$ sends register B to $\mathcal{A}_1$, and C to $\mathcal{A}_2$.
- The classical description of A is then sent to both $\mathcal{A}_1, \mathcal{A}_2$.
- $\mathcal{A}_1$ and $\mathcal{A}_2$ return respectively s_1 and s_2.

$(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ wins if and only if $s_1 \in A + s$ and $s_2 \in A^\perp + s'$.

Let $\mathsf{CompStrongMonogamy}((\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2), n)$ be a random variable which takes the value 1 if the game above is won by adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, and takes the value 0 otherwise.

Theorem 3. *Assuming the existence of sub-exponentially secure post-quantum* iO *and one-way functions, then for any QPT adversary* $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$,

$$\Pr[\mathsf{CompStrongMonogamy}((\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2), n) = 1] \leq 1/\mathsf{subexp}(n)\,.$$

[15] proved an information-theoretic version of the strong monogamy property (without giving out the iO programs to the adversary). [13] showed that one can obtain the computational statement by lifting the information-theoretic statement.

2.2 Measure Success Probabilities of Quantum Adversaries: Projective/Threshold Implementation

In this section, we include several definitions and results about estimating success probabilities or estimating whether the probability is above a threshold. Part of this section is taken verbatim from [4,13]. In this section, we will mainly talk about how to measure probability in an inefficient way. The proofs in the main body of the proof use this inefficient measuring procedure as subroutines. All these proofs can be translated easily using the efficient version of such measuring procedures. We will cover those in the full version.

Estimating success probabilities of adversaries is essential in many settings, especially for a reduction to know whether the adversary is good or if an extraction on the adversary can succeed with high probability. Classically it is easy. Let $\mathcal{D}$ be a testing input distribution and C be a classical program for which we want to estimate probability. We can keep running C on uniformly fresh inputs sampled from $\mathcal{D}$ to estimate the probability up to any inverse polynomial error. Such procedure is infeasible for quantum adversaries, since a single execution of a quantum program may completely collapse the program, leading to failure for future executions.

Projective Implementation Zhandry [30] formalizes the following probability measurement procedure for a quantum program ρ under some test distribution $\mathcal{D}$.

Consider the following procedure as a binary POVM $\mathcal{P}_{\mathcal{D}} = (P_{\mathcal{D}}, Q_{\mathcal{D}})$ acting on a quantum program ρ (whose success probability is equal to p): sample an input x from $\mathcal{D}$, evaluates the quantum program ρ on x, and checks if the output is correct. Let $P_{\mathcal{D}}$ denote the operator for output being correct and $Q_{\mathcal{D}}$ be the quantum operator for the output being incorrect.

Zhandry proposed a procedure that applies an appropriate projective measurement which *measures* the success probability of ρ on input $x \leftarrow \mathcal{D}$, and outputs the probability p'. Conditioned on the outcome is some probability p', the quantum program collapsed to ρ' whose success probability is exactly p'. Furthermore, the expectation of p' equals to p.

Theorem 4 (Projective Implementation). *Let $\mathcal{D}$ be a distribution of inputs. Let $\mathcal{P}_{\mathcal{D}} = (P_{\mathcal{D}}, Q_{\mathcal{D}})$ be a binary outcome POVM described above with respect to the distribution $\mathcal{D}$. There exists a projective measurement $\mathsf{PI}(\mathcal{P}_{\mathcal{D}})$ such that for any quantum program ρ with success probability p on $\mathcal{D}$:*

(i) Applying $\mathsf{PI}(\mathcal{P}_{\mathcal{D}})$ on ρ yields ρ', p'.
(ii) ρ' has success probability p' with respect to $\mathcal{D}$. Furthermore, applying $\mathsf{PI}(\mathcal{P}_{\mathcal{D}})$ on ρ' always produces p'.
(iii) The expectation of p' equals to p.

We say the above measurement procedure is a projective implementation of $\mathcal{P}_{\mathcal{D}}$. When the distribution is clear from the context, we sometimes ignore the subscript $\mathcal{D}$ in both $\mathcal{P}_{\mathcal{D}}$ and $\mathsf{PI}(\mathcal{P}_{\mathcal{D}})$.

Threshold Implementation The concept of threshold implementation [4] is similar to projective implementation, except it now outputs a binary outcome indicating whether the probability is above or below some threshold.

Theorem 5 (Threshold Implementation). *Let $\mathcal{D}$ be a distribution of inputs. Let $\mathcal{P}_{\mathcal{D}} = (P_{\mathcal{D}}, Q_{\mathcal{D}})$ be a binary outcome POVM described above with respect to the distribution $\mathcal{D}$. For any $0 \leq \gamma \leq 1$, there exists a projective measurement $\mathsf{TI}_{\gamma}(\mathcal{P}_{\mathcal{D}})$ such that for any quantum program ρ:*

(i) Applying $\mathsf{TI}_{\gamma}(\mathcal{P}_{\mathcal{D}})$ on ρ yields a binary outcome b' and a collapsed program ρ'.

(ii) If $b' = 1$, ρ' has success probability at least γ with respect to $\mathcal{D}$. Furthermore, applying $\mathsf{TI}_\gamma(\mathcal{P}_\mathcal{D})$ on ρ' always produces 1.
(iii) If $b' = 0$, ρ' has success probability less than γ with respect to $\mathcal{D}$. Furthermore, applying $\mathsf{TI}_\gamma(\mathcal{P}_\mathcal{D})$ on ρ' always produces 0.

We say the above measurement procedure is a threshold implementation of $\mathcal{P}_\mathcal{D}$ with threshold γ. When the distribution is clear from the context, we sometimes ignore the subscript $\mathcal{D}$ in $\mathsf{TI}(\mathcal{P}_\mathcal{D})$.

Moreover, $\mathsf{TI}(\mathcal{P}_\mathcal{D})$ can be implemented by first applying $\mathsf{PI}(\mathcal{P}_\mathcal{D})$ to get a outcome p and outputting 1 if $p \geq \gamma$ or 0 otherwise.

For simplicity, we denote by $\mathrm{Tr}[\mathsf{TI}_\gamma(\mathcal{P}_\mathcal{D})\,\rho]$ the probability that the threshold implementation applied to ρ **outputs 1**. Thus, whenever $\mathsf{TI}_\gamma(\mathcal{P}_\mathcal{D})$ appears inside a trace Tr, we treat $\mathsf{TI}_\gamma(\mathcal{P}_\mathcal{D})$ as a projection onto the 1 outcome.

The approximate and efficient versions of both PI and TI will be covered in the full version

3 Collusion Resistant Unclonable Decryption

In this section, we give the formal definition of collusion resistant unclonable decryption. We will then show the construction for achieving bounded collusion resistance for any k — polynomial number of parties. Finally, we prove the construction satisfies correctness, semantic security and anti-piracy against colluding adversaries. Our scheme has security against bounded number of parties. It requires to know the parameter k in the setup phase and only k copies of keys can be generated later. Furthermore, the public key, secret key and ciphertext have length linear in the number of parties k. Note that our scheme is secure even if an adversary takes control of all copies of decryption keys; the adversary still can not produce any additional functioning key.

3.1 Definitions

Definition 3 (Bounded Collusion Resistant Unclonable Decryption Scheme). *A bounded collusion resistant unclonable decryption scheme* CRUD *for a message space $\mathcal{M}$ consists of the following efficient algorithms:*

- $\mathsf{Setup}(1^\lambda, k) \to (\mathsf{sk}, \mathsf{pk})$: *a (classical) probabilistic polynomial-time (in λ, k) algorithm that takes as input an upper bound k on the number of users and a security parameter λ and outputs a classical secret key* sk *and a classical public key* pk.
- $\mathsf{QKeyGen}(\mathsf{sk}) \to \rho_{\mathsf{sk},1} \otimes \rho_{\mathsf{sk},2} \otimes \cdots \otimes \rho_{\mathsf{sk},k}$: *a quantum algorithm that takes as input a secret key* sk *and outputs k copies of quantum secret keys.*
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$: *a (classical) probabilistic algorithm that takes as input a public key* pk*, a message m and outputs a classical ciphertext* ct.
- $\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct}) \to m/\perp$: *a quantum algorithm that takes as input a quantum secret key ρ_{sk} and a classical ciphertext* ct*, and outputs a message m or a decryption failure symbol $\perp$.*

Here 'bounded' refers to the restriction that the Setup procedure requires to know the maximal number of keys distributed in the QKeyGen.

A bounded collusion resistant unclonable decryption scheme should satisfy the following:

Correctness: For every polynomial $k(\cdot)$, there exists a negligible function $\mathsf{negl}(\cdot)$, for all $\lambda \in \mathbb{N}$, let $k := k(\lambda)$, for all $m \in \mathcal{M}$, all $i \in [k]$,

$$\Pr\left[\mathsf{Dec}(\rho_{\mathsf{sk},i}, \mathsf{ct}) = m \;\middle|\; \begin{array}{c} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, k), \\ \rho_{\mathsf{sk},1} \otimes \cdots \otimes \rho_{\mathsf{sk},k} \leftarrow \mathsf{QKeyGen}(\mathsf{sk}), \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

In other words, correctness says the i-th quantum decryption key will always decrypt correctly (except with negligible probability). By the gentle measurement lemma [1], each decryption key can function correctly polynomially many times for honestly generated encryptions.

CPA Security: This is the regular semantic security for an encryption scheme. An adversary without getting any decryption key (neither sk nor these quantum keys) can not distinguish ciphertexts of chosen plaintexts.

Formally, for every (stateful) QPT adversary $\mathcal{A}$, for every polynomial $k(\cdot)$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, the following holds:

$$\Pr\left[\begin{array}{c} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, k) \\ \mathcal{A}(\mathsf{ct}) = b : ((m_0, m_1) \in \mathcal{M}^2) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk}) \\ b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

Anti-Piracy Security Finally, we define anti-piracy against colluding adversaries. Anti-piracy intuitively says there is no adversary who gets all copies of the decryption keys can successfully produce one additional "working" key.

We will follow the two different definitions of "working" proposed in [13] and give two definitions for anti-piracy. The first definition allows a pirate to announce two messages (m_0, m_1), much like the semantic security. A decryption key is good if an adversary can distinguish encryptions of m_0 and m_1 by using the decryption key. The second definition of a "working" decryption key is basing on whether it decrypts correctly with high probability on uniformly random inputs.

Before describing the security games, we first recall the concept of a quantum decryptor (or a quantum decryption key) [13] with respect to a collusion resistant unclonable decryption scheme.

Definition 4 (Quantum Decryptor). *A* quantum decryptor ρ *for ciphertexts of length m, is an ℓ-qubit state for some polynomial ℓ. For a ciphertext c of length m, we say that we run the quantum decryptor ρ on ciphertext c to mean that we execute a universal quantum circuit U on inputs $|c\rangle$ and ρ, and measure the output registers.*

We are now ready to describe the CPA-style anti-piracy game as well as the random challenge anti-piracy game. We first introduce the notion of good decryptors with respect to two messages (m_0, m_1).

Definition 5 ($(\frac{1}{2}+\gamma)$-good Test with respect to (m_0, m_1)). *Let $\gamma \in [0, 1/2]$. Let pk be a public key, and (m_0, m_1) be a pair of messages. We refer to the following procedure as a test for a γ-good quantum decryptor with respect to pk and (m_0, m_1):*

- *The procedure takes as input a quantum decryptor ρ.*
- *Let $\mathcal{P} = (P, I - P)$ be the following POVM acting on some quantum state ρ':*
  - *Sample a uniform $b \leftarrow \{0, 1\}$ and random coins r. Compute $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b; r)$.*
  - *Run the quantum decryptor on input c. Check whether the outcome is m_b. If so, output 1; otherwise output 0.*
- *Let $(\mathsf{TI}_{1/2+\gamma}, I - \mathsf{TI}_{1/2+\gamma})$ be the threshold implementation of $\mathcal{P}$ with threshold value $\frac{1}{2} + \gamma$, as defined in Theorem 5. Run the threshold implementation on ρ, and output the outcome. If the output is 1, we say that the test passed, otherwise the test failed.*

Definition 6 (k-Strong-Anti-Piracy Game, CPA-style). *Let $\lambda, k \in \mathbb{N}^+$. The CPA-style strong anti-piracy game for a collusion resistant unclonable decryption scheme is the following game between a challenger and an adversary $\mathcal{A}$.*

1. ***Setup Phase**: The challenger samples keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, k)$.*
2. ***Quantum Key Generation Phase**: The challenger sends $\mathcal{A}$ the classical public key pk and all k copies of quantum decryption keys $\rho = \rho_{\mathsf{sk},1} \otimes \cdots \rho_{\mathsf{sk},k} \leftarrow \mathsf{KeyGen}(\mathsf{sk})$.*
3. ***Output Phase**: $\mathcal{A}$ outputs a pair of distinct messages (m_0, m_1). It also outputs a (possibly mixed and entangled) state σ over $k+1$ registers $R_1, R_2, \cdots, R_{k+1}$. We interpret σ as $k+1$ (possibly entangled) quantum decryptors $\sigma[R_1], \cdots, \sigma[R_{k+1}]$.*
4. ***Challenge Phase:** Let $\mathsf{TI}_{1/2+\gamma}$ be the $(\frac{1}{2}+\gamma)$-good test with respect to (m_0, m_1). The challenger applies $\mathsf{TI}_{1/2+\gamma}$ to each of these decryptors. The challenger outputs 1 if and only if all the measurements output 1.*

We denote by $\mathsf{StrongAntiPiracyCPA}(1^\lambda, 1/2+\gamma, k, \mathcal{A})$ a random variable for the output of the game.

Definition 7 (Strong Anti-Piracy-Security). *Let $\gamma : \mathbb{N}^+ \to [0, 1]$. An unclonable decryption scheme satisfies strong γ-anti-piracy security, if for any polynomial $k(\cdot)$, for any QPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds for all $\lambda \in \mathbb{N}$:*

$$\Pr\left[b = 1, b \leftarrow \mathsf{StrongAntiPiracyCPA}(1^\lambda, 1/2+\gamma(\lambda), k(\lambda), \mathcal{A})\right] \leq \mathsf{negl}(\lambda) \quad (1)$$

Note that the above strong anti-piracy security is defined by the threshold implementation TI. By [13], this definition implies a weaker notion called *regular CPA-style anti-piracy security*, which says the probability of all $k+1$ malicious parties simultaneously distinguish encryptions of m_0 or m_1 (m_0 and m_1 are chosen independently for each malicious parties) is at most negligibly greater than $1/2$.

We can similarly define *regular anti-piracy security with random message challenges*: the probability of all $k+1$ malicious parties simultaneously recover ciphertext of independent random messages is at most negligibly greater than $1/2^n$, where n is the message length.

3.2 Construction

We now give the construction of our collusion resistant unclonable decryption. Let UD be the unclonable decryption scheme based on coset states [13]. Our CRUD takes k as input and outputs k pairs of freshly generated keys for UD. A message is encrypted under each public key. Decryption works if a decryptor can decrypt any ciphertext. The construction of CRUD follows from the construction of UD. The security of our CRUD requires a non-black-box analysis for the last step.

$\mathsf{CRUD.Setup}(1^\lambda, k)$:
- For $i \in [k]$, $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{UD.Setup}(1^\lambda)$.
- Let $\mathsf{sk} = (\mathsf{sk}_1, \cdots, \mathsf{sk}_k)$ and $\mathsf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_k)$. Output $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{CRUD.QKeyGen}(\mathsf{sk})$:
- Parse $\mathsf{sk} = (\mathsf{sk}_1, \cdots, \mathsf{sk}_k)$. Let $\rho_i \leftarrow \mathsf{UD.QKeyGen}(\mathsf{sk}_i)$.
- Let $\rho_{\mathsf{sk},i}$ be ρ_i padded with a classical index i, i.e., $\rho_{\mathsf{sk},i} = \rho_i \otimes |i\rangle\langle i|$.
- Output $\rho_{\mathsf{sk},1} \otimes \cdots \otimes \rho_{\mathsf{sk},k}$.

$\mathsf{CRUD.Enc}(\mathsf{pk}, m)$:
- Parse $\mathsf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_k)$. Let $\mathsf{ct}_i \leftarrow \mathsf{UD.Enc}(\mathsf{pk}_i, m)$.
- Output $\mathsf{ct}_1, \cdots, \mathsf{ct}_k$.

$\mathsf{CRUD.Dec}(\rho_{\mathsf{sk}}, \mathsf{ct})$:
- Parse $\mathsf{ct} = (\mathsf{ct}_1, \cdots, \mathsf{ct}_k)$. Parse ρ_{sk} as ρ and i.
- Output $\mathsf{UD.Dec}(\rho, \mathsf{ct}_i)$.

Fig. 3: Collusion Resistant Unclonable Decryption.

We recall the unclonable decrytion scheme in [13] (see Figure 4).

There is one additional function Sim which takes a parameter n (message length) and outputs a junk ciphertext, which will be crucial for our anti-piracy proof. Intuitively, if one can distinguish from a honestly generated ciphertext with a simulated ciphertext, they can extract secrets for the underlying coset states.

The efficiency, correctness and CPA security of our CRUD scheme follows easily from those of UD. We are focusing on the proof of its anti-piracy in the next section.

3.3 Proof of Anti-Piracy

In this section, we prove that our construction satisfies anti-piracy. Although the proof requires to open up the structure of UD, this only happens for the last step: for arguing we can extract secrets for the underlying coset states using the properties of compute-and-compare obfuscation. Therefore, we will present the main idea of the proof here, leaving the proof of successful extraction (see Claim 5) in the full version.

$\mathsf{UD.Setup}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$:
- Sample ℓ random $(\lambda/2)$-dimensional subspaces $A_i \subseteq \mathbb{F}_2^\lambda$ for $i = 1, 2, \cdots, \ell$, where $\ell := \ell(\lambda)$ is a polynomial in λ.
- For each $i \in [\ell]$, choose two uniformly random vectors $s_i, s_i' \in \mathbb{F}_2^n$.
- Prepare the programs $\mathsf{iO}(A_i + s_i)$ and $\mathsf{iO}(A_i^\perp + s_i')$ (where we assume that the programs $A_i + s_i$ and $A_i^\perp + s_i'$ are padded to some appropriate length).
- Output $\mathsf{sk} = \{A_i, s_i, s_i'\}_{i\in[\ell]}, \mathsf{pk} = \{\mathsf{iO}(A_i + s_i), \mathsf{iO}(A_i^\perp + s_i')\}_{i\in[\ell]}$.

$\mathsf{UD.KeyGen}(\mathsf{sk}) \to \rho_{\mathsf{sk}}$: on input $\mathsf{sk} = \{A_i, s_i, s_i'\}_{i\in[\ell]}$, output the "quantum secret key" $\rho_{\mathsf{sk}} = \{|A_{i,s_i,s_i'}\rangle\}_{i\in[\ell]}$.

$\mathsf{UD.Enc}(\mathsf{pk}, m) \to \mathsf{ct}$: on input a public key $\mathsf{pk} = \{\mathsf{iO}(A_i + s_i), \mathsf{iO}(A_i^\perp + s_i')\}_{i\in[\ell]}$ and message m:
- Sample a uniformly random string $r \leftarrow \{0,1\}^\ell$.
- Let r_i be the i-th bit of r. Define $R_i^0 = \mathsf{iO}(A_i + s_i)$ and $R_i^1 = \mathsf{iO}(A_i^\perp + s_i')$. Let $\mathsf{P}_{\mathsf{m,r}}$ be the following program Figure 5.
- Let $\hat{\mathsf{P}}_{m,r} = \mathsf{iO}(\mathsf{P}_{\mathsf{m,r}})$. Output ciphertext $\mathsf{ct} = (\hat{\mathsf{P}}_{m,r}, r)$.

$\mathsf{UD.Dec}(\rho_{\mathsf{sk}}, \mathsf{ct}) \to m/\perp$: on input $\rho_{\mathsf{sk}} = \{|A_{i,s_i,s_i'}\rangle\}_{i\in[\ell]}$ and $\mathsf{ct} = (\hat{\mathsf{P}}_{m,r}, r)$:
- For each $i \in [\ell]$, if $r_i = 1$, apply $H^{\otimes n}$ to the i-th state $|A_{i,s_i,s_i'}\rangle$; if $r_i = 0$, leave the i-th state $|A_{i,s_i,s_i'}\rangle$ unchanged. Denote the resulting state by ρ_{sk}^*.
- Evaluate the program $\hat{\mathsf{P}}_{m,r}$ on input ρ_{sk}^* in superposition; measure the evaluation register and denote the outcome by m'. Output m'.
- Rewind by applying the operations in the first step again.

$\mathsf{UD.Sim}(n) \to \mathsf{ct}$: on input a message length n, $\mathsf{ct} \leftarrow \mathsf{iO}(\mathsf{Sim}(1^\lambda, P.\mathsf{param}))$ where Sim denotes the simulator for compute-and-compare obfuscator, $P.\mathsf{param}$ consists of all program parameters in $P_{m,r}$ as in $\mathsf{UD.Enc}$ for any m of length n.

Fig. 4: Unclonable Decryption in [13].

Theorem 6. *The construction in Section 3.2 has strong γ-anti-piracy for any inverse polynomial γ (as defined in Definition 7).*

Proof. We prove by contradiction. There exist inverse polynomials $\gamma(\cdot), \nu(\cdot), k(\cdot)$ and an adversary $\mathcal{A}$ such that for infinitely many $\lambda \in \mathbb{N}^+$, $\mathcal{A}$ outputs a pair of distinct messages (m_0, m_1) and a state σ over $k+1$ registers (which are $k+1$ decryptors) such that

$$\mathrm{Tr}\left[\left(\mathsf{TI}_{1/2+\gamma} \otimes \mathsf{TI}_{1/2+\gamma} \otimes \cdots \otimes \mathsf{TI}_{1/2+\gamma}\right)\sigma\right] \geq \nu. \tag{2}$$

Let σ^* be the leftover state (over the $k+1$ registers), conditioned on all $\mathsf{TI}_{1/2+\gamma}$ outputting 1. With Equation (2), we can get to σ^* with probability at least ν.

Next we will prove the theorem assuming we have perfect projective implementation (see below). Therefore, the resulting reduction is inefficient. At the end of the section, we will show the proof translates easily when we replace

On input $u = u_1||u_2||\cdots||u_\ell$ (where each $u_i \in \mathbb{F}_2^n$):

1. If for all $i \in [\ell]$, $R_i^{r_i}(u_i) = 1$:
   Output m
2. Else:
   Output $\perp$

Fig. 5: Program $P_{m,r}$

every projective implementation with its approximated and efficient version. This replacement will give us an efficient reduction and only incur a small loss.

Defining Probability Measurement PI. We start by defining the following measurements PI_i for each $i \in [k]$. PI_i stands for the projective implementation where the underlying ciphertext distribution is: the first i ciphertexts are "fake", without encoding any information about the plaintext; the rest are generated honestly. pk are $(\mathsf{pk}_1, \cdots, \mathsf{pk}_k)$ as defined in our construction Section 3.2; similarly for sk_i.

- Let $\mathcal{P}_i = (P_i, I - P_i)$ be the following POVM acting on a quantum decryptor:
  - Sample a uniform $b \leftarrow \{0, 1\}$ and random coins (which will be used to generated ciphertexts $\mathsf{ct}_1, \cdots, \mathsf{ct}_k$).
  - For each $j \in \{1, \cdots, i-1\}$, compute $\mathsf{ct}_j \leftarrow \mathsf{UD.Sim}(n)$ where n is the length of m_0 and m_1.
  - For each $j \in \{i, \cdots, k\}$, compute $\mathsf{ct}_j \leftarrow \mathsf{UD.Enc}(\mathsf{pk}_j, m_b)$.
  - Let $\mathsf{ct} = (\mathsf{ct}_1, \cdots, \mathsf{ct}_k)$.
  - Run the quantum decryptor on input ct. Check whether the outcome is m_b. If so, output 1; otherwise, output 0.
- Let PI_i be the projective implementation of $\mathcal{P}_i$.

It is easy to see that when a quantum decryptor is in the subspace defined by $\mathsf{TI}_{1/2+\gamma}$, applying PI_0 on the state will always produce a real number $\beta \geq 1/2 + \gamma$. This is a simple observation following Theorem 5: $\mathsf{TI}_{1/2+\gamma}$ is implemented by first applying PI_0 and comparing the outcome with $1/2 + \gamma$.

Let the outcome of applying PI_0 on the i-th quantum decryptor of σ^* be a random variable $b_{i,0}$. We have:

$$\Pr\left[\forall i \in [k+1], b_{i,0} \geq \frac{1}{2} + \gamma\right] = 1. \tag{3}$$

Repeated Probability Measure and Its Properties. We then define repeated projective implementation. For the first quantum decryptor $\sigma^*[1]$, we apply PI_0 to obtain a outcome $b_{1,0}$. Then we apply the next projective implementation PI_1 on the leftover state to obtain a outcome $b_{1,1}$. So on and so forth, until we stop after applying PI_k. The outcomes of all measurements are denoted by random variables $b_{1,0}, \cdots, b_{1,k}$.

Claim 1. There always exists $j \in [k]$ such that $b_{1,j-1} - b_{1,j} \geq \gamma/k$.

Proof. For any quantum decryptor, if we apply PI_k on it, the outcome will always be $1/2$. This is because the ciphertext in PI_k is always generated without any information about m_0 or m_1. Therefore, every decryptor's behavior is random guessing: $b_{1,k}$ is always $1/2$.

From Equation (3), we know that $b_{1,0} \geq 1/2 + \gamma$. By triangle inequality, the claim holds. □

We use a random variable j_1 for the first index such that $b_{1,j_1-1} - b_{1,j_1} \geq \gamma/k$.

We similarly define the above repeated projective implementation for every quantum decryptor $\sigma^*[i]$. Since the repeated measurement on the i-th decryptor commutes with the repeated measurement on the i'-th ($i' \neq i$) decryptor, we can safely assume they are done in any order. Let $(b_{i,0}, \cdots, b_{i,j}, \cdots, b_{i,k})$ be the outcome of the repeated projective implementation the i-th decryptor. Similarly, Claim 1 holds for every decryptor:

Claim 2. For every $i \in [k+1]$, there always exists $j \in [k]$ such that $b_{i,j-1} - b_{i,j} \geq \gamma/k$.

Let j_i be the first index such that $b_{i,j_i-1} - b_{i,j_i} \geq \gamma/k$. We next show that there always exist $x \neq y$ such that $j_x = j_y$.

Claim 3. $\Pr[\exists x \neq y, j_x = j_y] = 1$.

Proof. This is simply because for every $i \in [k+1]$, $j_i \in [k]$. The claim follows from the pigeonhole principle. □

Guessing x, y and j_x. We describe the first half of our reduction algorithm. The algorithm takes as input σ^* (postselecting on all $\mathsf{TI}_{1/2+\gamma}$ output 1, and aborting if it fails). In the second part of the reduction algorithm, it will extract a pair of

On input the $k+1$ quantum decryptors σ^*:

1. Randomly sample $1 \leq x < y \leq k+1$ and $j \in [k]$;
2. Apply repeated projective measurement PI_0 to PI_{j-1} to $\sigma^*[x]$. Let $b_{x,j-1}$ be the last outcome.
3. Apply repeated projective measurement PI_0 to PI_{j-1} to $\sigma^*[y]$. Let $b_{y,j-1}$ be the last outcome.
4. Output $(x, y, j, b_{x,j-1}, b_{y,j-1})$ and both the x-th and y-th decryptors, denoted by $\sigma^{**}[x, y]$.

Fig. 6: Reduction Algorithm Part 1

secrets for the same coset states from $\sigma^{**}[x, y]$.

We prove the following claim for the above algorithm.

Claim 4. With probability at least $1/(2k^3)$, the above procedure produces $(x, y, j, b_{x,j-1}, b_{y,j-1})$ and $\sigma^{**}[x, y]$ satisfy:

1. Applying $\mathsf{PI}_{j-1}^{\otimes 2}$ jointly on $\sigma^*[x, y]$ produces $b_{x,j-1}, b_{y,j-1}$ with probability 1.
2. Applying $\mathsf{PI}_j^{\otimes 2}$ jointly on $\sigma^*[x, y]$ produces $b_{x,j}, b_{y,j}$, such that:

$$\Pr\left[b_{x,j-1} - b_{x,j} \geq \frac{\gamma}{k} \wedge b_{y,j-1} - b_{y,j} \geq \frac{\gamma}{k}\right] \geq \frac{1}{2k^3}.$$

Proof for Claim 4. By Claim 2, there is always a pair of indices $x < y$ and an integer $j \in [k]$ such that $b_{x,j-1} - b_{x,j} \geq \frac{\gamma}{k}$ and $b_{y,j-1} - b_{y,j} \geq \frac{\gamma}{k}$ simultaneously. As a consequence, suppose that we guess x, y and j uniformly at random *after* applying the repeated projective implementation $\mathsf{PI}_0, \cdots, \mathsf{PI}_k$ on every quantum decryptor, then

$$\Pr\left[b_{x,j-1} - b_{x,j} \geq \frac{\gamma}{k} \wedge b_{y,j-1} - b_{y,j} \geq \frac{\gamma}{k}\right] \geq \frac{1}{\binom{k+1}{2} \cdot k} \geq \frac{1}{k^3}, \tag{4}$$

where the last inequality follows by $k \geq 1$.

Since the repeated projective implementations on disjoint quantum decryptors commute , the same probability can be achieved if we *only* apply the repeated measurements on the x-th and y-th decryptors, skipping the other $(k-1)$ ones (see Figure 7).

On input the $k+1$ quantum decryptors σ^*:

1. Randomly sample $1 \leq x < y \leq k+1$ and $j \in [k]$;
2. Apply repeated projective measurement PI_0 to PI_j to $\sigma^*[x]$. Let $b_{x,j-1}, b_{x,j}$ be the last two outcomes.
3. Apply repeated projective measurement PI_0 to PI_j to $\sigma^*[y]$. Let $b_{y,j-1}, b_{y,j}$ be the last two outcomes.
4. Output $(x, y, j, b_{x,j-1}, b_{x,j}, b_{y,j-1}, b_{y,j})$.

Fig. 7: Algorithm $\mathsf{RandomMeasure}(\sigma^*)$

We have

$$\Pr_{\mathsf{RandomMeasure}(\sigma^*)}\left[b_{x,j-1} - b_{x,j} \geq \frac{\gamma}{k} \wedge b_{y,j-1} - b_{y,j} \geq \frac{\gamma}{k}\right] \geq \frac{1}{k^3}. \tag{5}$$

Equation (4) and Equation (5) differ on how $b_{x,j-1}, b_{x,j}, b_{y,j-1}, b_{y,j}$ is sampled.

We can view our reduction algorithm (Figure 6) as the first step of RandomMeasure (Figure 7). More formally, RandomMeasure first runs the reduction algorithm to get $(x, y, j, b_{x,j-1}, b_{y,j-1})$ and $\sigma^{**}[x, y]$; it then applies PI_j on both registers to obtain $b_{x,j}$ and $b_{y,j}$.

If the claim we want to prove does not hold, then with probability $< 1/(2k^3)$, the outcome $(x, y, j, b_{x,j-1}, b_{y,j-1})$ and $\sigma^{**}[x, y]$ satisfy condition (2) in Claim 4. Therefore, the probability in Equation (5) is strictly smaller than $1/(2k^3)+1/(2k^3)$. This is a contradiction . □

Extracting Secrets from $\sigma^{**}[x, y]$. We describe the second half of our reduction algorithm. Given $(x, y, j, b_{x,j-1}, b_{y,j-1})$ and $\sigma^{**}[x, y]$ that satisfy both conditions in Claim 4, we can extract secrets for both coset states. This violates the strong computational monogamy-of-entanglement property of coset states, thus finishes the proof.

Recall the underlying ciphertext distribution of PI_{j-1} and PI_j:

1. The first $j-1$ ciphertexts $\mathsf{ct}_1, \cdots, \mathsf{ct}_{j-1}$ are generated by $\mathsf{Sim}(n)$.
2. The last $k-j$ ciphertexts $\mathsf{ct}_{j+1}, \cdots, \mathsf{ct}_k$ are generated honestly, using their corresponding public key.
3. The j-th ciphertext is either generated honestly using the j-th public key pk_j (in PI_{j-1}), or by $\mathsf{Sim}(n)$ (in PI_j). $\mathsf{pk}_j, \mathsf{sk}_j$ is generated by $\mathsf{UD.Setup}$ in Figure 4. Let the underlying cosets be $\{A_l + s_l, A_l^\perp + s'_l\}_{l=1}^{\ell}$:

$$\mathsf{pk}_j = \{\mathsf{iO}(A_l + s_l), \mathsf{iO}(A_l^\perp + s'_l)\}_{l\in[\ell]},$$
$$\mathsf{sk}_j = \{A_l, s_l, s'_l\}_{l\in[\ell]}.$$

The following claim says that if applying P_{j-1} or P_j on a quantum decryptor produce different values (with difference more than γ/k), then we can extract ℓ vectors $v_1, \cdots, v_\ell$: each v_l is uniformly in either $A_l + s_l$ or $A_l^\perp + s'_l$.

Claim 5. For any $k = \mathsf{poly}(\lambda)$, let $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{CRUD.Setup}(1^\lambda, k)$ where $\mathsf{sk} = (\mathsf{sk}_1, \cdots, \mathsf{sk}_k)$ and $\mathsf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_k)$. Let ρ_{sk} be the unclonable decryption key. For any $j \in [k]$, let PI_{j-1} and PI_j be defined at the beginning of the proof. Let $\mathsf{pk}_j = \{\mathsf{iO}(A_l + s_l), \mathsf{iO}(A_l^\perp + s'_l)\}_{l\in[\ell]}, \mathsf{sk}_j = \{A_l, s_l, s'_l\}_{l\in[\ell]}$.

If there exist inverse polynomials $\alpha_1(\cdot), \alpha_2(\cdot)$ and an quantum algorithm $\mathcal{B}$ that takes $(\rho_{\mathsf{sk}}, \mathsf{pk})$ outputs ρ such that with probability at least α_1, ρ satisfies the following:

1. There exists $b_{j-1} \in (0, 1]$, applying PI_{j-1} on ρ always produces b_{j-1}.
2. Let the outcome of applying PI_j on ρ be b_j. Then $\Pr[b_{j-1} - b_j > \gamma/k] > \alpha_2$.

Then there exists another inverse polynomial $\beta(\cdot)$ and an efficient quantum algorithm $\mathcal{C}$ that takes all the descriptions of $\{A_l\}_{l=1}^{\ell}$ (denoted by $\mathbf{A}$), ρ and ℓ random coins $r_1, \cdots, r_\ell \in \{0, 1\}$ such that:

$$\Pr_{\substack{\mathsf{sk},\mathsf{pk},\rho_{\mathsf{sk}},r\\ \rho\leftarrow\mathcal{B}(\rho_{\mathsf{sk}},\mathsf{pk})}}\left[\forall l \in [\ell], v_l \in \begin{cases} A_l + s_l & \text{if } r_l = 0 \\ A_l^\perp + s'_l & \text{if } r_l = 1 \end{cases}, (v_1, \cdots, v_\ell) \leftarrow \mathcal{C}(\mathbf{A}, \rho, r)\right] \geq \beta.$$

The proof of this is similar to the extraction technique in [13] using compute-and-compare obfuscation. We refer interested readers to the full version.

By setting $\alpha_1 = \alpha_2 := 1/(2k^3)$, $\mathcal{B}$ be the reduction algorithm in Figure 6 and $\rho := \sigma^{**}[x]$, we conclude that there exists another algorithm that takes $\sigma^{**}[x]$, random coins $r_1, \cdots, r_\ell$ and outputs $(v_1, \cdots, v_\ell)$ in the corresponding cosets (depending on each r_l).

Next, we show that after a successful extraction on the $\sigma^{**}[x]$, the other decryptor still satisfy the conditions (1) (2) for Claim 5. Therefore, we can extract another random set of vectors from the other decryptor, with non-negligible probability, even conditioned on a successful extraction on $\sigma^{**}[x]$.

Assume conditioned on a successful extraction on the $\sigma^{**}[x]$, the other decryptor becomes $\sigma'[y]$ and it does not satisfy the conditions in Claim 5.

First, applying PI_{j-1} on $\sigma'[y]$ always produces $b_{y,j-1}$. This is because the extraction on the $\sigma^{**}[x]$ register does not change the support of $\sigma'[y]$. Thus, condition (2) in Claim 5 can not hold. Let $\mathbf{E}_1$ denote a successful (E)xtraction on $\sigma^{**}[x]$ and $\mathbf{G}_2$ be a indicator that applying PI_j on $\sigma^{**}[y]$ to get $b_{y,j}$ and $b_{y,j} < b_{y,j-1} - \frac{\gamma}{k^3}$ (a big (G)ap). We know that in this case, $\Pr[\mathbf{E}_1 \wedge \mathbf{G}_2]$ is negligibly small.

However, this can not be true. We can imagine PI_j is implemented first. We know that $\Pr[\mathbf{G}_2]$ is non-negligible by the condition (2) in Claim 4. Conditioned on $\mathbf{G}_2$, let the x-th decryptor become $\sigma'[x]$. We know that $\sigma'[x]$ must satisfy both conditions in Claim 5. Otherwise, condition (2) in Claim 4 can not hold. Thus, $\Pr[\mathbf{G}_2|\mathbf{E}_1]$ must be non-negligible. This contradicts with the assumption that $\Pr[\mathbf{E}_1 \wedge \mathbf{G}_2]$ is negligibly small.

Thus, the reduction algorithm, with non-negligible probability, can extract $(v_1, \cdots, v_\ell)$ and $(v'_1, \cdots, v'_\ell)$ with respect to random $r_1, \cdots, r_\ell$ and $r'_1, \cdots, r'_\ell$. With probability at least $1 - 2^{-\ell}$, there exist $l \in [\ell]$ such that $r_l \neq r'_l$. Thus, v_l and v'_l will be two vectors in each of the cosets $A_l + s_l$ and $A'_l + s'_l$. By guessing this l, this breaks the computational strong monogamy-of-entanglement game (Theorem 3). □

References

1. Aaronson, S.: Limitations of quantum advice and one-way communication. Theory of Computing **1**(1), 1–28 (2005). https://doi.org/10.4086/toc.2005.v001a001
2. Aaronson, S.: Quantum copy-protection and quantum money. In: Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009. pp. 229–242. IEEE Computer Society (2009). https://doi.org/10.1109/CCC.2009.42
3. Aaronson, S., Christiano, P.: Quantum money from hidden subspaces. Theory of Computing **9**(9), 349–401 (2013). https://doi.org/10.4086/toc.2013.v009a009
4. Aaronson, S., Liu, J., Liu, Q., Zhandry, M., Zhang, R.: New approaches for quantum copy-protection. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 526–555. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_19

5. Ananth, P., Kaleoglu, F.: A note on copy-protection from random oracles (2022). https://doi.org/10.48550/ARXIV.2208.12884, https://arxiv.org/abs/2208.12884
6. Ananth, P., Kaleoglu, F., Li, X., Liu, Q., Zhandry, M.: On the feasibility of unclonable encryption, and more. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42st Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13507. Springer (2022)
7. Ananth, P., Placa, R.L.L.: Secure software leasing. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12697, pp. 501–530. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_17
8. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 1–18. Springer (2001). https://doi.org/10.1007/3-540-44647-8_1
9. Ben-David, S., Sattath, O.: Quantum tokens for digital signatures (2016)
10. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of International Conference on Computers, Systems & Signal Processing, Dec. 9-12, 1984, Bangalore, India. pp. 175–179 (1984)
11. Broadbent, A., Jeffery, S., Lord, S., Podder, S., Sundaram, A.: Secure software leasing without assumptions. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13042, pp. 90–120. Springer (2021). https://doi.org/10.1007/978-3-030-90459-3_4
12. Cohen, A., Holmgren, J., Nishimaki, R., Vaikuntanathan, V., Wichs, D.: Watermarking cryptographic capabilities. SIAM Journal on Computing **47**(6), 2157–2202 (2018)
13. Coladangelo, A., Liu, J., Liu, Q., Zhandry, M.: Hidden cosets and applications to unclonable cryptography. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 556–584. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_20
14. Coladangelo, A., Majenz, C., Poremba, A.: Quantum copy-protection of compute-and-compare programs in the quantum random oracle model (2020), https://arxiv.org/abs/2009.13865
15. Culf, E., Vidick, T.: A monogamy-of-entanglement game for subspace coset states (2021)
16. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. SIAM Journal on Computing **45**(3), 882–929 (2016). https://doi.org/10.1137/14095772X
17. Georgiou, M., Zhandry, M.: Unclonable decryption keys (2020), https://eprint.iacr.org/2020/877
18. Goyal, R., Kim, S., Manohar, N., Waters, B., Wu, D.J.: Watermarking public-key cryptographic primitives. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III. Lecture Notes in Com-

puter Science, vol. 11694, pp. 367–398. Springer (2019). https://doi.org/10.1007/978-3-030-26954-8_12
19. Kim, S., Wu, D.J.: Watermarking cryptographic functionalities from standard lattice assumptions. In: Annual International Cryptology Conference. pp. 503–536. Springer (2017)
20. Kim, S., Wu, D.J.: Watermarking prfs from lattices: stronger security via extractable prfs. In: Annual International Cryptology Conference. pp. 335–366. Springer (2019)
21. Kitagawa, F., Nishimaki, R., Yamakawa, T.: Secure software leasing from standard assumptions. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13042, pp. 31–61. Springer (2021). https://doi.org/10.1007/978-3-030-90459-3_2
22. Kretschmer, W.: Quantum pseudorandomness and classical complexity. In: Hsieh, M. (ed.) 16th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2021, July 5-8, 2021, Virtual Conference. LIPIcs, vol. 197, pp. 2:1–2:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.TQC.2021.2
23. Marriott, C., Watrous, J.: Quantum arthur–merlin games. computational complexity **14**(2), 122–152 (6 2005). https://doi.org/10.1007/s00037-005-0194-x
24. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010). https://doi.org/10.1017/CBO9780511976667
25. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: Deniable encryption, and more. SIAM Journal on Computing **50**(3), 857–908 (2021). https://doi.org/10.1137/15M1030108
26. Vidick, T., Zhang, T.: Classical proofs of quantum knowledge. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12697, pp. 630–660. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_22
27. Wiesner, S.: Conjugate coding. SIGACT News **15**(1), 78–88 (1983). https://doi.org/10.1145/1008908.1008920
28. Yang, R., Au, M.H., Lai, J., Xu, Q., Yu, Z.: Collusion resistant watermarking schemes for cryptographic functionalities. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11921, pp. 371–398. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_14
29. Yang, R., Au, M.H., Yu, Z., Xu, Q.: Collusion resistant watermarkable prfs from standard assumptions. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12170, pp. 590–620. Springer (2020). https://doi.org/10.1007/978-3-030-56784-2_20
30. Zhandry, M.: Schrödinger's pirate: How to trace a quantum decoder. In: Pass, R., Pietrzak, K. (eds.) Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12552, pp. 61–91. Springer (2020). https://doi.org/10.1007/978-3-030-64381-2_3