Hidden Cosets and Applications to Unclonable Cryptography

Andrea Coladangelo[1], Jiahui Liu[2], Qipeng Liu[3], and Mark Zhandry[4]

[1] University of California, Berkeley
[2] The University of Texas at Austin
[3] Princeton University
[4] Princeton University & NTT Research, USA

Abstract. In 2012, Aaronson and Christiano introduced the idea of *hidden subspace states* to build public-key quantum money [STOC '12]. Since then, this idea has been applied to realize several other cryptographic primitives which enjoy some form of unclonability.
In this work, we propose a generalization of hidden subspace states to hidden *coset* states. We study different unclonable properties of coset states and several applications:

- We show that, assuming indistinguishability obfuscation (iO), hidden coset states possess a certain *direct product hardness* property, which immediately implies a tokenized signature scheme in the plain model. Previously, a tokenized signature scheme was known only relative to an oracle, from a work of Ben-David and Sattath [QCrypt '17].
- Combining a tokenized signature scheme with extractable witness encryption, we give a construction of an unclonable decryption scheme in the plain model. The latter primitive was recently proposed by Georgiou and Zhandry [ePrint '20], who gave a construction relative to a classical oracle.
- We conjecture that coset states satisfy a certain natural (information-theoretic) monogamy-of-entanglement property. Assuming this conjecture is true, we remove the requirement for extractable witness encryption in our unclonable decryption construction, by relying instead on compute-and-compare obfuscation for the class of unpredictable distributions. As potential evidence in support of the monogamy conjecture, we prove a weaker version of this monogamy property, which we believe will still be of independent interest.
- Finally, we give the first construction of a copy-protection scheme for pseudorandom functions (PRFs) in the plain model. Our scheme is secure either assuming iO, OWF and extractable witness encryption, or assuming $\mathsf{iO}, \mathsf{OWF}$, compute-and-compare obfuscation for the class of unpredictable distributions, and the conjectured monogamy property mentioned above.

1 Introduction

The no-cloning principle of quantum mechanics asserts that quantum information cannot be generically copied. This principle has profound consequences

in quantum cryptography, as it puts a fundamental restriction on the possible strategies that a malicious party can implement. One of these consequences is that quantum information enables cryptographic tasks that are provably impossible to realize classically, the most famous example being information-theoretically secure key distribution [BB84].

Beyond this, the no-cloning principle opens up an exciting avenue to realize cryptographic tasks which enjoy some form of *unclonability*, e.g. quantum money [Wie83, AC12, FGH+12, Zha19a, Kan18], quantum tokens for digital signatures [BS16], copy-protection of programs [Aar09, ALL+20, CMP20], and more recently unclonable encryption [Got02, BL19] and decryption [GZ20].

In this work, we revisit the *hidden subspace* idea proposed by Aaronson and Christiano, which has been employed towards several of the applications above. We propose a generalization of this idea, which involves hidden *cosets* (affine subspaces), and we show applications of this to signature tokens, unclonable decryption and copy-protection.

Given a subspace $A \subseteq \mathbb{F}_2^n$, the corresponding *subspace state* is defined as a uniform superposition over all strings in the subspace A, i.e.

$$|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{x \in A} |x\rangle \ ,$$

The first property that makes this state useful is that applying a Hadamard on all qubits creates a uniform superposition over all strings in $A^\perp$, the orthogonal complement of A, i.e. $H^{\otimes n} |A\rangle = |A^\perp\rangle$.

The second property, which is crucial for constructing unclonable primitives with some form of verification, is the following. Given one copy of $|A\rangle$, where $A \subseteq F_2^n$ is uniformly random of dimension $n/2$, it is impossible to produce two copies of $|A\rangle$ except with negligible probability. As shown by [AC12], unclonability holds even when given quantum access to oracles for membership in A and $A^\perp$, as long as the number of queries is polynomially bounded. On the other hand, such membership oracles allow for verifying the state $|A\rangle$, leading to publicly-verifiable quantum money, where the verification procedure is the following:

- Given an alleged quantum money state $|\psi\rangle$, query the oracle for membership in A on input $|\psi\rangle$. Measure the outcome register, and verify that the outcome is 1.
- If so, apply $H^{\otimes n}$ to the query register, and query the oracle for membership in $A^\perp$. Measure the outcome register, and accept the money state if the outcome is 1.

It is not difficult to see that the unique state that passes this verification procedure is $|A\rangle$.

In order to obtain a quantum money scheme in the plain model (without oracles), Aaronson and Christiano suggest instantiating the oracles with some form of program obfuscation. This vision is realized subsequently in [Zha19a], where access to the oracles for subspace membership is replaced by a suitable

obfuscation of the membership programs, which can be built from indistinguishability obfuscation (iO). More precisely, Zhandry shows that, letting P_A and $P_{A^\perp}$ be programs that check membership in A and $A^\perp$ respectively, any computationally bounded adversary who receives a uniformly random subspace state $|A\rangle$ together with $\mathsf{iO}(P_A)$ and $\mathsf{iO}(P_{A^\perp})$ cannot produce two copies of $|A\rangle$ except with negligible probability.

The subspace state idea was later employed to obtain *quantum tokens* for digital signatures [BS16]. What these are is best explained by the (award-winning) infographic in [BS16] (see the ancillary arXiv files there). Concisely, a quantum signature token allows Alice to provide Bob with the ability to sign *one and only one* message in her name, where such signature can be publicly verified using Alice's public key. The construction of quantum tokens for digital signatures from [BS16] is the following.

- Alice samples a uniformly random subspace $A \subseteq \mathbb{F}_2^n$, which constitutes her secret key. A signature token is the state $|A\rangle$.
- Anyone in possession of a token $|A\rangle$ can sign message 0 by outputting a string $v \in A$ (this can be obtained by measuring $|A\rangle$ in the computational basis), and can sign message 1 by outputting a string $w \in A^\perp$ (this can be done by measuring $|A\rangle$ in the Hadamard basis).
- Signatures can be publicly verified assuming access to an oracle for subspace membership in A and in $A^\perp$ (such access can be thought of as Alice's public key).

In order to guarantee security of the scheme, i.e. that Bob cannot produce a valid signature for more than one message, Ben-David and Sattath prove the following strengthening of the original property proven by Aaronson and Christiano. Namely, they show that any query-bounded adversary with quantum access to oracles for membership in A and $A^\perp$ cannot produce, except with negligible probability, a pair (v, w) where $v \in A \setminus \{0\}$ and $w \in A^\perp \setminus \{0\}$. We refer to this property as a *direct product hardness* property.

The natural step to obtain a signature token scheme in the plain model is to instantiate the subspace membership oracles using iO, analogously to the quantum money application. However, unlike for the case of quantum money, here one runs into a technical barrier, which we expand upon in Section 2.1. Thus, a signature token scheme is not known in the plain model, and this has remained an open question since [BS16].

In general, a similar difficulty in obtaining schemes that are secure in the plain model as opposed to an oracle model seems prevalent in works about other unclonable primitives. For example, in the case of copy-protection of programs, we know that copy-protection of a large class of evasive programs, namely compute-and-compare programs, is possible with provable non-trivial security against fully malicious adversaries in the quantum random oracle model (QROM) [CMP20]. Other results achieving provable security in the plain model are secure only against a restricted class of adversaries [AP21, KNY20, BJL+21]. To make the contrast between plain model and oracle model even more stark, all unlearnable programs can be copy-protected assuming access to (highly structured) oracles

[ALL+20], but we know, on the other hand, that a copy-protection scheme for all unlearnable programs in the plain model does not exist (assuming Learning With Errors is hard for quantum computers) [AP21].

Likewise, for the recently proposed task of unclonable decryption, the only currently known scheme is secure only in a model with access to subspace membership oracles [GZ20].

1.1 Our Results

We propose a generalization of subspace states, which we call *coset* states. For $A \subseteq \mathbb{F}_2^n$, and $s, s' \in \mathbb{F}_2^n$, the corresponding coset state is:

$$|A_{s,s'}\rangle := \sum_{x \in A} (-1)^{\langle x, s' \rangle} |x + s\rangle \ ,$$

where here $\langle x, s' \rangle$ denotes the inner product of x and s'. In the computational basis, the quantum state is a superposition over all elements in the coset $A + s$, while, in the Hadamard basis, it is a superposition over all elements in $A^\perp + s'$. Let P_{A+s} and $P_{A^\perp + s'}$ be programs that check membership in the cosets $A + s$ and $A^\perp + s'$ respectively. To check if a state $|\psi\rangle$ is a coset state with respect to A, s, s', one can compute P_{A+s} in the computational basis, and check that the outcome is 1; then, apply $H^{\otimes n}$ followed by $P_{A^\perp + s'}$, and check that the outcome is 1.

Computational Direct Product Hardness. Our first technical result is establishing a *computational direct product hardness* property in the plain model, assuming post-quantum iO and one-way functions.

Theorem 1 (Informal). *Any quantum polynomial-time adversary who receives $|A_{s,s'}\rangle$ and programs $\mathsf{iO}(P_{A+s})$ and $\mathsf{iO}(P_{A^\perp + s'})$ for uniformly random $A \subseteq \mathbb{F}_2^n$, $s, s' \in \mathbb{F}_2^n$, cannot produce a pair $(v, w) \in (A + s) \times (A^\perp + s')$, except with negligible probability in n.*

As we mentioned earlier, this is in contrast to regular subspace states, for which a similar direct product hardness is currently not known in the plain model, but only in a model with access to subspace membership oracles.

We then apply this property to obtain the following primitives.

Signature Tokens. Our direct product hardness immediately implies a *signature token* scheme in the plain model (from post-quantum iO and one-way functions), thus resolving the main question left open in [BS16].

Theorem 2 (Informal). *Assuming post-quantum iO and one-way functions, there exists a signature token scheme.*

In this signature token scheme, the public verification key is the pair of programs $(\mathsf{iO}(P_{A+s}), \mathsf{iO}(P_{A^\perp + s'}))$, and a signature token is the coset state $|A_{s,s'}\rangle$. Producing signatures for both messages 0 and 1 is equivalent to finding elements in both $A + s$ and $A^\perp + s'$, which violates our computational direct product hardness property.

Unclonable Decryption. Unclonable decryption, also known as *single-decryptor encryption*, was introduced in [GZ20]. Informally, a single-decryptor encryption scheme is a (public-key) encryption scheme in which the secret key is a *quantum state*. The scheme satisfies a standard notion of security (in our case, CPA security), as well as the following additional security guarantee: no efficient quantum algorithm with one decryption key is able to produce two working decryption keys. We build a single-decryptor encryption scheme using a signature tokens scheme and extractable witness encryption in a black-box way. By leveraging our previous result about the existence of a signature token scheme in the plain model, we are able to prove security without the need for the structured oracles used in the original construction of [GZ20].

Theorem 3 (Informal). *Assuming post-quantum* iO*, one-way functions, and extractable witness encryption, there exists a public-key single-decryptor encryption scheme.*

Copy-protection of PRFs. The notion of a copy-protection scheme was introduced by Aaronson in [Aar09] and recently explored further in [AP21, CMP20, ALL$^+$20, BJL$^+$21].

In a copy-protection scheme, the vendor of a classical program wishes to provide a user the ability to run the program on any input, while ensuring that the functionality cannot be "pirated": informally, the adversary, given one copy of the program, cannot produce two programs that enable evaluating the program correctly.

Copy-protection is trivially impossible classically, since classical information can always be copied. This impossibility can be in principle circumvented if the classical program is encoded in a quantum state, due to the no-cloning principle. However, positive results have so far been limited. A copy-protection scheme [CMP20] is known for a class of evasive programs, known as compute-and-compare programs, with provable non-trivial security against fully malicious adversaries in the Quantum Random Oracle Model (QROM). Other schemes in the plain model are only secure against restricted classes of adversaries (which behave honestly in certain parts of the protocol) [AP21, KNY20, BJL$^+$21]. Copy-protection schemes for more general functionalities are known [ALL$^+$20], but these are only secure assuming very structured oracles (which depend on the functionality that is being copy-protected).

In this work, we present a copy-protection scheme for a family of pseudorandom functions (PRFs). In such a scheme, for any classical key K for the PRF, anyone in possession of a *quantum* key ρ_K is able to evaluate $PRF(K, x)$ on any input x.

The copy-protection property that our scheme satisfies is that given a quantum key ρ_K, no efficient algorithm can produce two (possibly entangled) keys such that these two keys allow for simultaneous correct evaluation on uniformly random inputs, with noticeable probability.

Similarly to the unclonable decryption scheme, our copy-protection scheme is secure assuming post-quantum iO, one-way functions, and extractable witness encryption.

Theorem 4 (Informal). *Assuming post-quantum* iO*, one-way functions, and extractable witness encryption, there exists a copy-protection scheme for a family of PRFs.*

We remark that our scheme requires a particular kind of PRFs, namely puncturing and extracting with small enough error. However, PRFs satisfying these properties can be built from just one-way functions.

The existence of extractable witness encryption is considered to be a very strong assumption. In particular, it was shown to be impossible in general (under a special-purpose obfuscation conjecture) [GGHW17]. However, we emphasize that no provably secure copy-protection schemes with standard malicious security in the plain model are known at all. Given the central role of PRFs in the construction of many other cryptographic primitives, we expect that our copy-protection scheme, and the techniques developed along the way, will play an important role as a building block to realize *unclonable* versions of other primitives.

To avoid the use of extractable witness encryption, we put forth a (information-theoretic) conjecture about a *monogamy of entanglement* property of coset states, which we discuss below. Assuming this conjecture is true, we show that both unclonable decryption and copy-protection of PRFs can be constructed *without* extractable witness encryption, by relying instead on compute-and-compare obfuscation [WZ17, GKW17] (more details on the latter can be found in Section 3.1).

Theorem 5 (Informal). *Assuming post-quantum* iO*, one-way functions, and obfuscation of compute-and-compare programs against unpredictable distributions, there exist: (i) a public-key single-decryptor encryption scheme, and (ii) a copy-protection scheme for a family of PRFs.*

As potential evidence in support of the monogamy-of-entanglement conjecture, we prove a weaker version of the monogamy of entanglement property, which we believe will still be of independent interest (more details on this are below).

Remark 1. While iO was recently constructed based on widely-believed computational assumptions [JLS20], the latter construction is not quantum resistant, and the situation is less clear quantumly. However, several works have proposed candidate post-quantum obfuscation schemes [BGMZ18, WW20, BDGM20], and based on these works iO seems plausible in the post-quantum setting as well.

Remark 2. Compute-and-compare obfuscation against unpredictable distributions is known to exist assuming LWE (or iO) and assuming the existence of Extremely Lossy Functions (ELFs) [Zha19c] [WZ17, GKW17]. Unfortunately, the

only known constructions of ELFs rely on hardness assumptions that are broken by quantum computers (exponential hardness of decisional Diffie-Hellman). To remedy this, we give a construction of computate-and-compare obfuscation against *sub-exponentially* unpredictable distributions, from plain LWE (see Theorem 6, and its proof in the full version). The latter weaker obfuscation is sufficient to prove security of our single-decryptor encryption scheme, and copy-protection scheme for PRFs, if one additionally assumes *sub-exponentially* secure iO and one-way functions.

Monogamy-of-Entanglement. As previously mentioned, we conjecture that coset states additionally satisfy a certain (information-theoretic) *monogamy of entanglement* property, similar to the one satisfied by BB84 states, which is studied extensively in [TFKW13]. Unlike the monogamy property of BB84 states, the monogamy property we put forth is well-suited for applications with public verification, in a sense made more precise below.

This monogamy property states that Alice, Bob and Charlie cannot cooperatively win the following game with a challenger, except with negligible probability. The challenger first prepares a uniformly random coset state $|A_{s,s'}\rangle$ and gives the state to Alice. Alice outputs two (possibly entangled) quantum states and sends them to Bob and Charlie respectively. Finally, Bob and Charlie both get the description of the subspace A. The game is won if Bob outputs a vector in $A + s$ and Charlie outputs a vector in $A^{\perp} + s'$.

Notice that if Alice were told A before she had to send the quantum states to Bob and Charlie, then she could recover s and s' (efficiently) given $|A_{s,s'}\rangle$. Crucially, A is only revealed to Bob and Charlie *after* Alice has sent them the quantum states (analogously to the usual monogamy-of-entanglement game based on BB84 states, where θ is only revealed to Bob and Charlie after they receive their states from Alice.).

We note that the hardness of this game is an *information-theoretic* conjecture. As such, there is hope that it can be proven unconditionally.

Under this conjecture, we show that the problem remains hard (computationally) even if Alice additionally receives the programs $\mathsf{iO}(P_{A+s})$ and $\mathsf{iO}(P_{A^{\perp}+s'})$. Based on this result, we then obtain unclonable decryption and copy-protection of PRFs from post-quantum iO and one-way functions, and compute-and-compare obfuscation against unpredictable distributions. We thus remove the need for extractable witness encryption (more details on this are provided in the technical overview, Section 2.1).

As evidence in support of our conjecture, we prove a weaker information-theoretic monogamy property, namely that Alice, Bob and Charlie cannot win at a monogamy game that is identical to the one described above, except that at the last step, Bob and Charlie are each required to return a pair in $(A+s)\times(A^{\perp}+s')$, instead of a single element each. Since coset states have more algebraic structure than BB84 states, a more refined analysis is required to prove this (weaker) property compared to that of [TFKW13]. We again extend this monogamy result to the case where Alice receives programs $\mathsf{iO}(P_{A+s})$ and $\mathsf{iO}(P_{A^{\perp}+s'})$.

We emphasize that our monogamy result for coset states differs from the similar monogamy result for BB84 states in one crucial way: the result still holds when Alice receives programs that allow her to verify the correctness of her state (namely $\mathsf{iO}(P_{A+s})$ and $\mathsf{iO}(P_{A^\perp+s'})$). This is not the case for the BB84 monogamy result. In fact, Lutomirski [Lut10] showed that an adversary who is given $|x^\theta\rangle$ and a public verification oracle that outputs 1 if the input state is correct and 0 otherwise, can efficiently copy the state $|x^\theta\rangle$. At the core of this difference is the fact that coset states are highly entangled, whereas strings of BB84 states have no entanglement at all.

For this reason, we believe that the monogamy property of coset states may be of independent interest, and may find application in contexts where public verification of states is important.

2 Technical Overview

2.1 Computational Direct Product Hardness for Coset States

Our first technical contribution is to establish a *computational* direct product hardness property for coset states. In this section, we aim to give some intuition for the barrier to proving such a property for regular subspace states, and why resorting to coset states helps.

We establish the following: a computationally bounded adversary who receives $|A_{s,s'}\rangle$ and programs $\mathsf{iO}(P_{A+s})$ and $\mathsf{iO}(P_{A^\perp+s'})$ for uniformly random A, s, s', cannot produce a pair (v, w), where $v \in A + s$ and $w \in A^\perp + s'$, except with negligible probability.

The first version of this direct product hardness property involved regular subspace states, and was *information-theoretic*. It was proven by Ben-David and Sattath [BS16], and it established the following: given a uniformly random subspace state $|A\rangle$, where $A \subseteq \mathbb{F}_2^n$ has dimension $n/2$, no adversary can produce a pair of vectors v, w such that $v \in A$ and $w \in A^\perp$ respectively, even with access to oracles for membership in A and in $A^\perp$.

The first successful instantiation of the membership oracles in the plain model is due to Zhandry, in the context of public-key quantum money [Zha19a]. Zhandry showed that replacing the membership oracles with indistinguishability obfuscations of the membership programs P_A and $P_{A^\perp}$ is sufficient to prevent an adversary from copying the subspace state, and thus is sufficient for public-key quantum money. In what follows, we provide some intuition as to how one proves this "computational no-cloning" property, and why the same proof idea does not extend naturally to the direct product hardness property for regular subspace states.

In [Zha19a], Zhandry shows that iO realizes what he refers to as a *subspace-hiding obfuscator*. A subspace hiding obfuscator shO has the property that any computationally bounded adversary who chooses a subspace A cannot distinguish between $\mathsf{shO}(P_A)$ and $\mathsf{shO}(P_B)$ for a uniformly random superspace B of A (of not too large dimension). In turn, a subspace hiding obfuscator can then

be used to show that an adversary who receives $|A\rangle$, $\mathsf{shO}(P_A)$ and $\mathsf{shO}(P_{A^\perp})$, for a uniformly random A, cannot produce two copies of $|A\rangle$. This is done in the following way. For the rest of the section, we assume that $A \subseteq \mathbb{F}_2^n$ has dimension $n/2$.

- Replace $\mathsf{shO}(P_A)$ with $\mathsf{shO}(P_B)$ for a uniformly random superspace B of A, where $\dim(B) = \frac{3}{4}n$. Replace $\mathsf{shO}(P_{A^\perp})$ with $\mathsf{shO}(P_C)$ for a uniformly random superspace C of $A^\perp$, where $\dim(C) = \frac{3}{4}n$.
- Argue that the task of copying a subspace state $|A\rangle$, for a uniformly random subspace $C^\perp \subseteq A \subseteq B$ (even knowing B and C directly) is just as hard as the task of copying a uniformly random subspace state of dimension $|A'\rangle \subseteq \mathbb{F}_2^{n/2}$ where $\dim(A') = \frac{n}{4}$. The intuition for this is that knowing $C^\perp$ fixes $\frac{n}{4}$ dimensions out of the $\frac{n}{2}$ original dimensions of A. Then, you can think of the first copying task as equivalent to the second up to a change of basis. Such reduction completely removes the adversary's knowledge about the membership programs.
- The latter task is of course hard (it would even be hard with access to membership oracles for A' and $A'^\perp$).

One can try to apply the same idea to prove a *computational direct product hardness property* for subspace states, where the task is no longer to copy $|A\rangle$, but rather we wish to show that a bounded adversary receiving $|A\rangle$ and programs $\mathsf{iO}(P_A)$ and $\mathsf{iO}(P_{A^\perp})$, for uniformly random A, cannot produce a pair (v, w), where $v \in A$ and $w \in A^\perp$. Applying the same replacements as above using shO allows us to reduce this task to the task of finding a pair of vectors in $A \times A^\perp$ given $|A\rangle$,B, C, such that $C^\perp \subseteq A \subseteq B$. Unfortunately, unlike in the case of copying, this task is easy, because any pair of vectors in $C^\perp \times B^\perp$ also belongs to $A \times A^\perp$. This is the technical hurdle that ones runs into when trying to apply the proof idea from [Zha19a] to obtain a computational direct hardness property for subspace states.

Our first result is that we overcome this hurdle by using coset states. In the case of cosets, the natural analog of the argument above results in a replacement of the program that checks membership in $A + s$ with a program that checks membership in $B + s$. Similarly, we replace $A^\perp + s'$ with $C + s'$. The crucial observation is that, since $B + s = B + s + t$ for any $t \in B$, the programs P_{B+s} and P_{B+s+t} are functionally equivalent. So, an adversary who receives $\mathsf{iO}(P_{B+s})$ cannot distinguish this from $\mathsf{iO}(P_{B+s+t})$ for any t. We can thus argue that t functions as a randomizing mask that prevents the adversary from guessing s and finding a vector in $A + s$.

Signature Tokens. The computational direct product hardness immediately gives a signature token scheme in the plain model:

- Alice samples a key (A, s, s') uniformly at random. This constitutes her secret key. The verification key is $(\mathsf{iO}(P_{A+s}), \mathsf{iO}(P_{A^\perp+s'}))$. A signature token is $|A_{s,s'}\rangle$.

– Anyone in possession of a token can sign message 0 by outputting a string $v \in A+s$ (this can be obtained by measuring the token in the computational basis), and can sign message 1 by outputting a string $w \in A^\perp + s'$ (this can be done by measuring the token in the Hadamard basis).
– Signatures can be publicly verified using Alice's public key.

If an algorithm produces both signatures for messages 0 and 1, it finds vectors $v \in A+s$ and $w \in A^\perp+s'$, which violates computational direct product hardness.

2.2 Unclonable Decryption

Our second result is an *unclonable decryption* scheme (also known as a *single-decryptor encryption* scheme [GZ20] - we will use the two terms interchangeably in the rest of the paper) from black-box use of a signature token scheme and extractable witness encryption. This construction removes the need for structured oracles, as used in the construction of [GZ20].

Additionally, we show that, assuming the conjectured monogamy property described in Section 1.1, we obtain an unclonable decryption scheme from just iO and post-quantum one-way functions, where iO is used to construct obfuscators for both subspace-membership programs and compute-and-compare programs [GKW17, WZ17].

In this overview, we focus on the construction from the monogamy property, as we think it is conceptually more interesting.

Recall that a single-decryptor encryption scheme is a public-key encryption scheme in which the secret key is a quantum state. On top of the usual encryption security notions, one can define "single-decryptor" security: this requires that it is not possible for an adversary who is given the secret key to produce two (possibly entangled) decryption keys, which both enable simultaneous successful decryption of ciphertexts. A simplified version of our single-decryptor encryption scheme is the following. Let $n \in \mathbb{N}$.

– The key generation procedure samples uniformly at random $A \subseteq \mathbb{F}_2^n$, with $\dim(A) = \frac{n}{2}$ and $s, s' \in \mathbb{F}_2^n$ uniformly at random. The public key is the pair $(\mathsf{iO}(P_{A+s}), \mathsf{iO}(P_{A^\perp+s'}))$. The (quantum) secret key is the coset state $|A_{s,s'}\rangle$.
– To encrypt a message m, sample uniformly $r \leftarrow \{0,1\}$, and set $R = \mathsf{iO}(P_{A+s})$ if $r = 0$ and $R = \mathsf{iO}(P_{A^\perp+s'})$ if $r = 1$. Then, let C be the following program:

  C: on input v, output the message m if $R(v) = 1$ and otherwise output $\perp$.

  The ciphertext is then $(r, \mathsf{iO}(C))$.
– To decrypt a ciphertext $(r, \mathsf{iO}(C))$ with the quantum key $|A_{s,s'}\rangle$, one simply runs the program $\mathsf{iO}(C)$ coherently on input $|A_{s,s'}\rangle$ if $r = 0$, and on $H^{\otimes n}|A_{s,s'}\rangle$ if $r = 1$.

In the full scheme, we actually amplify security by sampling $r \leftarrow \{0,1\}^\lambda$, and having λ coset states, but we choose to keep the presentation in this section as simple as possible.

The high level idea for single-decryptor security is the following. Assume for the moment that iO were an ideal obfuscator (we will argue after this that iO is good enough). Consider a pirate who receives a secret key, produces two copies of it, and gives one to Bob and the other to Charlie. Suppose both Bob and Charlie can decrypt ciphertexts $(r, \mathsf{iO}(C))$ correctly with probability close to 1, over the randomness in the choice of r (which is crucially chosen only after Bob and Charlie have received their copies). Then, there must be some efficient quantum algorithm, which uses Bob's (resp. Charlie's) auxiliary quantum information (whatever state he has received from the pirate), and is able to output a vector in $A + s$. This is because in the case of $r = 0$, the program C outputs the plaintext message m exclusively on inputs $v \in A + s$. Similarly, there must be an algorithm that outputs a vector in $A^{\perp} + s'$ starting from Bob's (resp. Charlie's) auxiliary quantum information. Notice that this doesn't imply that Bob can *simultaneously* output a pair in $(A + s) \times (A^{\perp} + s')$, because explicitly recovering a vector in one coset might destroy the auxiliary quantum information preventing recovery of a vector in the other (and this very fact is of course crucial to the direct product hardness). Hence, in order to argue that it is not possible for both Bob and Charlie to be decrypting with probability close to 1, we have to use the fact that Bob and Charlie have separate auxiliary quantum information, and that each of them can recover vectors in $A + s$ or $A^{\perp} + s'$, which means that this can be done simultaneously, now violating the direct product hardness property.

The crux of the security proof is establishing that iO is a good enough obfuscator to enable this argument to go through.

To this end, we first notice that there is an alternative way of computing membership in $A + s$, which is functionally equivalent to the program C defined above.

Let $\mathsf{Can}_A(s)$ be a function that computes the lexicographically smallest vector in $A + s$ (think of this as a representative of the coset). It is not hard to see that a vector t is in $A + s$ if and only if $\mathsf{Can}_A(t) = \mathsf{Can}_A(s)$. Also Can_A is efficiently computable given A. Therefore, a functionally equivalent program to C, in the case that $r = 0$, is:

$\widetilde{C}$: on input v, output m if $\mathsf{Can}_A(v) = \mathsf{Can}_A(s)$, otherwise output $\perp$.

By the security of iO, an adversary can't distinguish $\mathsf{iO}(C)$ from $\mathsf{iO}(\widetilde{C})$.

The key insight is that now the program $\widetilde{C}$ is a *compute-and-compare* program [GKW17, WZ17]. The latter is a program described by three parameters: an efficiently computable function f, a target y and an output z. The program outputs z on input x if $f(x) = y$, and otherwise outputs $\perp$. In our case, $f = \mathsf{Can}_A$, $y = \mathsf{Can}_A(s)$, and $z = m$. Goyal et al. [GKW17] and Wichs et al. [WZ17] show that, assuming LWE or assuming iO and certain PRGs, a compute-and-compare program can be obfuscated provided y is (computationally) unpredictable given the function f and the auxiliary information. More precisely, the obfuscation guarantee is that the obfuscated compute-and-compare program is indistinguishable from the obfuscation of a (simulated) program that outputs zero on every input (notice, as a sanity check, that if y is unpredictable given f, then the compute-and-compare program must output zero almost everywhere as well).

We will provide more discussion on compute-and-compare obfuscation for unpredictable distributions in the presence of quantum auxiliary input in Section 3.1 and the full version.

- By the security of iO, we can replace the ciphertext $(0, \mathsf{iO}(C))$, with the ciphertext $(0, \mathsf{iO}(\mathsf{CC.Obf}(\widetilde{C})))$ where $\mathsf{CC.Obf}$ is an obfuscator for compute-and-compare programs (this is because C has the same functionality as $\mathsf{CC.Obf}(\widetilde{C})$).
- By the security of $\mathsf{CC.Obf}$, we can replace the latter with $(0, \mathsf{iO}(\mathsf{CC.Obf}(Z)))$, where Z is the zero program. It is clearly impossible to decrypt from the latter, since no information about the message is present.

Thus, assuming iO cannot be broken, a Bob that is able to decrypt implies an adversary breaking the compute-and-compare obfuscation. This implies that there must be an efficient algorithm that can predict $y = \mathsf{Can}_A(s)$ with non-negligible probability given the function Can_A and the auxiliary information received by Bob. Similarly for Charlie.

Therefore, if Bob and Charlie, with their own quantum auxiliary information, can both independently decrypt respectively $(0, \mathsf{iO}(C))$ and $(1, \mathsf{iO}(C'))$ with high probability (where here C and C' only differ in that the former releases the encrypted message on input a vector in $A + s$, and C' on input a vector in $A^\perp + s'$), then there exist efficient quantum algorithms for Bob and Charlie that take as input the descriptions of $\mathsf{Can}_A(\cdot)$ and $\mathsf{Can}_{A^\perp}(\cdot)$ respectively (or of the subspace A), and their respective auxiliary information, and recover $\mathsf{Can}_A(s)$ and $\mathsf{Can}_{A^\perp}(s')$ respectively with non-negligible probability. Since $\mathsf{Can}_A(s) \in A + s$ and $\mathsf{Can}_{A^\perp}(s') \in A^\perp + s'$, this violates the strong monogamy property of coset states described in Section 1.1.

Recall that this states that Alice, Bob and Charlie cannot cooperatively win the following game with a challenger, except with negligible probability. The challenger first prepares a uniformly random coset state $|A_{s,s'}\rangle$ and gives the state to Alice. Alice outputs two (possibly entangled) quantum states and sends them to Bob and Charlie respectively. Finally, Bob and Charlie both get the description of the subspace A. The game is won if Bob outputs a vector in $A + s$ and Charlie outputs a vector in $A^\perp + s'$. Crucially, in this monogamy property, Bob and Charlie will both receive the description of the subspace A in the final stage, yet it is still not possible for both of them to be simultaneously successful.

What allows to deduce the existence of efficient extracting algorithms is the fact that the obfuscation of compute-and-compare programs from [GKW17, WZ17] holds provided y is computationally unpredictable given f (and the auxiliary information). Thus, an algorithm that breaks the obfuscation property implies an efficient algorithm that outputs y (with noticeable probability) given f (and the auxiliary information).

In our other construction from signature tokens and extractable witness encryption, one can directly reduce unclonable decryption security to direct product hardness. We do not discuss the details of this construction here, instead we refer the reader to the full version.

2.3 Copy-Protecting PRFs

Our last contribution is the construction of copy-protected PRFs assuming post-quantum iO, one-way functions and the monogamy property we discussed in the previous section. Alternatively just as for unclonable decryption, we can do away with the monogamy property by assuming extractable witness encryption.

A copy-protectable PRF is a regular PRF $F : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^{m'}$, except that it is augmented with a *quantum key* generation procedure, which we refer to as $\mathsf{QKeyGen}$. This takes as input the classical PRF key K and outputs a quantum state ρ_K. The state ρ_K allows to efficiently compute $F(K,x)$ on any input x (where correctness holds with overwhelming probability). Beyond the standard PRF security, the copy-protected PRF satisfies the following additional security guarantee: any computationally bounded adversary that receives ρ_K cannot process ρ_K into two states, such that each state enables efficient evaluation of $F(K,\cdot)$ on uniformly random inputs.

A simplified version of our construction has the following structure. For the rest of the section, we take all subspaces to be of $\mathbb{F}_2^n$ with dimension $n/2$.

- The quantum key generation procedure $\mathsf{QKeyGen}$ takes as input a classical PRF key K and outputs a quantum key. The latter consists of a number of uniformly sampled coset states $|(A_i)_{s_i,s'_i}\rangle$, for $i \in [\lambda]$, together with a (classical) *obfuscation* of the classical program P that operates as follows. P takes an input of the form $(x, v_1, \ldots, v_\lambda)$; checks that each vector v_i belongs to the correct coset ($A_i + s_i$ if $x_i = 0$, and $A_i^\perp + s'_i$ if $x_i = 1$); if so, outputs the value $F(K,x)$, otherwise outputs $\perp$.
- A party in possession of the quantum key can evaluate the PRF on input x as follows: for each i such that $x_i = 1$, apply $H^{\otimes n}$ to $|(A_i)_{s_i,s'_i}\rangle$. Measure each resulting coset state in the standard basis to obtain vectors $v_1, \ldots, v_\lambda$. Run the obfuscated program on input $(x, v_1, \ldots, v_\lambda)$.

Notice that the program has the classical PRF key K hardcoded, as well as the values A_i, s_i, s'_i, so giving the program in the clear to the adversary would be completely insecure: once the adversary knows the key K, he can trivially copy the functionality $F(K,\cdot)$; and even if the key K is hidden by the obfuscation, but the A_i, s_i, s'_i are known, a copy of the (classical) obfuscated program P, together with the A_i, s_i, s'_i is sufficient to evaluate $F(K,\cdot)$ on any input.

So, the hope is that an appropriate obfuscation will be sufficient to hide all of these parameters. If this is the case, then the intuition for why the scheme is secure is that in order for two parties to simultaneously evaluate correctly on uniformly random inputs, each party should be able to produce a vector in $A_i + s$ or in $A_i^\perp + s'_i$. If the two parties accomplish this separately, then this implies that it is possible to simultaneously extract a vector in $A_i + s_i$ and one in $A_i^\perp + s'_i$, which should not be possible. [1]

[1] Again, we point out that we could not draw this conclusion if only a single party were able to do the following two things, each with non-negligible probability: produce a vector in $A + s_i$ and produce a vector in $A^\perp + s'_i$. This is because in a quantum

We will use iO to obfuscate the program P. In the next part of this overview, we will discuss how we are able to deal with the fact that the PRF key K and the cosets are hardcoded in the program P. First of all, we describe a bit more precisely the copy-protection security that we wish to achieve. The latter is captured by the following security game between a challenger and an adversary (A, B, C):

- The challenger samples a uniformly random PRF key K and runs $\mathsf{QKeyGen}$ to generate ρ_K. Sends ρ_K to A.
- A sends quantum registers to two spatially separated parties B and C.
- The challenger samples uniformly random inputs x, x' to $F(K, \cdot)$. Sends x to B and x' to C.
- B and C return y and y' respectively to the challenger.

(A, B, C) wins if $y = F(K, x)$ and $y' = F(K, x')$.

Since the obfuscation we are using is not VBB, but only iO, there are two potential issues with security. B and C could be returning correct answers not because they are able to produce vectors in the appropriate cosets, but because:

(i) $\mathsf{iO}(P)$ leaks information about the PRF key K.
(ii) $\mathsf{iO}(P)$ leaks information about the cosets.

We handle issue (i) via a delicate "puncturing" argument [SW14]. At a high level, a puncturable PRF F is a PRF augmented with a procedure that takes a key K and an input value x, and produces a "punctured" key $K \setminus \{x\}$, which enables evaluation of $F(K, \cdot)$ at any point other than x. The security guarantee is that a computationally bounded adversary possessing the punctured key $K \setminus \{x\}$ cannot distinguish between $F(K, x)$ and a uniformly random value (more generally, one can puncture the key at any polynomially sized set of points). Puncturable PRFs can be obtained from OWFs using the [GGM86] construction [BW13].

By puncturing K precisely at the challenge inputs x and x', one is able to hardcode a punctured PRF key $K \setminus \{x, x'\}$ in the program P, instead of K, and setting the output of program P at x to uniformly random z and z', instead of to $F(K, x)$ and $F(K, x')$ respectively. The full argument is technical, and relies on the "hidden trigger" technique introduced in [SW14], which allows the "puncturing" technique to work even when the program P is generated before x and x' are sampled.

Once we have replaced the outputs of the program P on the challenge inputs x, x' with uniformly random outputs z, z', we can handle issue (ii) in a similar way to the case of unclonable decryption in the previous section.

By the security of iO, we can replace the behaviour of program P at x by a suitable functionally equivalent compute-and-compare program that checks membership in the appropriate cosets. We then replace this by an obfuscation of the same compute-and-compare program, and finally by an obfuscation of the

world, being able to perform two tasks with good probability, does not imply being able to perform both tasks simultaneously. So it is crucial that both parties are able to separately recover the vectors.

zero program. We can then perform a similar reduction as in the previous section from an adversary breaking copy-protection security (and thus the security of the compute-and-compare obfuscation) to an adversary breaking the monogamy of entanglement game described in the previous section.

As in the previous section, we can replace the reliance on the conjectured monogamy property by extractable witness encryption. In fact, formally, we directly reduce the security of our copy-protected PRFs to the security of our unclonable decryption scheme.

3 Preliminaries

In this paper, we use λ to denote security parameters. We denote a function belonging to the class of polynomial functions by $\mathsf{poly}(\cdot)$. We say a function $f(\cdot) : \mathbb{N} \to \mathbb{R}^+$ is negligible if for all constant $c > 0$, $f(n) < \frac{1}{n^c}$ for all large enough n. We use $\mathsf{negl}(\cdot)$ to denote a negligible function. We say a function $f(\cdot) : \mathbb{N} \to \mathbb{R}^+$ is sub-exponential if there exists a constant $0 < c \leq 1$, such that $f(n) \geq 2^{n^c}$ for all large enough n. We use $\mathsf{subexp}(\cdot)$ to denote a sub-exponential function. When we refer to a probabilistic algorithm $\mathcal{A}$, sometimes we need to specify the randomness r used by $\mathcal{A}$ when running on some input x. We write this as $\mathcal{A}(x; r)$. For a finite set S, we use $x \leftarrow S$ to denote uniform sampling of x from the set S. We denote $[n] = \{1, 2, \cdots, n\}$. A binary string $x \in \{0,1\}^\ell$ is represented as $x_1 x_2 \cdots x_\ell$. For two strings x, y, $x||y$ is the concatenation of x and y. We refer to a probabilistic polynomial-time and quantum polynomial time algorithm as PPT and QPT respectively.

For the rest of this paper, we will assume that all the classical cryptographic primitives used are post-quantum secure, and we sometimes omit this description for simplicity, except in formal definitions and theorems.

We omit the definitions of extracting, puncturable PRFs, injective punturable PRFs, indistinguishability obfuscation (iO), and subspace hiding obfuscation (shO). We refer the reader to the full version for these.

3.1 Compute-and-Compare Obfuscation

Definition 1 (Compute-and-Compare Program). *Given a function* $f : \{0,1\}^{\ell_{\mathsf{in}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ *along with a target value* $y \in \{0,1\}^{\ell_{\mathsf{out}}}$ *and a message* $z \in \{0,1\}^{\ell_{\mathsf{msg}}}$*, we define the compute-and-compare program:*

$$\mathsf{CC}[f, y, z](x) = \begin{cases} z & \text{if } f(x) = y \\ \bot & \text{otherwise} \end{cases}$$

We define the following class of *unpredictable distributions* over pairs of the form $(\mathsf{CC}[f, y, z], \mathsf{aux})$, where aux is auxiliary quantum information. These distributions are such that y is computationally unpredictable given f and aux.

Definition 2 (Unpredictable/Sub-exponentially Unpredictable Distributions). *We say that a family of distributions $D = \{D_\lambda\}$ where D_λ is a distribution over pairs of the form $(\mathsf{CC}[f, y, z], \mathsf{aux})$ where aux is a quantum state, belongs to the class of* unpredictable distributions *if the following holds. There exists a negligible function negl, for all QPT algorithms $\mathcal{A}$,*

$$\Pr_{(\mathsf{CC}[f,y,z],\mathsf{aux})\leftarrow D_\lambda}\left[A(1^\lambda, f, \mathsf{aux}) = y\right] \leq \mathsf{negl}(\lambda).$$

If there exists a sub-exponential function subexp such that, for all QPT algorithms $\mathcal{A}$, the above probability is at most $1/\mathsf{subexp}(\lambda)$, we say it belongs to the class of sub-exponentially unpredictable distributions.

Definition 3 (Compute-and-Compare Obfuscation). *A PPT algorithm $\mathsf{CC.Obf}$ is an obfuscator for the class of unpredictable distributions (or sub-exponentially unpredictable distributions) if for any family of distributions $D = \{D_\lambda\}$ belonging to the class, the following holds:*

- *Functionality Preserving: there exists a negligible function negl such that for all λ, every program P in the support of D_λ,*

$$\Pr[\forall x,\ \widetilde{P}(x) = P(x),\ \widetilde{P} \leftarrow \mathsf{CC.Obf}(1^\lambda, P)] \geq 1 - \mathsf{negl}(\lambda)$$

- *Distributional Indistinguishability: there exists an efficient simulator Sim such that:*

$$(\mathsf{CC.Obf}(1^\lambda, P), \mathsf{aux}) \approx_c (\mathsf{Sim}(1^\lambda, P.\mathsf{param}), \mathsf{aux})$$

  where $(P, \mathsf{aux}) \leftarrow D_\lambda$ and $P.\mathsf{param}$ consists the parameters of the circuit, including input size, output size, circuit size and etc.

Combining the results of [WZ17, GKW17] with those of [Zha19c], one obtains the following theorem. We refer to the full version for proofs and discussions.

Theorem 6. *Assuming the existence of post-quantum iO and the quantum hardness of LWE, there exist obfuscators for sub-exponentially unpredictable distributions, as in Definition 3.*

4 Coset States

This section is organized as follows. In Section 4.1, we introduce coset states. In Section 4.2, we show that coset states satisfy both an information-theoretic and a computational *direct product hardness* property. The latter immediately yields a signature token scheme in the plain model assuming iO, (this is described in Section 5). In Section 4.3 we show that coset states satisfy both an information-theoretic *monogamy of entanglement* property (analogous to that satisfied by BB84 states [TFKW13]), and a computational monogamy of entanglement property. The latter is used to obtain an unclonable decryption scheme

from iO and extractable witness encryption (which will be presented in the full version). In Section 4.4, we describe a *strong version* of the monogamy property, which we conjecture to be true. The latter is used in Section 6.2 to obtain an unclonable decryption scheme which does not assume extractable witness encryption.

4.1 Definitions

In this subsection, we provide the basic definitions and properties of coset states.

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

Definition 4 (Subspace States). *For any subspace $A \subseteq \mathbb{F}_2^n$, the subspace state $|A\rangle$ is defined as*

$$|A\rangle = \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle \ .$$

Note that given A, the subspace state $|A\rangle$ can be constructed efficiently.

Definition 5 (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:*

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

Note that by applying $H^{\otimes n}$, which is QFT for $\mathbb{F}_2^n$, to the state $|A_{s,s'}\rangle$, one obtains exactly $|A^\perp_{s',s}\rangle$.

Additionally, note that given $|A\rangle$ and s, s', one can efficiently construct $|A_{s,s'}\rangle$ as follows:

$$\sum_a |a\rangle \xrightarrow{\text{add } s} \sum_a |a+s\rangle \xrightarrow{H^{\otimes n}} \sum_{a' \in A^\perp} (-1)^{\langle a', s\rangle} |a'\rangle$$
$$\xrightarrow{\text{adding } s'} \sum_{a' \in A^\perp} (-1)^{\langle a', s\rangle} |a' + s'\rangle \xrightarrow{H^{\otimes n}} \sum_{a \in A} (-1)^{\langle a, s'\rangle} |a + s\rangle$$

For a subspace A and vectors s, s', we define $A + s = \{v + s : v \in A\}$, and $A^\perp + s' = \{v + s' : v \in A^\perp\}$.

When it is clear from the context, for ease of notation, we will write $A + s$ to mean the *program* that checks membership in $A + s$. For example, we will often write $\mathsf{iO}(A + s)$ to mean an iO obfuscation of the program that checks membership in $A + s$.

4.2 Direct Product Hardness

We describe the computational *direct product hardness* property satisfied by coset states. For more details, and a proof, we refer the reader to the full version.

Theorem 7. *Assume the existence of post-quantum* iO *and one-way function. Let* $A \subseteq \mathbb{F}_2^n$ *be a uniformly random subspace of dimension* $n/2$*, and* s, s' *be uniformly random in* $\mathbb{F}_2^n$*. Given one copy of* $|A_{s,s'}\rangle$*,* $\mathsf{iO}(A+s)$ *and* $\mathsf{iO}(A^\perp + s')$*, any polynomial time adversary outputs a pair* (v, w) *such that* $v \in A + s$ *and* $w \in A^\perp + s'$ *with negligible probability.*

The proof follows a similar outline to the proof of security of public-key quantum money in [Zha19b]. The main difference is that our proof handles (and leverages) coset states, instead of regular subspace states.

4.3 Monogamy-of-Entanglement Property

In this subsection, we argue that coset states satisfy an information-theoretic and a computational monogamy-of-entanglement property. We will not make use of these properties directly, instead we will have to rely on a stronger conjectured monogamy-of-entanglement property, which is presented in subsection 4.4. Thus, the properties that we prove in this subsection serve merely as "evidence" in support of the stronger conjecture. Due to lack of space, we only discuss the computational monogamy-of-entanglement property.

The game is between a challenger and an adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$.

- The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}^n$ 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 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, s_1') and (s_2, s_2').

$(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ wins if, for $i \in \{1, 2\}$, $s_i \in A + s$ and $s_i' \in A^\perp + s'$.

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

Theorem 8. *Assume the existence of post-quantum* iO *and one-way function, there exists a negligible function* $\mathsf{negl}(\cdot)$*, for any QPT adversary* $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*,*

$$\Pr[\mathsf{CompMonogamy}((\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2), n) = 1] = \mathsf{negl}(n)\,.$$

4.4 Conjectured Strong Monogamy Property

In this section, we describe a stronger version of the monogamy property, which we conjecture to hold. The monogamy property is a slight (but significant) variation of the one stated in the last section (which we proved to be true). Recall that there $\mathcal{A}_1$ and $\mathcal{A}_2$ are required to return pairs (s_1, s_1') and (s_2, s_2') respectively, such that both $s_1, s_2 \in A + s$ and $s_1', s_2' \in A^\perp + s'$. Now, we require that it is hard for $\mathcal{A}_1$ and $\mathcal{A}_2$ to even return a single string s_1 and s_2 respectively such that $s_1 \in A + s$ and $s_2 \in A^\perp + s'$.

Formally, consider the following 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 $|A_{s,s'}\rangle$ 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 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.

Let $\mathsf{ITStrongMonogamy}((\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2), n)$ 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. We conjecture the following:

Conjecture 1. There exists a sub-exponential function subexp such that, for any (unbounded) adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$,

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

Assuming the conjecture is true, and assuming post-quantum iO and one-way functions, we are able to prove the following computational strong monogamy statement. Consider a game between a challenger and an adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, which is identical to the one described above except that all $\mathcal{A}_0$ additionally gets the membership checking programs $\mathsf{iO}(A + s)$ and $\mathsf{iO}(A^\perp + s')$.

– 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 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, for $s_1 \in A + s$ 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, and takes the value 0 otherwise.

Theorem 9. *Assuming Conjecture 1 holds, and assuming the existence of post-quantum* iO *and one-way functions, then there exists a negligible function* $\mathsf{negl}(\cdot)$*, for any QPT adversary* $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*,*

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

We can further show a 'sub-exponential strong monogamy property' if we additionally assume sub-exponentially secure iO and one-way functions.

Theorem 10. *Assuming Conjecture 1 holds, and assuming the existence of sub-exponentially secure post-quantum* iO *and one-way functions, then for any QPT adversary* $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*,*

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

In the rest of the work, whenever we mention the 'strong monogamy property', we refer to the computational monogamy property of Theorem 9 above. Whenever we mention the 'sub-exponentially strong monogamy property', we refer to the computational monogamy property of Theorem 10.

5 Tokenized Signature Scheme from iO

In this section, we present tokenized signature scheme based on the computational direct product hardness property (Theorem 7).

5.1 Definitions

Definition 6 (Tokenized signature scheme). *A tokenized signature (TS) scheme consists of a tuple of QPT algorithms* $(\mathsf{KeyGen}, \mathsf{TokenGen}, \mathsf{Sign}, \mathsf{Verify})$ *with the following properties:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$*: Takes as input* 1^λ*, where* λ *is a security parameter, and outputs a secret key, public (verification) key pair* $(\mathsf{sk}, \mathsf{pk})$*.*
- $\mathsf{TokenGen}(\mathsf{sk}) \to |\mathsf{tk}\rangle$*: Takes as input a secret key* sk *and outputs a signing token* $|\mathsf{tk}\rangle$*.*
- $\mathsf{Sign}(m, |\mathsf{tk}\rangle) \to (m, \mathsf{sig})/\bot$*: Takes as input a message* $m \in \{0,1\}^*$ *and a token* $|\mathsf{tk}\rangle$*, and outputs either a message, signature pair* (m, sig) *or* $\bot$*.*
- $\mathsf{Verify}(\mathsf{pk}, m, \mathsf{sig}) \to 0/1$*: Takes as input an verification key, an alleged message, signature pair* (m, sig)*, and outputs* 0 *("reject") or* 1 *("accept").*

These algorithms satisfy the following. First is correctness. There exists a negligible function $\mathsf{negl}(\cdot)$, for any $\lambda \in \mathbb{N}$, $m \in \{0,1\}^*$,

$$\Pr[\mathsf{Verify}(\mathsf{pk}, m, \mathsf{sig}) = 1 : (m, \mathsf{sig}) \leftarrow \mathsf{Sign}(m, |\mathsf{tk}\rangle), |\mathsf{tk}\rangle \leftarrow \mathsf{TokenGen}(\mathsf{sk}), (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)] \geq 1 - \mathsf{negl}(\lambda)\,.$$

Definition 7 (Length restricted TS scheme). *A TS scheme is* r*-restricted if it holds only for* $m \in \{0,1\}^r$*. We refer to a scheme that is* 1*-restricted as a one-bit TS scheme.*

For notational purposes, we introduce an additional algorithm Verify_ℓ. The latter takes as input a public key pk and ℓ pairs $(m_\ell, \mathsf{sig}_\ell), \ldots, (m_\ell, \mathsf{sig}_\ell)$. It checks that $m_i \neq m_j$ for all $i \neq j$, and $\mathsf{Verify}(m_i, \mathsf{sig}_i) = 1$ for all $i \in [\ell]$; it outputs 1 if and only if they all hold. Next we define unforgeability.

Definition 8 (1-Unforgeability). *A TS scheme is* 1*-unforgeable if for every QPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$*, for every* λ*:*

$$\Pr\left[\begin{matrix}(m_0, \mathsf{sig}_0, m_1, \mathsf{sig}_1) \leftarrow \mathcal{A}(\mathsf{pk}, |\mathsf{tk}\rangle) \\ \mathsf{Verify}_2(\mathsf{pk}, m_0, \mathsf{sig}_0, m_1, \mathsf{sig}_1) = 1\end{matrix} : \begin{matrix}(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ |\mathsf{tk}\rangle \leftarrow \mathsf{TokenGen}(\mathsf{sk})\end{matrix}\right] \leq \mathsf{negl}(\lambda)\,.$$

Definition 9 (Unforgeability). *A TS scheme is unforgeable if for every QPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$*, for every* λ*,* $l = \mathsf{poly}(\lambda)$*:*

$$\Pr\left[\begin{matrix}\{m_i, \mathsf{sig}_i\}_{i\in[l+1]} \leftarrow \mathcal{A}(\mathsf{pk}, \{|\mathsf{tk}_i\rangle\}_{i\in[l]}) \\ \mathsf{Verify}_{l+1}(\mathsf{pk}, \{m_i, \mathsf{sig}_i\}_{i\in[l+1]}) = 1\end{matrix} : \begin{matrix}(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ |\mathsf{tk}_1\rangle \leftarrow \mathsf{TokenGen}(\mathsf{sk}) \\ \vdots \\ |\mathsf{tk}_l\rangle \leftarrow \mathsf{TokenGen}(\mathsf{sk})\end{matrix}\right] \leq \mathsf{negl}(\lambda)\,.$$

A tokenized signature scheme should also satisfy a revocability property. The revocability property follows straightforwardly from unforgeability [BS16]. Thus to show a construction is secure, we only need to focus on proving unforgeability.

The following theorem says that 1-unforgeability is sufficient to achieve a full blown TS scheme.

Theorem 11 ([BS16]). *A one-bit* 1*-unforgeable TS scheme implies a (full blown) TS scheme, assuming the existence of a quantum-secure digital signature scheme.*

In the next section, we give our construction of a one-bit 1-unforgeable TS scheme from coset states.

5.2 Tokenized Signature Construction

Construction.

- $\mathsf{KeyGen}(1^\lambda)$*: Set* $n = \mathsf{poly}(\lambda)$*. Sample uniformly* $A \subseteq \mathbb{F}_2^n$*. Sample* $s, s' \leftarrow \mathbb{F}_2^n$*. Output* $\mathsf{sk} = (A, s, s')$ *(where by* A *we mean a description of the subspace* A*) and* $\mathsf{pk} = (\mathsf{iO}(A + s), \mathsf{iO}(A^\perp + s'))$*.*
- $\mathsf{TokenGen}(\mathsf{sk})$*: Takes as input* sk *of the form* (A, s, s')*. Outputs* $|\mathsf{tk}\rangle = |A_{s,s'}\rangle$*.*
- $\mathsf{Sign}(m, |\mathsf{tk}\rangle)$*: Takes as input* $m \in \{0, 1\}$ *and a state* $|\mathsf{tk}\rangle$ *on* n *qubits. Compute* $H^{\otimes n}|\mathsf{tk}\rangle$ *if* $m = 1$*, otherwise do nothing to the quantum state. It then measures in the standard basis. Let* sig *be the outcome. Output* (m, sig)*.*
- $\mathsf{Verify}(\mathsf{pk}, (m, \mathsf{sig}))$*: Parse* pk *as* $\mathsf{pk} = (C_0, C_1)$ *where* C_0 *and* C_1 *are circuits. Output* $C_m(\mathsf{sig})$*.*

Theorem 12. *Assuming post-quantum* iO *and one-way function, the scheme of Construction 5.2 is a one-bit* 1*-unforgeable tokenized signature scheme.*

Proof. Security follows immediately from Theorem 7. □

Corollary 1. *Assuming post-quantum* iO*, one-way function(which implies digital signature) and a quantum-secure digital signature scheme, there exists a (full blown) tokenized signature scheme.*

Proof. This is an immediate consequence of Theorems 11 and 12. □

6 Single-Decryptor Encryption

In this section, we formally introduce unclonable decryption, i.e. single-decryptor encryption [GZ20]. Then we describe two constructions and prove their security.

Our first construction (Section 6.2) relies on the strong monogamy-of-entanglement property (Conjecture 1), the existence of post-quantum one-way function, indistinguishability obfuscation and compute-and-compare obfuscation for (sub-exponentially) unpredictable distributions (whose existence has been discussed in Section 3.1). Our second construction has a similar structure. It does not

rely on the strong monogamy-of-entanglement property for coset states, but on the (weaker) direct product hardness property (Theorem 7). However, the construction additionally relies on a much stronger cryptographic primitive – post-quantum extractable witness encryption (as well post-quantum one-way functions and indistinguishability obfuscation). Due to lack of space, we refer the reader to the full version for further the latter construction.

6.1 Definitions

Definition 10 (Single-Decryptor Encryption Scheme). *A single-decryptor encryption scheme consists of the following efficient algorithms:*

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$: *a (classical) probabilistic algorithm that takes as input a security parameter λ and outputs a classical secret key sk and public key pk.*
- $\mathsf{QKeyGen}(\mathsf{sk}) \to \rho_{\mathsf{sk}}$: *a quantum algorithm that takes as input a secret key sk and outputs a quantum secret key ρ_{sk}.*
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$: *a (classical) probabilistic algorithm that takes as input a public key pk, a message m and outputs a classical ciphertext ct.*
- $\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct}) \to m/\bot$: *a quantum algorithm that takes as input a quantum secret key ρ_{sk} and a ciphertext ct, and outputs a message m or a decryption failure symbol $\bot$.*

A secure single-decryptor encryption scheme should satisfy the following:

Correctness: There exists a negligible function $\mathsf{negl}(\cdot)$, for all $\lambda \in \mathbb{N}$, for all $m \in \mathcal{M}$,

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

Note that correctness implies that a honestly generated quantum decryption key can be used to decrypt correctly polynomially many times, from the gentle measurement lemma [Aar05].

CPA Security: The scheme should satisfy (post-quantum) CPA security, i.e. indistinguishability under chosen-plaintext attacks: for every (stateful) QPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, the following holds:

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

Anti-Piracy Security Next, we define anti-piracy security via the anti-piracy game below. Recall that, intuitively, anti-piracy security says that it is infeasible for a pirate who receives a quantum secret key to produce two quantum keys, which both allow successful decryption. This can be formalized as:

(CPA-style Anti-piracy) We ask the pirate to provide a pair of messages (m_0, m_1) along with two quantum secret keys, and we test whether the two keys allow to (simultanoeusly) distinguish encryptions of m_0 and m_1.

In order to describe the security games, it is convenient to first introduce the concept of a *quantum decryptor*. The following definition is implicitly with respect to some single-decryptor encryption scheme $(\mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Enc}, \mathsf{Dec})$.

Definition 11 (Quantum decryptor). *A* quantum decryptor *for ciphertexts of length n, is a pair (ρ, U) where ρ is a state, and U is a general quantum circuit acting on $n + m$ qubits, where m is the number of qubits of ρ.*

For a ciphertext c of length n, we say that we run the quantum decryptor (ρ, U) on ciphertext c to mean that we execute the circuit U on inputs $|c\rangle$ and ρ.

We are now ready to describe the CPA-style anti-piracy game.

Definition 12 (Anti-Piracy Game, CPA-style). *Let $\lambda \in \mathbb{N}^+$. The CPA-style anti-piracy game is the following game between a challenger and an adversary $\mathcal{A}$.*

1. ***Setup Phase**: The challenger samples keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$.*
2. ***Quantum Key Generation Phase**: The challenger sends $\mathcal{A}$ the classical public key pk and one copy of quantum decryption key $\rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$.*
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 two registers R_1, R_2 and two general quantum circuits U_1 and U_2. We interpret $\mathcal{A}$'s output as two (possibly entangled) quantum decryptors $\mathsf{D}_1 = (\sigma[R_1], U_1)$ and $\mathsf{D}_2 = (\sigma[R_2], U_2)$.*
4. ***Challenge Phase:** The challenger samples b_1, b_2 and r_1, r_2 uniformly at random and generates ciphertexts $c_1 = \mathsf{Enc}(\mathsf{pk}, m_{b_1}; r_1)$ and $c_2 = \mathsf{Enc}(\mathsf{pk}, m_{b_2}; r_2)$. The challenger runs quantum decryptor D_1 on c_1 and D_2 on c_2, and checks that D_1 outputs m_{b_1} and D_2 outputs m_{b_2}. If so, the challenger outputs 1 (the game is won by the adversary), otherwise outputs 0.*

We denote by $\mathsf{AntiPiracyCPA}(1^\lambda, \mathcal{A})$ a random variable for the output of the game.

Note that an adversary can succeed in this game with probability at least $1/2$. It simply gives ρ_{sk} to the first quantum decryptor and the second decryptor randomly guesses the plaintext.

We remark that one could have equivalently formulated this definition by having the pirate send registers R_1 and R_2 to two separated parties Bob and Charlie, who then receive ciphertexts from the challenger sampled as in the Challenge Phase above. The two formulations are equivalent upon identifying the quantum circuits U_1 and U_2.

Definition 13 (Anti-Piracy Security, CPA-style). *Let $\gamma : \mathbb{N}^+ \to [0, 1]$. A single-decryptor encryption scheme satisfies γ-anti-piracy security, if for any QPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds for all $\lambda \in \mathbb{N}$:*

$$\Pr\left[b = 1, b \leftarrow \mathsf{AntiPiracyCPA}(1^\lambda, \mathcal{A})\right] \leq \frac{1}{2} + \gamma(\lambda) + \mathsf{negl}(\lambda) \tag{1}$$

It is not difficult to show that if γ-anti-piracy security holds for all inverse poly γ, then this directly implies CPA security (we refer the reader to the full version for the proof of this implication).

6.2 Construction from Strong Monogamy Property

In this section, we give our first construction of a single-decryptor encryption scheme, whose security relies on the strong monogamy-of-entanglement property from Section 4.4.

In the rest of the paper, to simplify notation, whenever it is clear from the context, we will denote a program that checks membership in a set S simply by S.

Construction 13.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$:
  - *Sample κ random $(n/2)$-dimensional subspaces $A_i \subseteq \mathbb{F}_2^n$ for $i = 1, 2, \cdots, \kappa$, where $n = \lambda$ and $\kappa = \kappa(\lambda)$ is a polynomial in λ.*
  - *For each $i \in [\kappa]$, 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 $\mathsf{sk} = \{A_i, s_i, s'_i\}_{i\in[\kappa]}, \mathsf{pk} = \{\mathsf{iO}(A_i + s_i), \mathsf{iO}(A_i^\perp + s'_i)\}_{i\in[\kappa]}$.*
- $\mathsf{QKeyGen}(\mathsf{sk}) \to \rho_{\mathsf{sk}}$: *on input $\mathsf{sk} = \{A_i, s_i, s'_i\}_{i\in[\kappa]}$, output the "quantum secret key" $\rho_{\mathsf{sk}} = \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\kappa]}$. Recall that each $|A_{i,s_i,s'_i}\rangle$ is*

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

- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$: *on input a public key $\mathsf{pk} = \{\mathsf{iO}(A_i + s_i), \mathsf{iO}(A_i^\perp + s'_i)\}_{i\in[\kappa]}$ and message m:*
  - *Sample a uniformly random string $r \leftarrow \{0,1\}^\kappa$.*
  - *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 $P_{m,r}$ be the following program:*

> On input $u = u_1||u_2||\cdots||u_\kappa$ (where each $u_i \in \mathbb{F}_2^n$):
> 1. If for all $i \in [\kappa]$, $R_i^{r_i}(u_i) = 1$:
> Output m
> 2. Else:
> Output $\perp$

Fig. 1. Program $P_{m,r}$

  - *Let $\hat{P}_{m,r} = \mathsf{iO}(P_{m,r})$. Output ciphertext $\mathsf{ct} = (\hat{P}_{m,r}, r)$.*

– $\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct}) \to m/\bot$: *on input* $\rho_{\mathsf{sk}} = \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\kappa]}$ *and* $\mathsf{ct} = (\hat{P}_{m,r}, r)$*:*
  - *For each* $i \in [\kappa]$*, 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* $\hat{P}_{m,r}$ *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.*

Correctness. Correctness and efficiency easily follow from the construction.

For security, we have the following theorem (proved in the full version):

Theorem 14 (Regular Anti-Piracy). *Assuming the existence of post-quantum* iO*, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions (as in Definition 3), and the strong monogamy-of-entanglement property (Conjecture 1), the single-decryptor encryption scheme of Construction 13 has regular* γ*-anti-piracy security for* $\gamma = 0$*.*

Similarly, assuming the existence of post-quantum sub-exponentially secure iO*, one-way functions, the quantum hardness of LWE and assuming the strong monogamy-of-entanglement property (Conjecture 1), the single-decryptor encryption scheme of Construction 13 has regular* γ*-anti-piracy security for* $\gamma = 0$*.*

In the above theorem, the quantum hardness of LWE is used to build compute-and-compare obfuscation for sub-exponentially unpredictable distributions.

7 Copy-Protection of Pseudorandom Functions

In this section, we formally define copy-protection of pseudorandom functions. Then, we describe a construction that essentially builds on the single-decryptor encryption scheme described in Section 6.2 (together with post-quantum sub-exponentially secure one-way functions and iO). We remark that all of the PRFs that we use can be constructed from post-quantum one-way functions. We refer the reader to [SW14] and the full version for further details.

7.1 Definitions

In what follows, the PRF $F : [K] \times [N] \to [M]$, implicitly depends on a security parameter λ. We denote by $\mathsf{Setup}(\cdot)$ the procedure that samples a PRF key.

Definition 14 (Copy-Protection of PRF). *A copy-protection scheme for a PRF* $F : [K] \times [N] \to [M]$ *consists of the following polynomial-time algorithms:*

$\mathsf{QKeyGen}(K)$*: takes a key* K *and outputs a quantum key* ρ_K*;*
$\mathsf{Eval}(\rho_K, x)$*: takes a quantum key* ρ_K *and an input* $x \in [N]$*. It outputs a classical string* $y \in [M]$*.*

A copy-protection scheme should satisfy the following properties:

Definition 15 (Correctness). *There exists a negligible function* $\mathsf{negl}(\cdot)$*, for all* λ*, all* $K \leftarrow \mathsf{Setup}(1^\lambda)$*, all inputs* x*,*

$$\Pr[\mathsf{Eval}(\rho_K, x) = F(K, x) : \rho_K \leftarrow \mathsf{QKeyGen}(K)] \geq 1 - \mathsf{negl}(\lambda)\,.$$

Note that the correctness property implies that the evaluation procedure has an "almost unique" output. This means that the PRF can be evaluated (and rewound) polynomially many times, without disturbing the quantum key ρ_K, except negligibly.

Definition 16 (Anti-Piracy Security). *Let* $\lambda \in \mathbb{N}^+$*. Consider the following game between a challenger and an adversary* $\mathcal{A}$*:*

1. *The challenger samples* $K \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $\rho_K \leftarrow \mathsf{QKeyGen}(K)$*. It gives* ρ_K *to* $\mathcal{A}$*;*
2. $\mathcal{A}$ *returns to the challenger a bipartite state* σ *on registers* R_1 *and* R_2*, as well as general quantum circuits* U_1 *and* U_2*.*
3. *The challenger samples uniformly random* $u, w \leftarrow [N]$*. Then runs* U_1 *on input* $(\sigma[R_1], u)$*, and runs* U_2 *on input* $(\sigma[R_2], w)$*. The outcome of the game is* 1 *if and only if the outputs are* $F(K, u)$ *and* $F(K, w)$ *respectively.*

Denote by $\mathsf{CopyProtectionGame}(1^\lambda, \mathcal{A})$ *a random variable for the output of the game.*

We say the scheme has anti-piracy security if for every polynomial-time quantum algorithm $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$*, for all* $\lambda \in \mathbb{N}^+$*,*

$$\Pr\left[b = 1, b \leftarrow \mathsf{CopyProtectionGame}(1^\lambda, \mathcal{A})\right] = \mathsf{negl}(\lambda)\,.$$

7.2 Construction

In this section, we describe a construction of a copy-protection scheme for a class of PRFs. We will eventually reduce security of this construction to security of the single-decryptor encryption scheme of Section 6.2, and we will therefore inherit the same assumptions.

Let λ be the security parameter. Our construction copy-protects a PRF $F_1 : [K_\lambda] \times [N_\lambda] \to [M_\lambda]$ where $N = 2^{n(\lambda)}$ and $M = 2^{m(\lambda)}$, for some polynomials $n(\lambda)$ and $m(\lambda)$, satisfying $n(\lambda) \geq m(\lambda) + 2\lambda + 4$. For convenience, we will omit writing the dependence on λ, when it is clear from the context. Moreover, F_1 should be a puncturable extracting PRF with error $2^{-\lambda-1}$. Such PRFs exist assuming post-quantum one-way functions.

Our copy-protection construction for F_1, will make use of the following additional building blocks:

1. A puncturable extracting PRF $F_1(K_1, \cdot)$ that accepts inputs of length $n = \ell_0 + \ell_1 + \ell_2$ and outputs strings of length m. It is extracting when the input min-entropy is greater than $m + 2\lambda + 4$. By Theorem 3 in [SW14], assuming one-way functions exist, as long as $n \geq m + 2\lambda + 4$, F_1 is a puncturable extracting PRF with error less than $2^{-\lambda-1}$.

2. A puncturable statistically injective PRF $F_2(K_2, \cdot)$ that accepts inputs of length ℓ_2 and outputs strings of length ℓ_1. By Theorem 2 in [SW14], assuming one-way functions exist, as long as $\ell_1 \geq 2\ell_2 + \lambda$, F_2 is a puncturable statistically injective PRF with failure probability $2^{-\lambda}$.
3. A puncturable PRF $F_3(K_3, \cdot)$ that accepts inputs of length ℓ_1 and outputs strings of length ℓ_2. By Theorem 1 in [SW14], assuming one-way functions exist, F_3 is a puncturable PRF.

Note that PRF $F_1(K_1, \cdot)$ is the PRF functionality we will copy-protect. The PRFs $F_2(K_2, \cdot)$, $F_3(K_3, \cdot)$ are just building blocks in the construction.

In Figures 2, 3, we describe a copy-protection construction for PRF F_1.

$\mathsf{QKeyGen}(K_1)$: Sample uniformly random subspaces A_i of dimension $\lambda/2$ and vectors s_i, s'_i for $i = 1, 2, \cdots, \ell_0$. Sample PRF keys K_2, K_3 for F_2, F_3. Prepare the programs $R_i^0 = \mathsf{iO}(A_i + s_i)$ and $R_i^1 = \mathsf{iO}(A_i^{\perp} + s'_i)$ (with appropriately padded length), and let P be the program described in Figure 3. Output the quantum key $\rho_K = (\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\ell_0]}, \mathsf{iO}(P))$.

$\mathsf{Eval}(\rho_K, x)$: Let $\rho_K = (\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\ell_0]}, \mathsf{iO}(P))$. Let $x = x_0||x_1||x_2$ and x_0 is of length ℓ_0. For all $i \in [\ell_0]$, if $x_{0,i}$ is 1, apply $H^{\otimes n}$ to $|A_{i,s_i,s'_i}\rangle$. Otherwise, leave the state unchanged.
Let σ be the resulting state. Run $\mathsf{iO}(P)$ coherently on input x and σ, and measure the final output register to obtain y.

Fig. 2. Quantum copy-protection scheme for PRFs.

The program P takes as input x and a list of ℓ_0 vectors $v_1, \cdots, v_{\ell_0}$ and has two modes. If x is not in the sparse hidden trigger set (not passing the 'if' check in the first line), the program is in the *normal mode*: it only outputs a PRF evaluation of x if every v_i is in the corresponding coset. Otherwise, the program is in the *hidden trigger mode*. It will compute a circuit Q' from the input x and output $Q'(v_1, \cdots, v_{\ell_0})$. On almost all inputs except those sparse hidden triggers, the program runs in its normal mode.

We obtain the following theorem (which we prove in the full version):

Theorem 15. *Assuming the existence of post-quantum* iO*, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions (as in Definition 3), and the strong monogamy-of-entanglement property (Conjecture 1), our construction satisfies anti-piracy security (as in Definition 16).*

Similarly, assuming the existence of post-quantum sub-exponentially secure iO*, one-way functions, the quantum hardness of LWE and assuming the strong monogamy-of-entanglement property (Conjecture 1), our construction satisfies anti-piracy security.*

Hardcoded: Keys K_1, K_2, K_3, R_i^0, R_i^1 for all $i \in [\ell_0]$.
On input $x = x_0||x_1||x_2$ and vectors $v_1, \cdots, v_{\ell_0}$:

1. If $F_3(K_3, x_1) \oplus x_2 = x_0'||Q'$ and $x_0 = x_0'$ and $x_1 = F_2(K_2, x_0'||Q')$:
   Hidden Trigger Mode: It treats Q' as a (classical) circuit and outputs $Q'(v_1, \cdots, v_{\ell_0})$.
2. Otherwise, it checks if the following holds: for all $i \in [\ell_0]$, $R^{x_{0,i}}(v_i) = 1$.
   Normal Mode: If all hold, it outputs $F_1(K_1, x)$. Otherwise, it outputs $\perp$.

Fig. 3. Program P

Acknowledgements

A.C. is supported by the Simons Institute for the Theory of Computing, through a Quantum Postdoctoral Fellowship. J. L., Q. L. and M. Z. are supported by the NSF. J. L. is also supported by Scott Aaronson's Simons Investigator award. The authors are grateful for the support of the Simons Institute, where this collaboration was initiated.

References

[Aar05] Scott Aaronson. "Limitations of Quantum Advice and One-Way Communication". In: *Theory of Computing* 1.1 (2005), pp. 1–28. DOI: 10.4086/toc.2005.v001a001. URL: https://doi.org/10.4086/toc.2005.v001a001.

[Aar09] Scott Aaronson. "Quantum copy-protection and quantum money". In: *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE. 2009, pp. 229–242.

[AC12] Scott Aaronson and Paul Christiano. "Quantum money from hidden subspaces". In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM. 2012, pp. 41–60.

[ALL+20] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. *New approaches for quantum copy-protection*. 2020.

[AP21] Prabhanjan Ananth and Rolando L. La Placa. *Secure Software Leasing*. 2021.

[BB84] Charles H Bennett and Gilles Brassard. *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*. 1984.

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. *Factoring and Pairings are not Necessary for iO: Circular-Secure LWE Suffices*. Cryptology ePrint Archive, Report 2020/1024. https://eprint.iacr.org/2020/1024. 2020.

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. "Preventing Zeroizing Attacks on GGH15". In: *Proceedings of TCC 2018*. 2018.

[BJL+21] Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram. *Secure Software Leasing Without Assumptions*. 2021. arXiv: 2101.12739 [quant-ph].

[BL19] Anne Broadbent and Sébastien Lord. "Uncloneable Quantum Encryption via Random Oracles". In: *IACR Cryptology ePrint Archive* 2019 (2019), p. 257.

[BS16] Shalev Ben-David and Or Sattath. "Quantum tokens for digital signatures". In: *arXiv preprint arXiv:1609.09047* (2016).

[BW13] Dan Boneh and Brent Waters. "Constrained pseudorandom functions and their applications". In: *International conference on the theory and application of cryptology and information security*. Springer. 2013, pp. 280–300.

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. *Quantum copy-protection of compute-and-compare programs in the quantum random oracle model*. 2020. arXiv: 2009.13865 [quant-ph].

[FGH+12] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. "Quantum money from knots". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. 2012, pp. 276–289.

[GGHW17] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. "On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input". In: *Algorithmica* 79.4 (2017), pp. 1353–1373.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to Construct Random Functions". In: *J. ACM* 33.4 (Aug. 1986), pp. 792–807. ISSN: 0004-5411. DOI: 10.1145/6490.6503. URL: https://doi.org/10.1145/6490.6503.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. "Lockable obfuscation". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 612–621.

[Got02] Daniel Gottesman. "Uncloneable encryption". In: *arXiv preprint quant-ph/0210062* (2002).

[GZ20] Marios Georgiou and Mark Zhandry. *Unclonable Decryption Keys*. Cryptology ePrint Archive, Report 2020/877. https://eprint.iacr.org/2020/877. 2020.

[JLS20] Aayush Jain, Huijia Lin, and Amit Sahai. *Indistinguishability Obfuscation from Well-Founded Assumptions*. Cryptology ePrint Archive, Report 2020/1003. https://eprint.iacr.org/2020/1003. 2020.

[Kan18] Daniel M Kane. "Quantum money from modular forms". In: *arXiv preprint arXiv:1809.05925* (2018).

[KNY20] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. *Secure Software Leasing from Standard Assumptions*. 2020. arXiv: 2010.11186 [quant-ph].

[Lut10] Andrew Lutomirski. "An online attack against Wiesner's quantum money". In: *arXiv preprint arXiv:1010.0256* (2010).

[SW14] Amit Sahai and Brent Waters. "How to use indistinguishability obfuscation: deniable encryption, and more". In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 2014, pp. 475–484.

[TFKW13] Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. "A monogamy-of-entanglement game with applications to device-independent quantum cryptography". In: *New Journal of Physics* 15.10 (2013), p. 103002.

[Wie83] Stephen Wiesner. "Conjugate coding". In: *ACM Sigact News* 15.1 (1983), pp. 78–88.

[WW20] Hoeteck Wee and Daniel Wichs. *Candidate Obfuscation via Oblivious LWE Sampling*. Cryptology ePrint Archive, Report 2020/1042. https://eprint.iacr.org/2020/1042. 2020.

[WZ17] Daniel Wichs and Giorgos Zirdelis. "Obfuscating compute-and-compare programs under LWE". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 600–611.

[Zha19a] Mark Zhandry. "Quantum lightning never strikes the same state twice". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2019, pp. 408–438.

[Zha19b] Mark Zhandry. "Quantum lightning never strikes the same state twice". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2019, pp. 408–438.

[Zha19c] Mark Zhandry. "The magic of ELFs". In: *Journal of Cryptology* 32.3 (2019), pp. 825–866.