Unclonable Cryptography in the Plain Model

Céline Chevalier[1,2], Paul Hermouet[1,2,3], and Quoc-Huy Vu[4]

[1] DIENS, École normale supérieure, PSL University, CNRS, INRIA, Paris, France
[2] CRED, Université Panthéon-Assas Paris II, Paris, France
[3] LIP6, Sorbonne Université, Paris, France
[4] Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France
{celine.chevalier, paul.hermouet, quoc.huy.vu}@ens.fr

Abstract. By leveraging the no-cloning principle of quantum mechanics, unclonable cryptography enables us to achieve novel cryptographic protocols that are otherwise impossible classically. Two most notable examples of unclonable cryptography are quantum copy-protection and unclonable encryption. Despite receiving a lot of attention in recent years, two important open questions still remain: copy-protection for point functions in the plain model, which is usually considered as feasibility demonstration, and unclonable encryption with unclonable indistinguishability security in the plain model.
In this work, by relying on previous works of Coladangelo, Liu, Liu, and Zhandry (Crypto'21) and Culf and Vidick (Quantum'22), we establish a new monogamy-of-entanglement property for subspace coset states, which allows us to obtain the following new results:

- We show that copy-protection of point functions exists in the plain model, with different challenge distributions (including arguably the most natural ones).
- We show, for the first time, that unclonable encryption with unclonable indistinguishability security exists in the plain model.

1 Introduction

Quantum information enables us to achieve new cryptographic primitives that are impossible classically, leading to a prominent research area named unclonable cryptography. At the heart of this area is the no-cloning principle of quantum mechanics [WZ82], which has given rise to many unclonable cryptographic primitives. This includes quantum money [Wie83], quantum copy-protection [Aar09], unclonable encryption [BL20], single-decryptor encryption [CLLZ21], and many more. In this work, our focus is on quantum copy-protection and unclonable encryption.

Copy-protection for point functions. Quantum copy-protection, introduced by Aaronson in [Aar09], is a functionality-preserving compiler that transforms programs into quantum states. Moreover, we require that the resulting copy-protected state should not allow the adversary to copy the functionality of the state. In particular, this unclonability property, which is often associated with challenge distributions, is stated as the following game. A splitting adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ first has $\mathcal{A}$ receive as input a copy-protected state that can be used to compute a function f. $\mathcal{A}$ then outputs a bipartite state to $\mathcal{B}$ and $\mathcal{C}$. We require that both $\mathcal{B}$ and $\mathcal{C}$ should not be able to simultaneously compute f, where the inputs given to $\mathcal{B}$ and $\mathcal{C}$ are drawn from a distribution, called a *challenge distribution*.

While copy-protection is known to be impossible for general unlearnable functions and the class of de-quantumizable algorithms [AL21], several feasibility results have been demonstrated for cryptographic functions (e.g., pseudorandom functions, decryption and signing algorithm [CLLZ21,LLQZ22]). Of particular interest to us is the class of point functions, which is of the form $f_y(\cdot)$: it takes as input x and outputs 1 if and only if $x = y$.

Prior works [CMP20,AK21,AKL$^+$22,AKL23,CHV23] achieved a copy-protection scheme for point functions with different type of states (e.g., BB84 states [BB20] or coset states [CLLZ21]) and different challenge distributions. However, in contrast to known constructions for copy-protection for cryptographic functions which are in the plain model, these constructions for copy-protection for point functions are all in the quantum

random oracle model. The only known copy-protection for point functions scheme in the plain model was recently constructed in [CHV23], but this scheme was shown to be secure with respect to a "less natural" challenge distribution. We note that different feasibility for the same copy-protection scheme, based on different challenge distributions, can be qualitatively incomparable. That is, security established under one challenge distribution might not necessarily guarantee security under a different challenge distribution.

Given the inability to prove security with respect to certain natural challenge distributions for copy-protection for point functions, an important question that has been left open from prior works is the following:

Question 1. *Do copy-protection schemes for point functions, with negligible security and natural challenge distributions, in the plain model exist?*

Unclonable Encryption. Unclonable encryption, introduced by Broadbent and Lord [BL20], is another beautiful primitive of unclonable cryptography. Roughly speaking, unclonable encryption is an encryption scheme with quantum ciphertexts having the following security guarantee: given a quantum ciphertext, no adversary can produce two (possibly entangled) states that both encode some information about the original plaintext. Interestingly, besides its own applications, unclonable encryption also implies private-key quantum money, and copy-protection for a restricted class of functions [BL20,AK21].

Despite being a natural primitive, constructing unclonable encryption has remained elusive. Prior works [BL20,AK21] established the feasibility of unclonable encryption satisfying a weaker property called unclonability, which can be seen as a *search*-type security. This weak security notion is far less useful, as it does not imply the standard semantic security of an encryption scheme, and also does not lead to the application implication listed above. The stronger notion, the so-called *unclonable indistinguishability*, is only known to be achievable in the quantum random oracle model [AKL$^+$22]. Given the notorious difficulty of building unclonable encryption in the standard model, the following question has been left open from prior works:

Question 2. *Do encryption schemes satisfying unclonable indistinguishability in the plain model exist?*

In this paper, we answer the two questions above affirmatively. Firstly, we present a construction of copy-protection of point functions with negligible security in the plain model. We show that this construction is secure, for three families of distributions: product distributions, identical distributions and non-colliding distribution. Secondly, we exhibit two constructions of unclonable encryption with unclonable indistinguishability security in the plain model: one for single-bit encryption and the other for multi-bit encryption. Our constructions based on the construction of single-decryptor, introduced by [CLLZ21], with new security variants. In the process, we also present a new monogamy-of-entanglement (MoE) property of coset states which might be of independent interest.

Concurrent and Independent Work. Very recently, two concurrent and independent works have consider similar tasks. However, at a high level, the themes of these two papers and ours are quite different. Coladangelo and Gunn [CG23] show the feasibility of copy-protection of puncturable functionalities and point functions through a new notion of quantum state indistinguishability obfuscation, which is also introduced in the same paper. Ananth and Behera [AB23] also show constructions for copy-protection of puncturable functionalities (including point functions) and unclonable encryption, based on a new notion of unclonable puncturable obfuscation. Among the two, the latter is most similar to our work. Their construction of unclonable puncturable obfuscation, which is the backbone for their applications (of copy-protection of point functions and unclonable encryption), is based on the recent construction of copy-protection of pseudorandom functions and single-decryptor of Coladangelo et al. [CLLZ21]. They show that a slightly modified construction of [CLLZ21] achieves anti-piracy security with different challenge distributions and preponed security. Apart from the naming, these security notions are identical to what we consider here in our paper. However, the security of their schemes are based on a new conjecture. We directly show that the same constructions of [CLLZ21] achieve these security notions, by proving a new monogamy-of-entanglement property of coset states. This allows us to obtain the first constructions of copy-protection of point functions and unclonable

encryption in the plain model based on well-studied cryptographic assumptions. Lastly, while the two concurrent works propose more generic approaches, we believe that our concrete constructions might be applicable on these approaches to obtain feasibility results based on standard assumptions.

1.1 Technical Overview

We first recall the anti-piracy security definition, discuss several challenge distributions for copy-protection of point functions, and then present techniques to achieve security with respect to these challenge distributions.

Anti-piracy security. A piracy game is formalized as a security experiment against cloning adversaries of the form $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$. The pirate $\mathcal{P}$ receives an copy-protected program $\rho_f \coloneqq \mathsf{Protect}(f)$, which can be used to evaluate a classical function f, prepares a bipartite state, and sends each half of the state to two non-communicating freeloaders $\mathcal{F}_1$ and $\mathcal{F}_2$. In the challenge phase, $\mathcal{F}_1$ and $\mathcal{F}_2$ receive inputs c_1, c_2, sampled from a challenge distribution and are asked to output b_1, b_2. $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ win if $b_i = f(c_i)$ for $i \in \{1, 2\}$.

It turns out that the choice of challenge distribution plays a crucial role in evaluating security of copy-protection schemes. Indeed, previous constructions of copy-protection of point functions have considered different challenge distributions [CMP20,BJL+21,AKL+22,CHV23]. Some are considered "less natural" than the others. Ideally, we would like to prove security of the scheme in a way that is independent of the chosen challenge distribution. In this paper, we make progress towards achieving this goal. In particular, in the following, let $y \in \{0,1\}^n$ the copy-protected point, x, x' random strings drawn from some distribution. We consider the following challenge distributions for copy-protecting point functions.

- *Identical:* $\mathcal{F}_1$ and $\mathcal{F}_2$ get either (y, y) or (x, x) with probability $\frac{1}{2}$ each, where x is drawn uniformly at random from $\{0,1\}^n \setminus \{y\}$.
- *Product:* $\mathcal{F}_1$ and $\mathcal{F}_2$ get either (y, y), (x, y), (y, x), or (x, x') each with probability $\frac{1}{4}$, where x, x' are drawn uniformly at random from $\{0,1\}^n \setminus \{y\}$.
- *Non-Colliding:* $\mathcal{F}_1$ and $\mathcal{F}_2$ get either (x, y), (y, x), or (x, x') each with probability $\frac{1}{3}$, where x, x' are drawn uniformly at random from $\{0,1\}^n \setminus \{y\}$.

Arguably, since the copy-protected point basically represents the entire functionality of the point function, one would say the product distribution is the most meaningful and natural one. However, the only known construction known before our work that achieves copy-protection of point functions in the plain model is the one given in [CHV23], which only achieves security w.r.t non-colliding distribution. Our construction is identical to that of [CHV23] and our main technical contribution lies in our proof technique showing that [CHV23] construction can achieve security with respect to product and identical distributions.

We continue by recalling [CHV23] construction and briefly explain where it fails when proving security w.r.t the product challenge distribution, then we describe our new proof techniques that allow us to overcome the problems.

[CHV23]'s copy-protection of point functions. At a high level, [CHV23] scheme uses a copy-protection scheme of pseudorandom functions (PRFs) $\mathsf{PRF}(\mathsf{k}, \cdot)$ from [CLLZ21]. Protecting a point function PF_y is done in the following way: sample a PRF key k; then copy-protect k using the PRF protection algorithm to get ρ_{k}; and finally compute $z \leftarrow \mathsf{PRF}(\mathsf{k}, y)$ and return the outcome z as well as ρ_{k}. One can evaluate the copy-protected point function PF_y on an input x in the following way: compute $\mathsf{PRF}(\mathsf{k}, x)$ using the evaluation algorithm of the PRF copy-protection scheme, then check whether the outcome equals z or not and return 1 or 0 accordingly. Although [CHV23] construction can be cast in the form, we note that their reduction (and ours) go through an intermediate notion of single decryptor, which ultimately reduces to some form of monogamy-of-entanglement of hidden coset states. We refer the reader to the formal proof provided in Section 4 and Section 5 for more details.

Challenges when proving anti-piracy security w.r.t the product distribution. A coset state is a quantum state of the form $|A_{s,s'}\rangle \coloneqq \frac{1}{\sqrt{|A|}} \sum_{x \in A} (-1)^{\langle x,s' \rangle} |x+s\rangle$ for a subspace $A \subseteq \mathbb{F}_2^n$ and two vectors $s, s' \in \mathbb{F}_2^n$. Loosely speaking, a coset state $|A_{s,s'}\rangle$ embeds information on both the coset $A+s$ and its dual $A^\perp + s'$, and has the following monogamy-of-entanglement property: given a random coset state $|A_{s,s'}\rangle$, no adversary can split the state and share it to two non-communicating freeloaders such that, given the description of the subspace A, the first freeloader returns a vector in the coset $A+s$ and the second one a vector in the dual $A^\perp + s'$. To prove security based on monogamy-of-entanglement of coset states, the authors of [CHV23] (based on techniques from [CLLZ21]) use an extraction property of lockable obfuscation to extract and outputs two vectors which, with non-negligible probability belong respectively to $A+s$ and $A^\perp + s'$, which works perfectly when the challenge distribution is non-colliding. However, when considering the identical distribution (or the product distribution for the case when the challenge inputs are (y,y)), the adversary is required to output two vectors (not necessarily different) from *the same* coset space: that is, they are either both in $A+s$ or $A^\perp + s'$. This in turn leads to no violation against the monogamy-of-entanglement game describe above. Worse, if the adversary knows which basis it would play with (either the computational basis for coset space $A+s$ or the Hadamard basis for coset space $A^\perp + s'$), the adversary can win the game trivially.

Unclonable encryption. In this paper, we also propose a construction for unclonable encryption with unclonable indistinguishability in the plain model. The unclonable indistinguishability for this primitive is also defined through a piracy game, in which the pirate $\mathcal{P}$ receives a quantum encryption of a bit b, prepares a bipartite state, and sends each half of the state to two non-communicating freeloaders $\mathcal{F}_1$ and $\mathcal{F}_2$. In the challenge phase, $\mathcal{F}_1$ and $\mathcal{F}_2$ both receive the decryption key k and are asked to output b_1, b_2. $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ win if $b_i = b$ for $i \in \{1,2\}$. Our construction of unclonable encryption also uses a copy-protection scheme of PRF. A key is simply a random bitstring k_S. Encrypting a bit b is done by in the following way: sample a PRF key k_P; then copy-protect k_P using the PRF protection algorithm to get ρ_{k_P}; finally sample a fresh random bitstring r and output $(r, y, \rho_{\mathsf{k}_P})$ where y is either $\mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$ if $b = 0$, or a random bitstring if $b = 1$. Similarly as for copy-protection of point function, the security of our unclonable encryption construction also reduces to a monogamy-of-entanglement game. As in the piracy game for this primitive, the same challenge is used for both $\mathcal{F}_1$ and $\mathcal{F}_2$, we meet the same problem as for our copy-protection construction, namely that the adversaries are required to output two vectors from the same coset space.

Our technique: a new monogamy-of-entanglement game of coset states. We resolve this problem by introducing a new variant of the monogamy-of-entanglement property of coset states. In this variant, the freeloaders both have to output a vector in the same coset space, either $A+s$ or $A^\perp + s'$, but they learn the challenge coset space only during the challenge phase after receiving the state from the pirate $\mathcal{P}$. Crucially the pirate also does not know the challenge coset space before the challenge phase. We call this new game as *Monogamy-of-Entanglement Game with Identical Basis*. An illustration of this new game is depicted in Section 1.1.

We show that the winning probability of this game is at most negligibly far way from $1/2$, which corresponds to the trivial strategy in which the pirate always measures the coset state in the computational basis and forwards the outcome to both the freeloaders, who in turn output it. An overview of the proof is given at the end of the section.

For now, assuming that we have this property. It is then not hard to see that a parallel repetition of this game reduces the winning probability to negligible.[1]Using this new monogamy-of-entanglement property, we are able to prove security of copy-protection of point functions w.r.t to the product distribution as well as the identical distribution. Our observation here is that the challenge inputs pair (y,y) corresponds to a description of the challenge basis for the monogamy-of-entanglement with identical basis game: in particular, let $y \coloneqq y_0 \dots y_n$, each y_i describes the challenge basis for the i-th instance of the monogamy game: if $y_i = 0$, it is the computational basis (corresponding to the coset space $A_i + s_i$), otherwise, it is the Hadamard basis (corresponding to the coset space $A_i^\perp + s_i'$). The final step in the proof is to show that, if there exists an adversary that wins the anti-piracy game with challenge input (y,y), we can construct two

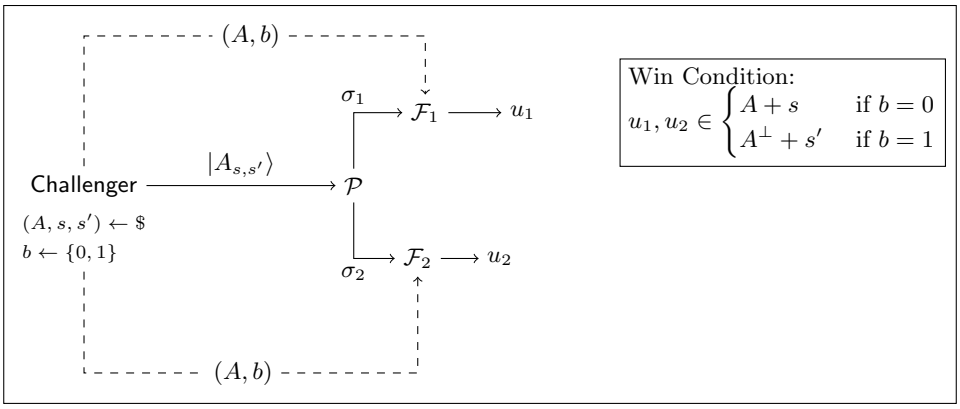

Fig. 1. Monogamy-of-Entanglement Game with Identical Basis (Coset Version)

non-communicating extractors that output n vectors $(v_i, w_i)_{i\in[1,n]}$ satisfying that v_i, w_i both belong to the same challenge coset space for all $i \in [1, n]$. This step can be done by using extracting lockable obfuscation technique from [CLLZ21]. .

Proof of the new monogamy-of-entanglement game. We describe the proof of the *BB84 version* of our new monogamy-of-entanglement game, as the coset version reduces to this game as proven in [CV22]. In the BB84 version, the challenger sends n BB84 states $\bigotimes_{i=1}^{n} |x_i\rangle^{\theta_i}$ to the pirate, and the freeloaders are given the basis θ and a random bit b. To win the game, the freeloaders both need to output a bitstring x^* such that x^* is equal to x on all the indices i such that $\theta_i = b$. This proof uses the template of [CV22] and can described in three steps. We refer the reader to Section 3 for the formal proof.

1. In the first step, we define the *extended non-local game* [JMRW16] associated to this monogamy-of-entanglement game. This game is between a challenger and two players. The players start by preparing a tripartite quantum state ρ_{012}; each of them keep one register, and they send the last one, say ρ_2, to the challenger. After this point, the players are not allowed to communicate. The challenger samples n BB84 basis $\theta \in \{0, 1\}^n$ at random, then measures each qubit $\rho_{C,i}$ of ρ_C in the corresponding basis θ_i; let x denote the outcome. Finally, the challenger sends θ, as well as a random bit b, to the two players. Each player is asked to output a bitstring x^* such that x^* is equal to x on all the indices i such that $\theta_i = b$. We show that the largest winning probability of the monogamy game is the same as the one of this extended non-local game. In this step, we use a technique from [TFKW13] to bound this winning probability.
2. In the second step, we express any strategy for this extended non-local game with security parameter $n \in \mathbb{N}$ as a tripartite quantum state ρ_{012} as well as two families of projective measurements, $\{B^{\theta,b}\}$ and $\{C^{\theta,b}\}$, both indexed by $\theta \in \Theta_n$ and $b \in \{0, 1\}$. We define the projector $\Pi_{\theta,b} = \sum_{x\in\{0,1\}^n} |x\rangle\langle x|^{\theta} \otimes B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}$ such that the winning probability of this strategy is $p_{win} = \mathbb{E}_{\theta,b} [\mathsf{Tr} (\Pi_{\theta,b}\ \rho_{012})]$. Then, we show the following upper-bound:

$$\begin{aligned} p_{win} &\leq \frac{1}{2N} \sum_{\substack{1\leq k\leq N \\ \alpha\in\{0,1\}}} \max_{\theta,b} \left\| \Pi_{\theta,b} \Pi_{\pi_{k,\alpha}(\theta\|b)} \right\| \\ &= \frac{1}{2} + \frac{1}{2N} \sum_{1\leq k\leq N} \max_{\theta,b} \left\| \Pi_{\theta,b} \Pi_{\pi_{k,1}(\theta\|b)} \right\| \end{aligned}$$

[1] We provide a proof for the parallel version of this game in Section 3.6

where $\{\pi_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ is a mutually orthogonal family of permutations to be defined later in the proof. We want the maximum in the equation above to be as small as possible. The goal of step 3 is to find such a family.

3. In the third step, we show that, as long as $b' \neq b$, the quantity $\|\Pi_{\theta,b}\ \Pi_{\theta',b'}\|$ depends on the number of indices on which θ and θ_i differ. More precisely, $\|\Pi_{\theta,b}\ \Pi_{\theta',b'}\|$ is upper-bounded by $2^{-d(\theta)/4}$, where $d(\theta)$ is the number of such indices. Thus, we choose our family of permutations such that, for all k, the last bit of $\pi_{k,1}(\theta,b)$ is $1-b$ and $d(\theta)$ is constant. We build upon a result of [CV22] to construct a family of permutations with the aforementioned properties. More concretely, [CV22] define a mutually orthogonal family of permutations π_k, indexed by $1 \geq k \geq N$, with the latter property. We then define another family $\widetilde{\pi}_{k,b}$, indexed by $1 \geq k \geq N$ and $b \in \{0,1\}$, where $\widetilde{\pi}_{k,b}(\theta,b) = \pi_k(\theta)\|1-b$. It is easy to see that this new family of permutations has both the former and the latter properties, and we prove that it is also a mutually orthogonal family.

Acknowledgements

This work was supported in part by the French ANR projects CryptiQ (ANR-18-CE39-0015) and SecNISQ (ANR-21-CE47-0014).

2 Preliminaries

2.1 Notations

Throughout this paper, λ denotes the security parameter. The notation $\mathsf{negl}(\lambda)$ denotes any function f such that $f(\lambda) = \lambda^{-\omega(1)}$, and $\mathsf{poly}(\lambda)$ denotes any function f such that $f(\lambda) = \mathcal{O}(\lambda^c)$ for some $c > 0$. When sampling uniformly at random a value a from a set $\mathcal{U}$, we employ the notation $a \leftarrow\!\!\$\, \mathcal{U}$. When sampling a value a from a probabilistic algorithm $\mathcal{A}$, we employ the notation $a \leftarrow \mathcal{A}$. By PPT we mean a polynomial-time non-uniform family of probabilistic circuits, and by QPT we mean a polynomial-time family of quantum circuits.

2.2 Coset States

Given a subspace $A \subset \mathbb{F}_2^n$ of dimension $n/2$ and a pair of vectors $(s,s') \in \mathbb{F}_2^n$, the coset state $|A_{s,s'}\rangle$ is defined as

$$|A_{s,s'}\rangle := \frac{1}{\sqrt{2^{n/2}}} \sum_{a\in A} (-1)^{a\cdot s'} |a+s\rangle$$

where $a \cdot s'$ denotes the inner product between a and s'.

In particular, a coset state is such that $\mathsf{H}^{\otimes n}\,|A_{s,s'}\rangle = |A^{\perp}_{s',s}\rangle$, where $A^{\perp}$ is the complement of A, i.e. $A^{\perp} := \{u \in \mathbb{F}_2^n \mid u \cdot v = 0\ \forall v \in A\}$.

Canonical representation. As the canonical representation of a coset $A+s$, we use the lexicographically smallest vector of the the coset; and for $u \in \mathbb{F}_2^n$, we note $\mathsf{Can}_A(u)$ the function that returns the canonical representation (also noted coset representative) of $A+u$. We note that if $u \in A+s$, then $\mathsf{Can}_A(u) = \mathsf{Can}_A(s)$. Also, the function $\mathsf{Can}_A(\cdot)$ is efficiently computable given a description of A.

2.3 Indistinguishable Obfuscation

Definition 1 (Indistinguishability Obfuscator [BGI+01]). *A uniform* PPT *machine* iO *is called an indistinguishability obfuscator for a classical circuit class* $\{\mathcal{C}_\lambda\}$ *if the following conditions are satisfied:*

- *For all security parameters $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all input x, we have that*

$$\Pr[C'(x) = C(x) \mid C' \leftarrow \mathsf{iO}(\lambda, C)] = 1.$$

- *For any (not necessarily uniform) distinguisher $\mathcal{D}$, for all security parameters $\lambda \in \mathbb{N}$, for all pairs of circuits $C_0, C_1 \in \mathcal{C}_\lambda$, we have that if $C_0(x) = C_1(x)$ for all inputs x, then*

$$\mathcal{A}^{\mathsf{iO}}(\lambda, \mathcal{A}) \coloneqq |\Pr[\mathcal{D}(\mathsf{iO}(\lambda, C_0)) = 1] - \Pr[\mathcal{D}(\mathsf{iO}(\lambda, C_1)) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that iO is δ-secure, for some concrete negligible function $\delta(\lambda)$, if for all QPT adversaries $\mathcal{A}$, the advantage $\mathcal{A}^{\mathsf{iO}}(\lambda, \mathcal{A})$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

2.4 Compute-and-Compare Obfuscation

This subsection is taken verbatim from [CHV23].

Definition 2 (Compute-and-Compare Programs). *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 $m \in \{0,1\}^{\ell_{\mathsf{msg}}}$, we define the compute-and-compare program:*

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

Definition 3 (Unpredictable Distribution). *Let $\mathcal{D} \coloneqq \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ be a distribution over pairs of the form $(\mathsf{CC}[f, y, m], \mathsf{aux})$ where $\mathsf{CC}[f, y, m]$ is a compute-and-compare program and aux is some (possibly quantum) auxiliary information. We say that $\mathcal{D}$ is an* unpredictable distribution *if for all QPT algorithm $\mathcal{A}$, we have that*

$$\Pr\big[\mathcal{A}(1^\lambda, f, \mathsf{aux}) = y \,:\, (\mathsf{CC}[f, y, m], \mathsf{aux}) \leftarrow \mathcal{D}_\lambda\big] \leq \mathsf{negl}(\lambda).$$

Definition 4 (Compute-and-Compare Obfuscator). *A PPT algorithm $\mathsf{CC\text{-}Obf}$ is said to be a compute-and-compare obfuscator for a family of unpredictable distributions $\mathcal{D} \coloneqq \{\mathcal{D}_\lambda\}$ if for all $\lambda \in \mathbb{N}$:*

- $\mathsf{CC\text{-}Obf}$ *is functionality preserving: for all x*

$$\Pr\big[\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}[f, y, m])(x) = \mathsf{CC}[f, y, m](x)\big] \geq 1 - \mathsf{negl}(\lambda)$$

- $\mathsf{CC\text{-}Obf}$ *has distributional indistinguishability: there exists a QPT simulator $\mathcal{S}$ such that*

$$\big\{\mathsf{CC\text{-}Obf}(1^\lambda, C), \mathsf{aux}\big\} \approx_c \big\{\mathcal{S}(1^\lambda, C.\mathsf{param}), \mathsf{aux}\big\},$$

  where $(C, \mathsf{aux}) \leftarrow \mathcal{D}_\lambda$.

2.5 Pseudorandom functions

This subsection is adapted from [CHV23,CLLZ21]. A pseudorandom function [GGM84] consists of a keyed function PRF and a set of keys $\mathcal{K}$ such that for a randomly chosen key $\mathsf{k} \in \mathcal{K}$, the output of the function $\mathsf{PRF}(\mathsf{k}, x)$ for any input x in the input space $\mathcal{X}$ "looks" random to a QPT adversary, even when given a polynomially many evaluations of $\mathsf{PRF}(\mathsf{k}, \cdot)$. Puncturable pseudorandom functions have an additional property that some keys can be generated *punctured* at some point, so that they allow to evaluate the pseudorandom function at all points except for the punctured points. Furthermore, even with the punctured key, the pseudorandom function evaluation at a punctured point still looks random.

Punctured pseudorandom functions are originally introduced in [BW13,BGI14,KPTZ13], who observed that it is possible to construct such puncturable pseudorandom functions for the construction from [GGM84], which can be based on any one-way function [HILL99].

Definition 5 (Puncturable Pseudorandom Function). *A pseudorandom function* $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *is a* puncturable pseudorandom function *if there is an addition key space* $\mathcal{K}_p$ *and three* PPT *algorithms* $\mathsf{PRF} = \langle \mathsf{KeyGen}, \mathsf{Puncture}, \mathsf{Eval} \rangle$ *such that:*

- $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$. *The key generation algorithm* KeyGen *takes the security parameter* 1^λ *as input and outputs a random key* $\mathsf{k} \in \mathcal{K}$.
- $\mathsf{k}\{x\} \leftarrow \mathsf{Puncture}(\mathsf{k}, x)$. *The puncturing algorithm* Puncture *takes as input a pseudorandom function key* $\mathsf{k} \in \mathcal{K}$ *and* $x \in \mathcal{X}$*, and outputs a key* $\mathsf{k}\{x\} \in \mathcal{K}_p$.
- $y \leftarrow \mathsf{Eval}(\mathsf{k}\{x\}, x')$. *The evaluation algorithm takes as input a punctured key* $\mathsf{k}\{x\} \in \mathcal{K}_p$ *and* $x' \in \mathcal{X}$*, and outputs a classical string* $y \in \mathcal{Y}$.

  We require the following properties of PRF.

- ***Functionality preserved under puncturing.*** *For all* $\lambda \in \mathbb{N}$*, for all* $x \in \mathcal{X}$*,*

$$\Pr\left[\forall x' \in \mathcal{X} \setminus \{x\} : \mathsf{Eval}(\mathsf{k}\{x\}, x') = \mathsf{Eval}(\mathsf{k}, x') \;\middle|\; \begin{matrix} \mathsf{k} \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \mathsf{k}\{x\} \leftarrow\!\!\$\ \mathsf{Puncture}(\mathsf{k}, x) \end{matrix}\right] = 1.$$

- ***Pseudorandom at punctured points.*** *For every* QPT *adversary* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$*, and every* $\lambda \in \mathbb{N}$*, the following holds:*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}_2(\mathsf{k}\{x^*\}, y, \tau) \;\middle|\; \begin{matrix} (x^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda, \tau) \\ \mathsf{k} \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \mathsf{k}\{x^*\} \leftarrow\!\!\$\ \mathsf{Puncture}(\mathsf{k}, x^*) \\ y \leftarrow \mathsf{Eval}(\mathsf{k}, x^*) \end{matrix}\right]\right.$$
$$\left. - \Pr\left[1 \leftarrow \mathcal{A}_2(\mathsf{k}\{x^*\}, y, \tau) \;\middle|\; \begin{matrix} (x^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda, \tau) \\ \mathsf{k} \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda) \\ \mathsf{k}\{x^*\} \leftarrow\!\!\$\ \mathsf{Puncture}(\mathsf{k}, x^*) \\ y \leftarrow\!\!\$\ \mathcal{Y} \end{matrix}\right]\right| \leq \mathsf{negl}(\lambda),$$

  where the probability is taken over the randomness of KeyGen, Puncture, *and* $\mathcal{A}_1$.

Denote the above probability as $\mathcal{A}^{\mathsf{PRF}}(\lambda, \mathcal{A})$*. We further say that* PRF *is* δ*-secure, for some concrete negligible function* $\delta(\lambda)$*, if for all* QPT *adversaries* $\mathcal{A}$*, the advantage* $\mathcal{A}^{\mathsf{PRF}}(\lambda, \mathcal{A})$ *is smaller than* $\delta(\lambda)^{\Omega(1)}$.

Definition 6 (Statistically injective pseudorandom function). *A family of statistically injective (puncturable) pseudorandom functions with (negligible) failure probability* $\varepsilon(\cdot)$ *is a (puncturable) pseudorandom functions family* PRF *such that with probability* $1 - \varepsilon(\lambda)$ *over the random choice of key* $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$*, we have that* $\mathsf{PRF}(\mathsf{k}, \cdot)$ *is injective.*

Definition 7 (Extracting pseudorandom function). *A family of extracting (puncturable) pseudorandom functions with error* $\varepsilon(\cdot)$ *for min-entropy* $k(\cdot)$ *is a (puncturable) pseudorandom functions family* PRF *mapping* $n(\lambda)$ *bits to* $m(\lambda)$ *bits such that for all* $\lambda \in \mathbb{N}$*, if* X *is any distribution over* $n(\lambda)$ *bits with min-entropy greater than* $k(\lambda)$*, then the statistical distance between* $(\mathsf{k}, \mathsf{PRF}(\mathsf{k}, X))$ *and* $(\mathsf{k}, r \leftarrow \{0,1\}^{m(\lambda)})$ *is at most* $\varepsilon(\cdot)$*, where* $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$.

3 A New Monogamy-of-Entanglement Game for Coset States

In this section, we present a new monogamy-of-entanglement game for coset states and prove an upper-bound on the probability of winning this game. Along the way, we present a BB84 version of this game with the same upper-bound.

3.1 The Coset Version

Definition 8 (Monogamy-of-Entanglement Game with Identical Basis (Coset Version)). *This game is between a challenger and a triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ - where $\mathcal{A}_1$ and $\mathcal{A}_2$ are not communicating, and is parametrized by a security parameter λ.*

- *The challenger samples a subspace $A \leftarrow \{0,1\}^{\lambda \times \frac{\lambda}{2}}$ and two vectors $(s, s') \leftarrow \mathbb{F}_2^n \times \mathbb{F}_2^n$. Then the challenger prepares the coset state $|A_{s,s'}\rangle$ and sends $|A_{s,s'}\rangle$ to $\mathcal{A}_0$.*
- *$\mathcal{A}_0$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.*
- *The challenger samples $b \leftarrow \{0,1\}$, then sends (A, b) to both $\mathcal{A}_1$ and $\mathcal{A}_2$.*
- *$\mathcal{A}_1$ returns u_1 and $\mathcal{A}_2$ returns u_2.*

For $i \in \{1,2\}$, we say that $\mathcal{A}_i$ makes a correct guess if $(b = 0 \ \wedge \ u_i' \in A + s)$ or if $(b = 1 \ \wedge \ u_i' \in A^\perp + s')$. We say that $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ win the game if both $\mathcal{A}_1$ and $\mathcal{A}_2$ makes a correct guess. For any triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and any security parameter $\lambda \in \mathbb{N}$ for this game, we note $\mathsf{MoE}^\lambda_{coset}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ the random variable indicating whether $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ win the game or not.

We note that there is a trivial way for a triple of adversaries to win this game with probability $1/2$, by applying the following strategy. $\mathcal{A}_0$ samples a random bit b^*. $\mathcal{A}_0$ measures $|A_{s,s'}\rangle$ in the computational basis if $b^* = 0$, or in the Hadamard basis if $b^* = 1$. In both cases, $\mathcal{A}_0$ sends the outcome u to both $\mathcal{A}_1$ and $\mathcal{A}_2$. Regardless of the value of A and b, $\mathcal{A}_1$ and $\mathcal{A}_2$ both return u. Because when $b^* = b$ (which happens with probability $1/2$), the outcome of the measurement is a vector of the expected coset space, the adversaries win the game with probability $1/2$. In the rest of this section we prove that no triple of adversaries can actually win the game with a probability significantly greater than $1/2$.

Theorem 1. *There exists a negligible function $\mathsf{negl}(\cdot)$ such that, for any triple of algorithms $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and any security parameter $\lambda \in \mathbb{N}$, $\Pr\left[\mathsf{MoE}^\lambda_{coset}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda)$.*

The proof of this theorem is given in subsequent sections.

3.2 The BB84 Version

We introduce below the BB84 version of this game. We show in the following that it is sufficient to study the BB84 version (which is simpler) to prove Theorem 1, as any triple of adversaries for the BB84 version can be turned into a triple of adversaries for the coset version without changing the probability of winning.

Notations. Through all Section 3.2 and Section 3.3, we use the following notations. Let $n \in \mathbb{N}$, we note $\Theta_n := \{\theta \in \{0,1\}^n \ : \ |\theta| = n/2\}$ - where $|\cdot|$ denotes the Hamming weight - and $N := \binom{n}{n/2}$. Thus Θ_λ has exactly N elements.

Definition 9 (Monogamy-of-Entanglement Game with Identical Basis (BB84 Version)). *This game is between a challenger and a triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ - where $\mathcal{A}_1$ and $\mathcal{A}_2$ are non-communicating, and is parametrized by a security parameter λ.*

- *The challenger samples $x \leftarrow \{0,1\}^\lambda$ and $\theta \leftarrow \Theta_\lambda$. Then the challenger prepares the state $|x^\theta\rangle := \bigotimes_{i \in [\![1,\lambda]\!]} \mathsf{H}^{\theta_i} |x_i\rangle$ and sends $|x^\theta\rangle$ to $\mathcal{A}_0$.*
- *$\mathcal{A}_0$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.*
- *The challenger samples $b \leftarrow \{0,1\}$, then sends (θ, b) to both $\mathcal{A}_1$ and $\mathcal{A}_2$.*
- *$\mathcal{A}_1$ returns x_1 and $\mathcal{A}_2$ returns x_2.*

Let $x_{T_b} := \{x_i \mid \theta_i = b\}$. We say that $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ win the game if $x_1 = x_2 = x_{T_b}$. For any triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and any security parameter $\lambda \in \mathbb{N}$ for this game, we note $\mathsf{MoE}^{\lambda}_{BB84}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ the random variable indicating whether $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ win the game or not.

We note that the trivial strategy for the coset version can be easily adapted for the BB84 one. Hence the greatest probability of winning this game is also lower bounded by 1/2.

Theorem 2. *There exists a negligible function $\mathsf{negl}(\cdot)$ such that, for any triple of algorithms $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and any security parameter $\lambda \in \mathbb{N}$, $\Pr\left[\mathsf{MoE}^{\lambda}_{BB84}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda)$.*

Proof of Theorem 1 follows similarly as that of [CV22], in which the winning probability of cloning adversaries in the monogamy-of-entanglement game of coset states reduces to the winning probability of the adversaries in the game of BB84 states. We thus provide the proof of Theorem 2 below.

3.3 Proof of Theorem 2

This proof follows the same structure as [CV22]. We can separate the proof in four main steps.

1. In the first step, we define the *extended non-local game* [JMRW16] associated the monogamy-of-entanglement game (BB84 version), and show that the greatest winning probability of the monogamy game is the same as the one of this extended non-local game. This step allows us to use a technique from [TFKW13] to bound the winning probability.
2. In the second step, we express any strategy for this extended non-local game with security parameter $n \in \mathbb{N}$ as a tripartite quantum state ρ_{012} as well as two families of projective measurements, $\{B^{\theta,b}\}$ and $\{C^{\theta,b}\}$, both indexed by $\theta \in \Theta_n$ and $b \in \{0,1\}$. We define the projector $\Pi_{\theta,b} = \sum_{x\in\{0,1\}^n} |x\rangle\langle x|^{\theta} \otimes B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}$ such that the winning probability of this strategy is $p_{win} = \mathbb{E}_{\theta,b}\left[\mathsf{Tr}\left(\Pi_{\theta,b}\ \rho_{012}\right)\right]$. Then, we show the following upper-bound:

$$\begin{aligned} p_{win} &\leq \frac{1}{2N} \sum_{\substack{1\leq k\leq N \\ \alpha\in\{0,1\}}} \max_{\theta,b} \left\|\Pi_{\theta,b}\Pi_{\pi_{k,\alpha}(\theta\|b)}\right\| \\ &= \frac{1}{2} + \frac{1}{2N} \sum_{1\leq k\leq N} \max_{\theta,b} \left\|\Pi_{\theta,b}\Pi_{\pi_{k,1}(\theta\|b)}\right\| \end{aligned}$$

   where $\{\pi_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ is a family of permutations to be defined later in the proof.
3. In the third step, we show that the quantity $\|\Pi_{\theta,b}\ \Pi_{\theta',b'}\|$ is upper-bounded by a small quantity as long as $b' \neq b$.
4. Finally, in the fourth step, we show that there exists a family of permutations such that, when $\alpha = 0$, $\pi_{k,\alpha}(\theta, b) = (\theta', b')$ for some θ' and $b' \neq b$, and conclude the proof.

Step 1: extended non-local game. We define the following extended non-local game, and show that any triple of adversaries that win the monogamy-of-entanglement game with same basis (BB84 version) with probability p can be turned into another triple of adversaries that win this extended non-local game with the same probability p.

Definition 10 (Extended non-local game). *This game is between a challenger and two adversaries $\mathcal{A}_0$ and $\mathcal{A}_1$, and is parametrized by a security parameter λ.*

- *$\mathcal{A}_1$ and $\mathcal{A}_2$ jointly prepare a quantum state ρ_{012} - where ρ_0 is a λ-qubits quantum state, then send ρ_0 to the challenger. $\mathcal{A}_1$ and $\mathcal{A}_2$ keep ρ_1 and ρ_2 respectively. From this step $\mathcal{A}_1$ and $\mathcal{A}_2$ cannot communicate.*

- *The challenger samples $\theta \leftarrow \Theta_n$ and $b \leftarrow \{0,1\}$. Then, for all $i \in [\![1,\lambda]\!]$, the challenger measures the i^{th} qubit of ρ_0 in computational basis if $\theta_i = 0$ or in Hadamard basis if $\theta_i = 1$. Let $m \in \{0,1\}^n$ denote the measurement outcome. Finally, the challenger sends (θ, b) to $\mathcal{A}_1$ and $\mathcal{A}_2$.*
- *$\mathcal{A}_1$ returns m_1 and $\mathcal{A}_2$ returns m_2.*

Let $m_{T_b} := \{m_i \mid \theta_i = b\}$. We say that $(\mathcal{A}_1, \mathcal{A}_2)$ win the game if $m_1 = m_2 = m_{T_b}$.

Lemma 1. *Let $n \in \mathbb{N}$ and $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ a triple of adversaries for the monogamy-of-entanglement game (Definition 9) parametrized by n, that win with probability p_n. Then there exists a quantum state ρ_{012} and a pair of adversaries $(\mathcal{A}'_1, \mathcal{A}'_2)$ for the extended non-local game (Definition 10) that win with the same probability p_n.*

Proof. Consider a triple of adversaries for the monogamy-of-entanglement game (Definition 9), parametrized by $n \in \mathbb{N}$, that win with probability p_n. We can model these adversaries as a CPTP map $\Phi : \mathcal{H}_0 \to \mathcal{H}_1 \times \mathcal{H}_2$, and POVMs families $\{B^{\theta,b}\}$ and $\{C^{\theta,b}\}$, both indexed by $\theta \in \Theta_n$ and $b \in \{0,1\}$. Then we have

$$p_n = \mathop{\mathbb{E}}_{\substack{\theta\in\Theta_n \\ b\in\{0,1\}}} \mathop{\mathbb{E}}_{x\in\{0,1\}^n} Tr\left[(B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}})\Phi(|x^\theta\rangle\langle x^\theta|)\right].$$

The strategy for the extended non-local game is as follows. $\mathcal{A}_1$ and $\mathcal{A}_2$ prepare the bipartite state $\rho_{00'} = \bigotimes_{1\leq i\leq n} |\phi^+\rangle\langle\phi^+|$ where ϕ^+ denotes the EPR state $(|00\rangle + |11\rangle)/\sqrt{2}$, and where ρ_0 (resp. $\rho_{0'}$) is composed of the first halves (resp. second halves) of these EPR states. Then, they apply Φ to $\rho_{0'}$. Let ρ_{012} denotes the resulting state. They send ρ_0 to the challenger, $\mathcal{A}_1$ keeps ρ_1 and $\mathcal{A}_2$ keeps ρ_2. Later, when $\mathcal{A}_1$ receives (θ, b), from the challenger, $\mathcal{A}_1$ applies the POVM $B^{\theta,b}$ to ρ_1 and returns the outcome. $\mathcal{A}_2$ does the same with POVM $C^{\theta,b}$ and ρ_2. The probability of winning of such strategy is then

$$p'_n = \mathop{\mathbb{E}}_{\substack{\theta\in\Theta_n \\ b\in\{0,1\}}} \sum_{x\in\{0,1\}^n} Tr\left[\left(|x^\theta\rangle\langle x^\theta| \otimes B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)\rho_{012}\right]. \tag{1}$$

We do the following calculation.

$$\begin{aligned}
Tr\left[\left(|x^\theta\rangle\langle x^\theta| \otimes B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)\rho_{012}\right] &= \frac{1}{2^n} \sum_{r,r'\in\{0,1\}^n} Tr\left[\left(|x^\theta\rangle\langle x^\theta| \otimes B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)(|r\rangle\langle r'| \otimes \Phi(|r\rangle\langle r'|))\right] \\
&= \frac{1}{2^n} \sum_{r,r'\in\{0,1\}^n} \langle r|x^\theta\rangle \langle x^\theta|r'\rangle\, Tr\left[\left(B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)\Phi(|r\rangle\langle r'|)\right] \\
&= \frac{1}{2^n} \sum_{r,r'\in\{0,1\}^n} Tr\left[\left(B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)\Phi\left(|r\rangle \langle r|x^\theta\rangle \langle x^\theta|r'\rangle \langle r'|\right)\right] \\
&= \frac{1}{2^n} Tr\left[\left(B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)\Phi\left(\frac{1}{2^n}\sum_{r\in\{0,1\}^n} |r\rangle\langle r|\ |x^\theta\rangle\langle x^\theta|\frac{1}{2^n}\sum_{r'\in\{0,1\}^n} |r'\rangle\langle r'|\right)\right] \\
&= \frac{1}{2^n} Tr\left[\left(B^{\theta,b}_{x_{T_b}} \otimes C^{\theta,b}_{x_{T_b}}\right)\Phi\left(|x^\theta\rangle\langle x^\theta|\right)\right]
\end{aligned}$$

By plugging this result into Equation (1), we get $p'_n = p_n$, which concludes the proof.

□

Step 2: first upper-bound of the winning probability. We prove an upper-bound for the extended non-local game above. We need the following lemma.

Lemma 2 (Lemma 2 of [TFKW13]). *Let $\Pi_1, \ldots, \Pi_n$ be projective positive semi-definite operators on a Hilbert space, and $\{\pi_i\}_{i\in[\![1,n]\!]}$ be a set of orthogonal permutations for some integer n. Then*

$$\left\|\sum_{i=1}^n \Pi_i\right\| \leq \sum_{i=1}^n \max_{j\in[\![1,n]\!]} \left\|\Pi_j \Pi_{\pi_i(j)}\right\|$$

Let $\left(\{B^{\theta,b}\}_{\theta\in\Theta_n,b\in\{0,1\}},\{C^{\theta,b}\}_{\theta\in\Theta_n,b\in\{0,1\}},\rho_{012}\right)$ be a strategy for the extended non-local game. Using Naimark's dilation theorem, we can assume without loss of generality that the $B^{\theta,b}$ and $C^{\theta,b}$ are all projective. Let $\Pi_{\theta,b}$ be the following projector: $\Pi_{\theta,b}:=\sum_{x\in\{0,1\}^n}|x\rangle\langle x|^\theta\otimes B^{\theta,b}_{x_{T_b}}\otimes C^{\theta,b}_{x_{T_b}}$. Then the winning probability of this strategy is

$$\begin{aligned}
p_{win} &= \mathop{\mathbb{E}}_{\theta\in\Theta_n,b\in\{0,1\}}\mathsf{Tr}\left(\Pi_{\theta,b}\ \rho_{012}\right)\\
&\leq \mathop{\mathbb{E}}_{\theta\in\Theta_n,b\in\{0,1\}}\|\Pi_{\theta,b}\|\\
&\leq \frac{1}{2N}\sum_{\substack{1\leq k\leq N\\ \alpha\in\{0,1\}}}\max_{\theta,b}\left\|\Pi_{\theta,b}\Pi_{\pi_{k,\alpha}(\theta,b)}\right\| \qquad (2)
\end{aligned}$$

where the first inequality follows from the definition of the norm and the second from Lemma 2; and where $\{\pi_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ is a family of mutually orthogonal permutations.

Step 3: upper-bound of $\|\Pi_{\theta,b}\Pi_{\theta',1-b}\|$. In this part, we show that for all $(\theta,\theta')\in\Theta_n$ and all $b\in\{0,1\}$, we can upper-bound $\|\Pi_{\theta,b}\Pi_{\theta',1-b}\|$ by a small quantity.

Let $(\theta,\theta')\in\Theta_n^2$ and $b\in\{0,1\}$. Note $R:=\{i\in[\![1,N]\!]\ :\ \theta_i\neq\theta'_i\}$, $T:=\{i\in[\![1,N]\!]\ :\ \theta_i=b\}$, $T':=\{i\in[\![1,N]\!]\ :\ \theta'_i=1-b\}$ and $S:=\{i\in R\ :\ \theta_i=b \text{ and } \theta'_i=1-b\}$. We define $\bar{\mathrm{P}}$ and $\bar{\mathrm{Q}}$ as follows:

$$\begin{aligned}
\bar{\mathrm{P}} &:= \sum_{x_T\in\{0,1\}^T}\mathsf{H}^b|x_S\rangle\langle x_S|\mathsf{H}^b\otimes\mathbb{I}_{\bar{S}}\otimes B^{\theta,b}_{x_T}\otimes\mathbb{I}_C\\
\bar{\mathrm{Q}} &:= \sum_{x_{T'}\in\{0,1\}^{T'}}\mathsf{H}^{1-b}|x_S\rangle\langle x_S|\mathsf{H}^{1-b}\otimes\mathbb{I}_{\bar{S}}\otimes C^{\theta',1-b}_{x_{T'}}\otimes\mathbb{I}_B
\end{aligned}$$

where $|x_S\rangle\langle x_S|$ denotes the subsystem of $|x_T\rangle\langle x_T|$ whose indices belong to S, and $\mathbb{I}_{\bar{S}}$ denotes the rest of the system.

Remark that we have:

$$\begin{aligned}
\|\Pi_{\theta,b}\Pi_{\theta',1-b}\|^2 &= \|\Pi_{\theta',1-b}\Pi_{\theta,b}\Pi_{\theta',1-b}\|\\
&\leq \|\Pi_{\theta',1-b}\bar{\mathrm{P}}\Pi_{\theta',1-b}\|\\
&= \|\bar{\mathrm{P}}\Pi_{\theta',1-b}\bar{\mathrm{P}}\|\\
&\leq \bar{\mathrm{P}}\bar{\mathrm{Q}}\bar{\mathrm{P}}
\end{aligned}$$

where we have the first line because $\Pi_{\theta,b}$ is a projection, the second because $\Pi_{\theta,b}\leq\bar{\mathrm{P}}$, the third because $\Pi_{\theta,b}$ and $\bar{\mathrm{P}}$ are projections and the last because $\Pi_{\theta',1-b}\leq\bar{\mathrm{Q}}$.

Consider now the quantity $\bar{\mathrm{P}}\bar{\mathrm{Q}}\bar{\mathrm{P}}$. We compute the following upper-bound for $\bar{\mathrm{P}}\bar{\mathrm{Q}}\bar{\mathrm{P}}$:

$$\begin{aligned}
\bar{\mathrm{P}}\bar{\mathrm{Q}}\bar{\mathrm{P}} &= \sum_{\substack{x_T,z_T\in\{0,1\}^T\\ y_{T'}\in\{0,1\}^{T'}}}\mathsf{H}^b|x_S\rangle\langle x_S|\mathsf{H}^b\mathsf{H}^{1-b}|y_S\rangle\langle y_S|\mathsf{H}^{1-b}\mathsf{H}^b|z_S\rangle\langle z_S|\mathsf{H}^b\otimes\mathbb{I}_{\bar{S}}\otimes B^{\theta,b}_{x_T}\ B^{\theta,b}_{z_T}\otimes C^{\theta',1-b}_{y_{T'}}\\
&= \sum_{\substack{x_T\in\{0,1\}^T\\ y_{T'}\in\{0,1\}^{T'}}}\mathsf{H}^b|x_S\rangle\langle x_S|\mathsf{H}^b\mathsf{H}^{1-b}|y_S\rangle\langle y_S|\mathsf{H}^{1-b}\mathsf{H}^b|x_S\rangle\langle x_S|\mathsf{H}^b\otimes\mathbb{I}_{\bar{S}}\otimes B^{\theta,b}_{x_T}\otimes C^{\theta',1-b}_{y_{T'}}\\
&= 2^{-|S|}\sum_{\substack{x_T\in\{0,1\}^T\\ y_{T'}\in\{0,1\}^{T'}}}\mathsf{H}^b|x_S\rangle\langle x_S|\mathsf{H}^b\otimes\mathbb{I}_{\bar{S}}\otimes B^{\theta,b}_{x_T}\otimes C^{\theta',1-b}_{y_{T'}}\\
&= 2^{-|S|}\sum_{x_T\in\{0,1\}^T}\mathsf{H}^b|x_S\rangle\langle x_S|\mathsf{H}^b\otimes\mathbb{I}_{\bar{S}}\otimes B^{\theta,b}_{x_T}\otimes\mathbb{I}_C
\end{aligned}$$

where the first equality comes from $B^{\theta,b}_{x_T}\, B^{\theta,b}_{z_T} = B^{\theta,b}_{x_T}$ if $x_T = z_T$ and 0 otherwise; the second comes from $\langle x_S|\,\mathsf{H}^b\mathsf{H}^{1-b}|y_S\rangle\langle y_S|\mathsf{H}^{1-b}\mathsf{H}^b\,|x_S\rangle = |\langle x_S|\,\mathsf{H}\,|y_S\rangle|^2 = 2^{-|S|}$ for all $x_T, y_{T'}$ and the third from $\sum_{y_{T'}} C^{\theta',1-b}_{y_{T'}} = \mathbb{I}_C$. Notice that we can assume without loss of generality that $|S|$ is larger than $|R|/2$: if it is not the case, we just swap the roles of θ and θ'. Thus, by linearity and from $\sum_{x_T} B^{\theta,b}_{x_T} = \mathbb{I}_B$, it comes $\|\bar{\mathrm{P}}\bar{\mathrm{Q}}\bar{\mathrm{P}}\| \leq 2^{-|S|} \leq 2^{-|R|/2}$ hence

$$\|\Pi_{\theta,b}\Pi_{\theta',1-b}\| \leq 2^{-|R|/4} \tag{3}$$

Remark 1. Remark that, when considering $\|\Pi_{\theta,b}\Pi_{\theta',b}\|$ instead, we have $S = \emptyset$. Thus, the reasoning above yields the trivial upper-bound

$$\|\Pi_{\theta,b}\Pi_{\theta',b}\| \leq 1 \tag{4}$$

Step 4: finding the permutation family. In this part, we construct a family of mutually orthogonal permutations $\{\pi_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ such for all $k \in [\![1, N]\!]$, $\pi_{k,0}$ "flips" the last input's bit and $\pi_{k,1}$ leaves it unchanged.

We use the following lemma, proven in [CV22].

Lemma 3 (Lemma 3.4 of [CV22]). *Let n be an even integer, $\Theta_n := \{\theta \in \{0,1\}^n \ : \ |\theta| = n/2\}$ and $N = \binom{n}{n/2}$. Then there is a family of N mutually orthogonal permutations $\{\tilde{\pi}_k\}_{k\in[\![1,N]\!]}$ of Θ_n such that the following holds. For each $i \in [\![1, n/2]\!]$, there are exactly $\binom{n/2}{i}^2$ permutations $\tilde{\pi}_k$ such that the number of positions at which θ and $\tilde{\pi}_k(\theta)$ are both 1 is $n/2 - i$.*

We prove the following corollary.

Corollary 1. *Let n be an even integer, $\Theta_n := \{\theta \in \{0,1\}^n \ : \ |\theta| = n/2\}$ and $N = \binom{n}{n/2}$. Then there is a family of $2N$ mutually orthogonal permutations $\{\pi_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ of $\Theta_n \times \{0,1\}$ such that the two following properties hold.*

- *For each $i \in [\![1, n/2]\!]$, there are exactly $\binom{n/2}{i}^2$ permutations $\pi_{k,0}$ such that the number of positions at which θ and θ' are both 1 is $n/2 - i$ (i.e. θ and θ' differ in $2i$ positions).*
- *If $\alpha = 0$, then $b' = 1 - b$. Otherwise, $b' = b$.*

where we use the notation $(\theta'\|b') := \pi_{k,\alpha}(\theta\|b)$.

Proof. Let $\{\tilde{\pi}_k\}_{k\in[\![1,N]\!]}$ be a family of orthogonal permutations promised in Lemma 3. Define the family $\{\pi_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ as follows. For all $k \in [\![1, N]\!]$:

$$\pi_{k,0}(\theta||b) = \tilde{\pi}_k(\theta)||(1-b)$$
$$\pi_{k,1}(\theta||b) = \tilde{\pi}_k(\theta)||b$$

The two properties follow directly by construction. It remains to prove that these $2N$ permutations are mutually orthogonal. Assume $\pi_{k,\alpha}(\theta) = \pi_{k',\alpha'}(\theta)$. Then we have $\alpha = \alpha'$, and $\tilde{\pi}_k(\theta) = \tilde{\pi}_{k'}(\theta)$, hence $k = k'$ because $\{\tilde{\pi}_k\}_k$ is a family of orthogonal permutations. □

Concluding the proof. We make use of the following lemma from [CV22].

Lemma 4 (Lemma 3.6 of [CV22]). *Let $n \geq 2$ an integer, and note $N = \binom{n}{n/2}$. Then we have*

$$\frac{1}{N}\sum_{i=0}^{n/2}\binom{n/2}{i}^2 2^{-i/2} \leq \sqrt{e}\left(\cos\frac{\pi}{8}\right)^n$$

The rest of the proof follows easily. We first rewrite Equation (2) as

$$p_{win} \leq \frac{1}{2N}\sum_{k=1}^{N} \max_{\theta,b} \left\|\Pi_{\theta,b}\Pi_{\pi_{k,1}(\theta,b)}\right\| + \frac{1}{2N}\sum_{k=1}^{N} \max_{\theta,b} \left\|\Pi_{\theta,b}\Pi_{\pi_{k,0}(\theta,b)}\right\|$$

Then, by plugging the permutations family of Corollary 1, and using the upper-bounds proved in Equation (3) and Equation (4), it comes

$$\begin{aligned} p_{win} &\leq \frac{1}{2} + \frac{1}{2N}\sum_{i=1}^{n/2} 2^{-i/2} \\ &\leq \frac{1}{2} + \frac{\sqrt{e}}{2}\left(\cos\frac{\pi}{8}\right)^n. \end{aligned}$$

3.4 Computational Version

We provide below a computational version of the monogamy-of-entanglement with identical basis. The only difference is that the adversaries are given access to obfuscated membership programs for the coset space and its dual. This game is still hard to win with probability significantly greater than $1/2$ if we make the assumption that the adversaries are polynomially bounded. The proof of this statement follows directly from the proof of hardness of the computational version of the regular monogamy-of-entanglement game [CLLZ21].

Definition 11 (Computational Monogamy-of-Entanglement Game with Identical Basis (Coset Version)). *This game is between a challenger and a triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ - where $\mathcal{A}_1$ and $\mathcal{A}_2$ are not communicating, and is parametrized by a security parameter λ.*

- *The challenger samples a subspace $A \leftarrow \{0,1\}^{\lambda \times \frac{\lambda}{2}}$ and two vectors $(s, s') \leftarrow \mathbb{F}_2^n \times \mathbb{F}_2^n$. Then the challenger prepares the coset state $|A_{s,s'}\rangle$ as well as two obfuscated membership programs $\widehat{\mathsf{P}}_{A+s} := \mathsf{iO}(A+s)$ and $\widehat{\mathsf{P}}_{A^\perp+s'} := \mathsf{iO}(A^\perp + s')$ and sends $\left(|A_{s,s'}\rangle, \widehat{\mathsf{P}}_{A+s}, \widehat{\mathsf{P}}_{A^\perp+s'}\right)$ to $\mathcal{A}_0$.*
- *$\mathcal{A}_0$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.*
- *The challenger samples $b \leftarrow \{0,1\}$, then sends (A, b) to both $\mathcal{A}_1$ and $\mathcal{A}_2$.*
- *$\mathcal{A}_1$ returns u_1 and $\mathcal{A}_2$ returns u_2.*

For $i \in \{1,2\}$, we say that $\mathcal{A}_i$ makes a correct guess if $(b = 0 \ \wedge \ u_i' \in A+s)$ or if $(b = 1 \ \wedge \ u_i' \in A^\perp + s')$. We say that $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ win the game if both $\mathcal{A}_1$ and $\mathcal{A}_2$ makes a correct guess. For any triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and any security parameter $\lambda \in \mathbb{N}$ for this game, we note $\mathsf{MoE}^{\lambda}_{coset(comp)}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ the random variable indicating whether $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ win the game or not.

Theorem 3. *There exists a negligible function $\mathsf{negl}(\cdot)$ such that, for any triple of QPT algorithms $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and any security parameter $\lambda \in \mathbb{N}$, $\Pr\left[\mathsf{MoE}^{\lambda}_{coset(comp)}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda)$.*

3.5 Parallel Repetition of the Game

For our proof of anti-piracy of copy-protection, we actually need a parallel version of this game, where the challenger samples $\kappa \in \mathbb{N}$ independent cosets and an independent basis choice for each coset; and the adversaries are suppose to return a vector in the correct space for all the cosets to win the game. We show that the winning probability of this game is negligible.

Definition 12 (κ-Parallel Computational Monogamy-of-Entanglement Game with Identical Basis (Coset Version)). *This game is between a challenger and a triple of adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ - where $\mathcal{A}_1$ and $\mathcal{A}_2$ are not communicating, and is parametrized by a security parameter λ.*

- *The challenger samples κ subspaces $\{A_i\}_{i\in[\![1,\kappa]\!]}$ and κ pairs of vectors $\{(s_i,s'_i)\}_{i\in[\![1,\kappa]\!]}$ where $A_i \leftarrow \{0,1\}^{\lambda\times\frac{\lambda}{2}}$ and $(s_i,s'_i) \leftarrow \mathbb{F}_2^n \times \mathbb{F}_2^n$ for all $i \in [\![1,\kappa]\!]$. Then the challenger prepares the coset states $\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\kappa]\!]}$ as well as the associated obfuscated membership programs $\widehat{\mathsf{P}}_{A_i+s_i} := \mathsf{iO}(A_i+s_i)$ and $\widehat{\mathsf{P}}_{A_i^\perp+s'_i} := \mathsf{iO}(A_i^\perp+s'_i)$ for $i\in[\![1,\kappa]\!]$; and sends $\left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\kappa]\!]}, \{\widehat{\mathsf{P}}_{A_i+s_i}, \widehat{\mathsf{P}}_{A_i^\perp+s'_i}\}_{i\in[\![1,\kappa]\!]}\right)$ to $\mathcal{A}_0$.*
- *$\mathcal{A}_0$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.*
- *The challenger samples $r \leftarrow \{0,1\}^\kappa$, then sends $\{A_i\}_{i\in[\![1,\kappa]\!]}$ and r to both $\mathcal{A}_1$ and $\mathcal{A}_2$.*
- *$\mathcal{A}_1$ returns κ vectors $\{u_i\}_{i\in[\![1,\kappa]\!]}$ and $\mathcal{A}_2$ returns κ vectors $\{u'_i\}_{i\in[\![1,\kappa]\!]}$.*

We say that $\mathcal{A}_1$ makes a correct guess if $(r_i = 0 \ \wedge \ u_i \in A_i + s_i)$ or if $(r_i = 1 \ \wedge \ u_i \in A_i^\perp + s'_i)$ for all $i \in [\![1,\kappa]\!]$. Similarly, we say that $\mathcal{A}_2$ makes a correct guess if $(r_i = 0 \ \wedge \ u'_i \in A_i + s_i)$ or if $(r_i = 1 \ \wedge \ u'_i \in A_i^\perp + s'_i)$ for all $i \in [\![1,\kappa]\!]$. We say that $(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2)$ win the game if both $\mathcal{A}_1$ and $\mathcal{A}_2$ makes a correct guess. For any triple of adversaries $(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2)$ and any security parameter $\lambda\in\mathbb{N}$ for this game, we note $\kappa - \mathsf{MoE}^\lambda_{coset(comp)}(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2)$ the random variable indicating whether $(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2)$ win the game or not.

Theorem 4. *There exists a negligible function $\mathsf{negl}(\cdot)$ such that, for any triple of QPT algorithms $(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2)$ and any security parameter $\lambda\in\mathbb{N}$, $\Pr\left[\kappa - \mathsf{MoE}^\lambda_{coset(comp)}(\mathcal{A}_0,\mathcal{A}_1,\mathcal{A}_2) = 1\right] \leq \mathsf{negl}(\lambda)$.*

3.6 Proof of Parallel Version of the Monogamy Game

In this subsection, we prove Theorem 4. We do it by proving that a parallel version of the BB84 version of the monogamy game has negligible security, as the coset version follows as for the single instance. As the proof follows the same structure as the one of Theorem 2, we only describe here the important steps of the proof.

Step 1: extended non-local game. We first describe the extended non-local game for this parallel version of the game. This game is between a challenger and two adversaries $\mathcal{A}_0$ and $\mathcal{A}_1$, and is parametrized by a security parameter λ and a number of repetitions $\kappa := \mathsf{poly}(\lambda)$.

- $\mathcal{A}_1$ and $\mathcal{A}_2$ jointly prepare a quantum state ρ_{012} - where ρ_0 is composed of κ λ-qubits registers, denoted as $\rho_0^1,\dots,\rho_0^\kappa$ - then send ρ_0 to the challenger. $\mathcal{A}_1$ and $\mathcal{A}_2$ keep ρ_1 and ρ_2 respectively. From this step $\mathcal{A}_1$ and $\mathcal{A}_2$ cannot communicate.
- For $j\in[\![1,\kappa]\!]$, the challenger samples $\theta^j \leftarrow \Theta_n$, then the challenger samples $r\leftarrow\{0,1\}^\kappa$. Then, for all $i\in[\![1,\lambda]\!]$ and $j\in[\![1,\kappa]\!]$, the challenger measures the i^{th} qubit of ρ_0^j in computational basis if $\theta_i^j = 0$ or in Hadamard basis if $\theta_i^j = 1$. Let $m^j\in\{0,1\}^n$ denote the measurement outcome for every j. Finally, the challenger sends $\theta := (\theta^1,\dots,\theta^\kappa)$ and r to $\mathcal{A}_1$ and $\mathcal{A}_2$.
- $\mathcal{A}_1$ returns $\{m_1^j\}_{j\in[\![1,\kappa]\!]}$ and $\mathcal{A}_2$ returns $\{m_2^j\}_{j\in[\![1,\kappa]\!]}$.

Let $m^j_{T_{r_j}} := \{m_i^j \mid \theta_i^j = r_j\}$. We say that $(\mathcal{A}_1,\mathcal{A}_2)$ win the game if $m_1^j = m_2^j = m_{T_{r_j}}$ for all $j\in[\![1,\kappa]\!]$.

Step 2: first upper-bound. Let $\theta = (\theta^1,...,\theta^\kappa)$, we define $\Pi_{\theta,r} := \bigotimes_{j=1}^{\kappa} \sum_{x\in\{0,1\}^n} |x\rangle\langle x|^{\theta^j} \otimes B^{\theta,r}_{x_{T_r}} \otimes C^{\theta,r}_{x_{T_r}}$. We then prove in the same way as in Theorem 2 that

$$p_{win} \leq \frac{1}{(2N)^\kappa} \sum_{\substack{k=k_1\|\dots\|k_\kappa \\ 1\leq k_j\leq N \ \forall j \\ \alpha\in\{0,1\}^\kappa}} \max_{\theta,r} \left\|\Pi_{\theta,r}\Pi_{\pi_{k,\alpha}(\theta,r)}\right\|$$

where $\{\pi_{k,\alpha}\}$ is a family of mutually orthogonal permutations indexed by $k = k_1\|\dots\|k_\kappa$ - where each $k_j\in[\![1,N]\!]$ - and $r\in\{0,1\}^\kappa$.

Step 3: upper-bound of $\|\Pi_{\theta,r}\Pi_{\theta',\bar{r}}\|$. Let $\theta = (\theta^1, ..., \theta^\kappa)$ and $\theta' = (\theta'^1, ..., \theta'^\kappa)$ where each θ^j and θ'^j belongs to Θ_n. Let $r \in \{0,1\}^\kappa$. For every $j \in [\![1,\kappa]\!]$, note $R^j := \{i \in [\![1,N]\!] \ : \ \theta_i^j \neq \theta_i'^j\}$, $T^j := \{i \in [\![1,N]\!] \ : \ \theta_i^j = r_j\}$, $T'^j := \{i \in [\![1,N]\!] \ : \ \theta_i'^j = 1 - r_j\}$ and $S^j := \{i \in R \ : \ \theta_i^j = r_j \text{ and } \theta_i'^j = 1 - r_j\}$. We define $\bar{\mathrm{P}}$ and $\bar{\mathrm{Q}}$ as follows:

$$\bar{\mathrm{P}} = \sum_{\substack{j\in[\![1,\kappa]\!] \\ x_{T^j}\in\{0,1\}^{T^j}}} \bigotimes_{j=1}^{\kappa} \mathsf{H}^{r_j}|x_{S^j}\rangle\langle x_{S^j}|\mathsf{H}^{r_j} \otimes \mathbb{I}_{\bar{S}^j} \otimes B_{x_T}^{\theta,r} \otimes \mathbb{I}_C$$

$$\bar{\mathrm{Q}} = \sum_{\substack{j\in[\![1,\kappa]\!] \\ x_{T'^j}\in\{0,1\}^{T'^j}}} \bigotimes_{j=1}^{\kappa} \mathsf{H}^{1-r_j}|x_{S^j}\rangle\langle x_{S^j}|\mathsf{H}^{1-r_j} \otimes \mathbb{I}_{\bar{S}^j} \otimes \mathbb{I}_B \otimes C_{x_{T'}}^{\theta',1-\bar{r}}$$

where $T := T^1\| \ldots \|T^\kappa$, $|x_{S^j}\rangle\langle x_{S^j}|$ denotes the subsystem of $|x_{T^j}\rangle\langle x_{T^j}|$ whose indices belong to S^j, and $\mathbb{I}_{\bar{S}^j}$ denotes the rest of the system.

Following the same reasoning as in Theorem 2 (step 3), it comes

$$\|\Pi_{\theta,r}\Pi_{\theta',\bar{r}}\| \leq 2^{-\frac{\sum_j |R^j|}{4}}$$

Step 4: finding the permutation family. Let $\{\pi^\star_{k,\alpha}\}_{k\in[\![1,N]\!],\alpha\in\{0,1\}}$ denotes the permutation family defined in step 4 of Theorem 2. We define the permutation family $\{\pi_{k,\beta}\}$ - indexed by $k = k_1\| \ldots \|k_\kappa$ where each $k_j \in [\![1,N]\!]$ and $\beta \in \{0,1\}^\kappa$ - as $\pi_{k,r}(\theta_1\| \ldots \|\theta_\kappa, r) = \pi^\star_{k_1,\beta_1}(\theta_1, r_1)\| \ldots \|\pi^\star_{k_\kappa,\beta_\kappa}(\theta_\kappa, r_\kappa)$. It is easy to see that this family is orthogonal and has the same required properties as in the single instance proof, that is that for every $j \in [\![1,\kappa]\!]$ and $i \in [\![1, n/2]\!]$, there are exactly $\binom{n/2}{i}^2$ permutations $\pi_{k,0}$ such that the number of positions at which θ^j and θ'^j are both 1 is $n/2 - i$ (i.e. $|R^j| = 2^i$). Using this set of permutations we have:

$$\begin{aligned}
p_{win} &\leq \frac{1}{(2N)^\kappa} \sum_{\substack{k=k_1\|\ldots\|k_\kappa \\ \beta\in\{0,1\}^\kappa}} \max_{\substack{\theta=\theta_1\|\ldots\|\theta_\kappa \\ r\in\{0,1\}^\kappa}} \|\Pi_{\theta,r}\Pi_{\theta',r'}\| \\
&= \frac{1}{(2N)^\kappa} \sum_{w=0}^{\kappa} \sum_{\substack{k=k_1\|\ldots\|k_\kappa \\ \beta\in\{0,1\}^\kappa,|\beta|=w}} \max_{\substack{\theta=\theta_1\|\ldots\|\theta_\kappa \\ r\in\{0,1\}^\kappa}} \|\Pi_{\theta,r}\Pi_{\theta',r'}\| \\
&\leq \frac{1}{(2N)^\kappa} \sum_{w=0}^{\kappa} \binom{\kappa}{w} \left(\sum_{\ell=0}^{n/2} \binom{n/2}{\ell}^2 2^{-\ell/2} \right)^w \\
&= \frac{1}{(2N)^\kappa} \left(1 + \sum_{\ell=0}^{n/2} \binom{n/2}{\ell}^2 2^{-\ell/2} \right)^\kappa \\
&\leq \frac{1}{(2N)^\kappa} \left(1 + \binom{n/2}{n/4}^2 \sum_{\ell=0}^{n/2} 2^{-\ell/2} \right)^\kappa \\
&= \frac{1}{(2N)^\kappa} \left(1 + \binom{n/2}{n/4}^2 \frac{1 - 2^{-n/4-1/2}}{1 - 2^{-1/2}} \right)^\kappa
\end{aligned}$$

Where in the first equality, we split the sum over the possible weights of β; the first inequality comes from Corollary 1; we obtain the second equality by applying the binomial theorem; the second inequality comes from $\binom{n}{k} \leq \binom{n}{n/2}$ for all n, k; and the last inequality comes from the fact that the sum is the sum of a geometric series.

Using both Stirling approximation and asymptotic development of logarithm, we get that the logarithm of this last inequality decreases linearly in k, meaning that the upper bound is negligible in n which concludes the proof.

4 Single-Decryptor and Copy-Protection of Pseudorandom Functions

In this section, we recall the notions of single-decryptor and copy-protection of pseudorandom functions. These primitives are used later to prove the security of our constructions of copy-protection of point functions and unclonable encryption. In [CLLZ21], the authors give a definition of anti-piracy security and provide a secure construction for these two primitives. We give two variants of anti-piracy security of single-decryptor and of anti-piracy security of copy-protection of pseudorandom functions and show that their constructions are secure with respect to these two variants.

4.1 Definition of a Single-Decryptor

Definition 13 (Single-decryptor encryption scheme). *A single-decryptor encryption scheme is a tuple of algorithms* $\langle \mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ *with the following properties:*

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$. *On input a security parameter* λ*, the classical setup algorithm* Setup *outputs a classical secret key* sk *and a public key* pk*.*
- $(\rho_{\mathsf{sk}}) \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$. *On input a classical secret key* sk*, the quantum key generation algorithm* QKeyGen *outputs a quantum secret key* ρ_{sk}*.*
- $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$. *On input a public key* pk *and a message* m *in the message space* $\mathcal{M}$*, the classical randomized encryption algorithm* Enc *outputs a classical ciphertext* c*. We sometimes write* $\mathsf{Enc}(\mathsf{pk}, m; r)$ *to precise that we use the the random bitstring* r *as the randomness in the algorithm.*
- $m/\bot \leftarrow \mathsf{Dec}(\rho_{\mathsf{sk}}, c)$. *On input a quantum secret key* ρ_{sk}*, a classical ciphertext* y*, the quantum decryption algorithm* Dec *outputs a classical message* m *or a decryption failure symbol* $\bot$*.*

Correctness. We say that a single-decryptor scheme $\langle \mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ has correctness if there exists a negligible function $\mathsf{negl}(\cdot)$, such that for all $\lambda \in \mathbb{N}$, for all $m \in \mathcal{M}$, the following holds:

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

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 [Wil11].

Anti-piracy security. We now define indistinguishable anti-piracy security of a single-decryptor scheme. This security notion is define with respect to two distributions used in the piracy game that we defined below: $\mathcal{D}_B$ that yields two bits and decides which message will be encrypted for each test, and $\mathcal{D}_R$ that yields two strings to be used as the randomness for the encryption in each test. Note that the original anti-piracy security proposed in [CLLZ21] is simply our notion with respect to $\mathcal{D}_B$ and $\mathcal{D}_R$ taken as uniform distributions. In contrast, in order to prove the security of our unclonable encryption and copy-protection schemes, we need to consider the security when $\mathcal{D}_R$ yields a pair of identical random string (r, r) (sampled uniformly at random) and $\mathcal{D}_B$ either is the uniform distribution and hence yields pairs of the form (b_1, b_2) or is the "identical distribution" and yields pairs of the form (b, b).

Definition 14 (Piracy game for single-decryptor). *We define below a* piracy game for single-decryptor*, parameterized by a single-decryptor scheme* $\mathcal{E} = \langle \mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$*, a security parameter* λ *and a pair of distributions* $(\mathcal{D}_B, \mathcal{D}_R)$*. This game is between a challenger and an adversary represented by a QPT algorithm* $\mathcal{A}$*.*

- ***Setup phase:***
  - *The challenger samples* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
  - *The challenger samples* $\rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$.
  - *The challenger sends* $(\mathsf{pk}, \rho_{\mathsf{sk}})$ *to* $\mathcal{A}$.
- ***Splitting phase:***
  - $\mathcal{A}$ *outputs a bipartite quantum state* σ_{12}*, two quantum circuits* U_1 *and* U_2*, and two pairs of messages* (m_0^1, m_1^1) *and* (m_0^2, m_1^2).
- ***Challenge phase:*** *The challenger samples* $(b_1, b_2) \leftarrow \mathcal{D}_B$*,* $(r_1, r_2) \leftarrow \mathcal{D}_R$ *and for* $i \in \{1, 2\}$*, the challenger performs the following test on* (σ_i, U_i)*:*
  - *compute* $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_{b_i}; r_i)$*;*
  - *run the circuit* U_i *on* $\sigma_i \otimes |c\rangle\langle c|$ *and checks whether the outcome is* b*. If so, we say that* (σ_i, U_i) *passes the test, otherwise that it does not.*

  $\mathcal{A}$ *wins the game if both* (σ_1, U_1) *and* (σ_2, U_2) *pass the test.*

We denote the random variable that indicates whether an adversary $\mathcal{A}$ *wins the game or not as* $\mathsf{SD} - \mathsf{AP}^{\mathcal{E}}_{(\mathcal{D}_B, \mathcal{D}_R)}(1^\lambda, \mathcal{A})$.

Remark 2. [CLLZ21] shows that we can assume that the tests of the challenge phase can be performed in a projective way - that is the test is actually a projective measurement. For sake of readability, we do not define formally the notions of *projective/threshold implementation* and their approximated versions and refer the interested reader to [CLLZ21] for a more in-depth discussion about them. Thus, in all the following proofs, we consider that all the tests for the different piracy games are performed in a projective way (Definitions 14, 17 and 20).

Definition 15 (Indistinguishable anti-piracy security). *A single-decryptor scheme* $\mathcal{E}$ *has* indistinguishable anti-piracy security *with respect to distributions* $(\mathcal{D}_B, \mathcal{D}_R)$ *if no* QPT *adversary can win the piracy game above with probability significantly greater than* $1/2$*. More precisely, for any* QPT *adversary* $\mathcal{A}$*:*

$$\Pr\Big[\mathsf{SD} - \mathsf{AP}^{\mathcal{E}}_{(\mathcal{D}_B, \mathcal{D}_R)}(1^\lambda, \mathcal{A}) = 1\Big] \leq 1/2 + \mathsf{negl}(\lambda).$$

Whenever, a single-decryptor is secure with respect to the uniform distribution $\mathcal{D}_B$ *and the "identical distribution"* $\mathcal{D}_R$ *- that yields uniformly random pairs of the form* (r, r) *- we say that the single-decryptor has indistinguishable anti-piracy security in the* identical randomness *style.*

Furthermore, whenever a single-decryptor is secure with respect to the identical distribution $\mathcal{D}_B$ *and the identical distribution* $\mathcal{D}_R$ *- that yield uniformly random pairs of the form* (b, b) *and* (r, r) *respectively - we say that the single-decryptor has indistinguishable anti-piracy security in the* identical randomness and challenge bits *style.*

4.2 Construction of Single-Decryptor

In this section, we present the single-decryptor construction of [CLLZ21].

Construction 1: [CLLZ21] Single-Decryptor Scheme

Given a security parameter λ, let $n := \lambda$ and κ be polynomial in λ.

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$:
  - Sample coset spaces $\{A_i, s_i, s'_i\}_{i \in [\![1,\kappa]\!]}$ where each A_i is of dimension $n/2$;
  - Construct the membership programs for each coset $\{\widehat{\mathsf{P}}_{A_i + si}, \widehat{\mathsf{P}}_{A_i^\perp + s'i}\}_{i \in [\![1,\kappa]\!]}$;
  - Return $\Big(\mathsf{sk} := \{A_i, s_i, s'_i\}_{i \in [\![1,\kappa]\!]}, \mathsf{pk} := \{\widehat{\mathsf{P}}_{A_i + si}, \widehat{\mathsf{P}}_{A_i^\perp + s'i}\}_{i \in [\![1,\kappa]\!]}\Big)$.

- $\rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$:
  - Parse sk as $\{A_i, s_i, s'_i\}_{i\in[\![1,\kappa]\!]}$;
  - Return $\bigotimes_{i=1}^{\kappa} |A_{i,s_i,s'_i}\rangle$.
- $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$:
  - Parse pk as $\{\widehat{\mathsf{P}}_{A_i+si}, \widehat{\mathsf{P}}_{A_i^{\perp}+s'i}\}_{i\in[\![1,\kappa]\!]}$;
  - Sample $r \leftarrow\$ \{0,1\}^{\kappa}$;
  - Generate an obfuscated program $\mathsf{iO}(\mathsf{Q}_{\mathsf{m},\mathsf{r}})$ of program $\mathsf{Q}_{\mathsf{m},\mathsf{r}}$ described in Section 4.2.
  - Return $c \coloneqq (r, \mathsf{iO}(\mathsf{Q}_{\mathsf{m},\mathsf{r}}))$.
- $m/\perp \leftarrow \mathsf{Dec}(\rho_{\mathsf{sk}}, c)$:
  - Parse ρ_{sk} as $\bigotimes_{i=1}^{\kappa} |A_{i,s_i,s'_i}\rangle$ and $c \leftarrow (r, \mathsf{iO}(\mathsf{Q}_{\mathsf{m},\mathsf{r}}))$;
  - For all $i \in [\![1,\kappa]\!]$: if $r_i = 1$, apply $\mathsf{H}^{\otimes n}$ to $|A_{i,s_i,s'_i}\rangle$;
  - Let ρ' be the resulting state, run $\mathsf{iO}(\mathsf{Q}_{\mathsf{m},\mathsf{r}})$ coherently on ρ' and measure the final register to get m;
  - Return m.

Hardcoded: Programs $\{\mathsf{P}_i\}_{i\in[\![1,\kappa]\!]}$ such that for all $i \in [\![1,\kappa]\!]$: $\mathsf{P}_i := \begin{cases} \widehat{\mathsf{P}}_{A_i+si} & \text{if } r_i = 0 \\ \widehat{\mathsf{P}}_{A_i^{\perp}+s'i} & \text{if } r_i = 1 \end{cases}$.

On input vectors $u_1, u_2, \ldots, u_\kappa$, do the following:

1. If for all $i \in [\![1,\kappa]\!]$: $\mathsf{P}_i(u_i) = 1$, then output m.
2. Otherwise: output $\perp$.

Fig. 2. Program $\mathsf{Q}_{\mathsf{m},\mathsf{r}}$.

Remark 3. Note that the underlying iO algorithm used in the encryption algorithm of Construction 1 might use randomness. In the following, we denote by $\mathsf{Enc}(\mathsf{pk}, m; (r_{\mathsf{iO}}, r))$ the encryption of a message m with the key pk and with random bitstrings r_{iO} and r respectively used for the iO algorithm and for the program $\mathsf{Q}_{\mathsf{m},\mathsf{r}}$.

Theorem 5. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, Construction 1 has* indistinguishable anti-piracy security in the identical randomness style*.*

Theorem 6. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, Construction 1 has* indistinguishable anti-piracy security in the identical randomness and challenge bits style*.*

4.3 Proof of Theorem 5

In this section, we prove Theorem 5. Our proof follows the structure of [CLLZ21]. We proceed in the proof through a sequence of hybrids. For any pair of hybrids (G_i, G_j), we say that G_i is *negligibly close to* G_j if for every QPT adversary $\mathcal{A} \coloneqq (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, the probability that $\mathcal{A}$ wins G_i is negligibly close to the probability that they win G_j.

Game G_0: This is the piracy game for the single-decryptor of Construction 1, in the identical randomness style.

- **Setup phase:**
  - The challenger samples coset spaces $\{A_i, s_i, s'_i\}_{i\in[\![1,\kappa]\!]}$ where each A_i is of dimension $n/2$;
  - Then the challenger constructs the membership programs for each coset $\{\widehat{\mathsf{P}}_{A_i+si}, \widehat{\mathsf{P}}_{A_i^\perp+s'i}\}_{i\in[\![1,\kappa]\!]}$;
  - Finally the challenger sends $\rho_{\mathsf{sk}} := \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\kappa]\!]}$ and $\mathsf{pk} := \{\widehat{\mathsf{P}}_{A_i+si}, \widehat{\mathsf{P}}_{A_i^\perp+s'i}\}_{i\in[\![1,\kappa]\!]}$ to $\mathcal{A}$.
- **Splitting phase:**
  - $\mathcal{A}$ outputs a bipartite quantum state σ_{12}, two quantum circuits U_1 and U_2, and two pairs of messages (m_0^1, m_1^1) and (m_0^2, m_1^2).
- **Challenge phase:** The challenger samples random bitstrings r_{iO} and r, to be used in the encryption algorithm, and performs the following test on each (σ_i, U_i) for $i \in \{1, 2\}$:
  - sample $b \in \{0, 1\}$;
  - compute $c := (r, \mathsf{Q}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b^i; (r, r_{\mathsf{iO}}))$ - note that the program Q has been obfuscated using r_{iO} as the randomness;
  - run the circuit U_i on $\sigma_i \otimes |c\rangle\langle c|$ and checks whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

$\mathcal{A}$ wins the game if both (σ_1, U_1) and (σ_2, U_2) pass the test.

Game G_1: In this second hybrid, we replace the obfuscated programs Q by obfuscated compute-and-compare programs. More formally, for $i \in [\![1, \kappa]\!]$, we note[2]

$$\mathsf{Can}_{i,b}(\cdot) := \begin{cases} \mathsf{Can}_{A_i}(\cdot) & \text{if } b = 0 \\ \mathsf{Can}_{A_i^\perp}(\cdot) & \text{if } b = 1 \end{cases} \text{ and } c_{i,b} := \begin{cases} \mathsf{Can}_{A_i}(s_i) & \text{if } b = 0 \\ \mathsf{Can}_{A_i^\perp}(s'_i) & \text{if } b = 1 \end{cases}$$

We similarly define $\mathsf{Can}_r(u_1, ..., u_\kappa) = (\mathsf{Can}_{1,r_1}(u_1), \ldots, \mathsf{Can}_{\kappa,r_\kappa}(u_\kappa))$ and $c_r = (c_{1,r_1}, \ldots, c_{\kappa,r_\kappa})$ for any $r \in \{0,1\}^\kappa$.

Then, we replace Q by $\mathsf{iO}(\mathsf{CC}[\mathsf{Can}_r, c_r, m_b^1])$ in the first test and by $\mathsf{iO}(\mathsf{CC}[\mathsf{Can}_r, c_r, m_b^2])$ in the second test. Because the programs Q in each test $i \in \{1, 2\}$ and $\mathsf{CC}[\mathsf{Can}_r, c_r, m_b^i]$ are functionally equivalent, G_0 and G_1 are negligibly close from iO security.

Game G_2: In this last hybrid, we replace $\mathsf{iO}(\mathsf{CC}[\mathsf{Can}_r, c_r, m_b^i])$ by $\mathsf{iO}(\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}[\mathsf{Can}_r, c_r, m_{b_i}^i]))$ in each test $i \in \{1, 2\}$. Because the programs $\mathsf{CC}[\mathsf{Can}_r, c_r, m_b^i]$ and $\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}[\mathsf{Can}_r, c_r, m_{b_i}^i])$ are functionally equivalent, G_1 and G_2 are negligibly close from iO security.

Leveraging compute-and-compare obfuscation. Before proceeding to the reduction, we introduce the two following lemmas.

Lemma 5. *There exists an efficient algorithm* Ext *such that the following holds. Sample uniformly at random* κ *cosets spaces* $\{A_i, s_i, s'_i\}_{i\in[\![1,\kappa]\!]}$ *and a* κ*-long bitstring* r*. Let* (σ, U) *be a pair of quantum state and quantum circuits that passes the test of* G_2 *with respect to a pair of messages* (m_0, m_1) *with non-negligible advantage over* $1/2$*.*

Then Ext*, on input the description of the subspaces* $(A_i)_{i\in[\![1,\kappa]\!]}$*, the bitstring* r*, and the pair* (σ, U)*, outputs the canonical vectors* $(u_1, \ldots, u_\kappa)$ *- where* u_i *is the canonical vector for the coset space* $A_i + s_i$ *if* $r_i = 0$ *or* $A_i^\perp + s'_i$ *if* $r_i = 1$ *- with non-negligible probability.*

Proof. We define two distributions $\mathcal{D}$ and $\mathcal{D}_{\mathsf{Sim}}$, both parametrized by a pair of messages m_0, m_1. Let $\mathcal{D}$ be the following distribution:

- sample $\mathsf{sk} := \{A_i, s_i, s'_i\}_{i\in[\![1,\kappa]\!]}$ as in the setup phase of G_2;
- sample r as in the challenge phase of G_2;

[2] Recall that for a subspace A and a vector u, $\mathsf{Can}_A(u)$ - defined in Section 2.2 - is the coset representative of $A + u$. Recall also that Can_A can be efficiently implemented given a description of A.

- compute $\mathsf{C} := \mathsf{iO}(\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}[\mathsf{Can}_r, c_r, m_b]))$ as in the challenge phase of G_2;
- let (σ, U) be a pair (quantum state, quantum circuit) that passes the test of G_2 with respect to (m_0, m_1) with non-negligible advantage over $1/2$;
- output $(\mathsf{C}, \mathsf{aux})$ where $\mathsf{aux} := (\sigma, \mathsf{U})$.

Let $\mathcal{D}_{\mathsf{Sim}}$ be the same distribution, except that C is replaced by the output of an efficient simulator Sim taking as input $(1^\lambda, \mathsf{C.params})$, where $\mathsf{C.params}$ denotes the parameters of C (input and output size, circuit size, etc.). From compute-and-compare obfuscation (Definition 4), we know that if, for any simulator Sim, there exists an efficient procedure to distinguish these two distributions $\mathcal{D}$ and $\mathcal{D}_{\mathsf{Sim}}$, then there exists an efficient algorithm Ext, that takes as input aux and the description of the function in the compute-and-compare program (here Can_r), and extracts the lock-value of the program C, namely the canonical vectors $(u_i)_{i\in[\![1,\kappa]\!]}$ respectively for the coset spaces $A_i + s_i$ if $r_i = 0$ or $A_i^\perp + s'_i$ if $r_i = 1$. We describe below such an efficient procedure to distinguish the two distributions. On input $(\mathsf{C}^*, \mathsf{aux} = (\sigma, \mathsf{U}))$, sampled from $\mathcal{D}$ or $\mathcal{D}_{\mathsf{Sim}}$, the procedure behaves in the following way:

- sample $b \leftarrow\!\!\$\ \{0,1\}$;
- run U on $\sigma \otimes |\mathsf{C}^*\rangle\langle\mathsf{C}^*|$; let b^* denotes the outcome;
- if $b^* = b$: return 0, meaning that C^* comes from $\mathcal{D}$;
- otherwise: return 1, meaning that C^* comes from $\mathcal{D}_{\mathsf{Sim}}$.

Because, (σ, U) passes the challenge test of G_2 with non-negligible advantage over $1/2$, the procedure succeeds with the same advantage conditioned on C^* coming from $\mathcal{D}$. On the other hand, when C^* comes from $\mathcal{D}_{\mathsf{Sim}}$, then C^* does not hold any information on b, hence $b^* = b$ only with probability $1/2$. Thus the procedure distinguishes the two distributions with non-negligible advantage over $1/2$. Because Can_r can be implemented given $(A_i)_{i\in[\![1,\kappa]\!]}$ and r, it concludes the proof. □

The following lemma is a claim from [CLLZ21] adapted to our settings. We refer the interested reader to [CLLZ21] for a proof of this lemma.

Lemma 6 (Claim 6.18 of [CLLZ21]). *Let $\sigma_{12}, \mathsf{U}_1, \mathsf{U}_2$ a bipartite quantum state and two quantum circuits such that both (σ_1, U_1) and (σ_2, U_2) pass the (projective) test in the challenge phase. Apply any POVM on σ_1; let σ'_{12} denotes the post-measurement state. Then $(\sigma'_2, \mathsf{U}_2)$ still passes the (projective) test of the challenge phase with non-negligible probability over $1/2$.*

Reduction to monogamy-of-entanglement. We are now ready to proceed to the reduction. Assume that there exists a QPT algorithm $\mathcal{A}$ that wins the last hybrid G_2 with non-negligible advantage over $1/2$. We construct a triple of QPT algorithms $(\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2)$ that win the κ-parallel computational version of monogamy-of-entanglement game (Definition 12) with with non-negligible advantage over $1/2$, which contradicts Theorem 4. $\mathcal{B}$ behaves in the following way; on input $|A_{i,s_i,s'_i}\rangle_{i\in[\![1,\kappa]\!]}$ and $\left(\widehat{\mathsf{P}}_{A_i+s_i}, \widehat{\mathsf{P}}_{A_i^\perp+s'_i}\right)_{i\in[\![1,\kappa]\!]}$:

- $\mathcal{B}_0$ sets $\rho_{\mathsf{sk}} := |A_{i,s_i,s'_i}\rangle_{i\in[\![1,\kappa]\!]}$ and $\mathsf{pk} := \left(\widehat{\mathsf{P}}_{A_i+s_i}, \widehat{\mathsf{P}}_{A_i^\perp+s'_i}\right)_{i\in[\![1,\kappa]\!]}$.
- $\mathcal{B}_0$ runs $\mathcal{A}$ on $(\rho_{\mathsf{sk}}, \mathsf{pk})$; let $\big(\sigma_{12}, (\mathsf{U}_1, \mathsf{U}_2), (m_0^1, m_1^1), (m_0^2, m_1^2)\big)$ denote the outcome.
- $\mathcal{B}_0$ samples r_{iO}: a uniformly random bitstring to be used in iO.
- For $i \in \{1,2\}$, let $\sigma'_i := \sigma_i \otimes |\mathsf{U}_i\rangle\langle\mathsf{U}_i|$; $\mathcal{B}_0$ sends σ'_1 to $\mathcal{B}_1$ and σ'_2 to $\mathcal{B}_2$.

For $i \in \{1,2\}$: on input $\sigma'_i := \sigma_i \otimes |\mathsf{U}_i\rangle\langle\mathsf{U}_i|$ as well as the challenge $\big((A_i)_{i\in[\![1,\kappa]\!]}, r\big)$: $\mathcal{B}_i$ runs the algorithm Ext from Lemma 5 on $\big((A_i)_{i\in[\![1,\kappa]\!]}, r\big)$ and (σ_i, U_i) to extract $(u_1, \ldots, u_\kappa)$. $\mathcal{B}_i$ then returns $(u_1, \ldots, u_\kappa)$. Note that both (σ_1, U_1) and (σ_2, U_2) pass the test of the challenge phase with non-negligible advantage over $1/2$, and that all the other elements are sampled as in Lemma 5, allowing us to use the extractor. Although, one might

fear the using the extractor on σ_1 would disturb σ_2, making this new $(\sigma_2', \mathsf{U}_2)$ unable to pass the test of the challenge phase (and hence we would not be able to perform the extraction procedure on σ_2'), we know from Lemma 6 that this does not happen and σ_2' still passes the test after the action of the extractor on σ_1. Thus, both $\mathcal{B}_1$ and $\mathcal{B}_2$ extract the expected $(u_1, \ldots, u_\kappa)$ with non-negligible probability, which concludes the proof.

4.4 Proof of Theorem 6

We now prove Theorem 6. As this proof is very similar to the one of Theorem 5, we only provide a sketch of it. We proceed with a sequence of hybrids. The first one G_0 is the piracy game in the identical randomness and challenge bits style. We recall that it is the same game as the first hybrid of the proof of Theorem 5, except that the tests in the challenge phase either both use the encryption of the first message of the corresponding pair, or both use the second message (depending on a random bit b). In the second and third hybrids G_1 and G_2, we do the same as in the proof above, namely we replace the obfuscated programs in the ciphertexts by their compute-and-compare obfuscated versions, with function Can_r and lock-value c_r. Finally, the reduction is exactly the same as above: we use Lemma 5 and Lemma 6 to argue that both $\mathcal{B}_1$ and $\mathcal{B}_2$ can use an extractor Ext on their respective shares of σ_{12} and obtain the expected vectors $(u_1, \ldots, u_\kappa)$, which concludes the proof.

4.5 Copy-Protection of Pseudorandom Functions

In this subsection, we formally define copy-protection of pseudorandom function and its correctness and anti-piracy notions.

Definition 16 (Pseudorandom function copy-protection scheme). *A pseudorandom function copy-protection scheme for the pseudorandom function* $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *(where* $\mathcal{Y} \subseteq \{0,1\}^m$*) associated with the key generation procedure* KeyGen *is a tuple of algorithms* $\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ *with the following properties:*

- $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$. *This is the key generation procedure of the underlying pseudorandom function: on input a security parameter, the* KeyGen *algorithm outputs a key* k.
- $\rho_\mathsf{k} \leftarrow \mathsf{Protect}(1^\lambda, \mathsf{k})$. *On input a pseudorandom function key* $\mathsf{k} \in \mathcal{K}$*, the quantum protection algorithm outputs a quantum state* ρ_k.
- $y \leftarrow \mathsf{Eval}(1^\lambda, \rho, x)$. *On input a quantum state* ρ *and an input* $x \in \mathcal{X}$*, the quantum evaluation algorithm outputs* $y \in \mathcal{Y}$.

Correctness. A pseudorandom function copy-protection scheme has *correctness* if the quantum protection of any key k computes $\mathsf{PRF}(\mathsf{k}, \cdot)$ on every x with overwhelming probability.

$$\forall \mathsf{k} \in \mathcal{K},\ \forall x \in \mathcal{X},\ \Pr\left[\mathsf{Eval}(1^\lambda, \rho_\mathsf{k}, x) = \mathsf{PRF}(\mathsf{k}, x)\ :\ \rho_\mathsf{k} \leftarrow \mathsf{Protect}(1^\lambda, \mathsf{k})\right] = 1 - \mathsf{negl}(\lambda)$$

Anti-piracy security. We now define anti-piracy security of a pseudorandom function copy-protection scheme in a similar way as the anti-piracy of single-decryptor. Anti-piracy security is defined through the following *piracy game*, in which the adversary is provided a quantum key and a pseudorandom function image, and must "split" the quantum key such that both shares can be used to distinguish between the input of this image or another "fake" input sampled uniformly at random. We define two different "styles" for this security: the *product challenge bits style* and the *identical distribution style*. In the *product challenge bits style*, the challenge in the test of each "share" is either the input of the image or a freshly sampled fake input (both with probability 1/2). In the *identical distribution style* on the other hand, a fake input is sampled before the tests and either both shares are tested with the input of the image, or with this fake input.

Definition 17 (Piracy game for pseudorandom function copy-protection). *We define below a* piracy game for pseudorandom function copy-protection*, parameterized by a pseudorandom function copy-protection scheme* $\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ *and a security parameter* λ*. As the game in the product challenge bits style and the game in the identical distribution style differ only in the challenge phase, we define a different challenge phase for each style.*

- ***Setup phase:***
  - *The challenger samples* $\mathsf{k} \in \mathsf{KeyGen}(1^\lambda)$ *and computes* $\rho_\mathsf{k} \leftarrow \mathsf{Protect}(1^\lambda, \mathsf{k})$.
  - *The challenger samples* $x \leftarrow\!\!\$\ \mathcal{X}$ *and computes* $y := \mathsf{PRF}(\mathsf{k}, x)$.
  - *The challenger sends* ρ_k *and* y *to* $\mathcal{A}$.
- ***Splitting phase:*** $\mathcal{A}$ *outputs a bipartite quantum state* σ_{12}*, as well as two quantum circuits* U_1 *and* U_2.
- ***Challenge phase (product challenge bits style):*** *The challenger performs the following test on* (σ_i, U_i) *for* $i \in \{1, 2\}$*:*
  - *sample* $b \leftarrow\!\!\$\ \{0, 1\}$*;*
  - *set* $x_0 := x$ *and sample* $x_1 \leftarrow\!\!\$\ \mathcal{X}$.
  - *run the circuit* U_i *on* $\sigma_i \otimes |x_b\rangle\langle x_b|$ *and check whether the outcome is* b*. If so, we say that* (σ_i, U_i) *passes the test, otherwise that it does not.*
- ***Challenge phase (identical distribution style):*** *The challenger samples a bit* $b \leftarrow\!\!\$\ \{0, 1\}$ *and a fake input* $x_1 \leftarrow\!\!\$\ \mathcal{X}$*, then performs the following test on* (σ_i, U_i) *for* $i \in \{1, 2\}$*:*
  - *set* $x_0 := x$*;*
  - *run the circuit* U_i *on* $\sigma_i \otimes |x_b\rangle\langle x_b|$ *and check whether the outcome is* b*. If so, we say that* (σ_i, U_i) *passes the test, otherwise that it does not.*

$\mathcal{A}$ *wins the game if both* (σ_1, U_1) *and* (σ_2, U_2) *pass the test.*

We denote the random variable that indicates whether an adversary $\mathcal{A}$ *wins the game or not as* $\mathsf{CP}-\mathsf{PRF}-\mathsf{AP}_{PCB}^{\langle \mathsf{KeyGen},\mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A})$ *or* $\mathsf{CP}-\mathsf{PRF}-\mathsf{AP}_{ID}^{\langle \mathsf{KeyGen},\mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A})$ *depending if we use the product challenge bits or the identical distribution style.*

Remark 4. As said in Remark 2, we assume that the tests of each challenge phase are done in a projective way.

Definition 18 (Indistinguishable anti-piracy security). *A pseudorandom function copy-protection scheme* $\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval}\rangle$ *has* indistinguishable anti-piracy security *in the product challenge bits style if no* QPT *adversary can win the piracy game above in the product challenge bits style with probability significantly greater than* $1/2$*. More precisely, for any* QPT *adversary* $\mathcal{A}$*:*

$$\Pr\Big[\mathsf{CP}-\mathsf{PRF}-\mathsf{AP}_{PCB}^{\langle \mathsf{KeyGen},\mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A}) = 1\Big] \leq 1/2 + \mathsf{negl}(\lambda).$$

Furthermore, we say that such a scheme has indistinguishable anti-piracy security *in the identical distribution style if no* QPT *adversary can win the piracy game above in the identical distribution style with probability significantly greater than* $1/2$*. More precisely, for any* QPT *adversary* $\mathcal{A}$*:*

$$\Pr\Big[\mathsf{CP}-\mathsf{PRF}-\mathsf{AP}_{ID}^{\langle \mathsf{KeyGen},\mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A}) = 1\Big] \leq 1/2 + \mathsf{negl}(\lambda).$$

Theorem 7. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, there exists a pseudorandom function copy-protection scheme with indistinguishable anti-piracy security both in the product challenge bits style and in the identical distribution style.*

We present the construction that achieves this security in the two styles and the corresponding proof in Appendix A.

5 Copy-Protection of Point Functions in the Plain Model

In this section, we present the definition of copy-protection of point functions. Then we present a construction of this primitive from [CHV23]. This construction was proven secure for a non-colliding anti-piracy game's challenge distribution. We prove that the same construction is actually secure for the product challenge distribution as well as the identical challenge distribution.[3]Through all this section, λ denotes a security parameter and $n = \mathsf{poly}(\lambda)$.

5.1 Definitions

We consider copy-protection of point functions for a family of point functions $\{\mathsf{PF}_y\}_{y\in\{0,1\}^n}$. We denote PF_y the point function with point y, i.e. the function such that

$$\mathsf{PF}_y(x) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

Definition 19 (Point functions copy-protection scheme). *A copy-protection scheme of a family of point functions $\{\mathsf{PF}\}_{y\in\{0,1\}^n}$ is a tuple of algorithms $\langle\mathsf{Protect}, \mathsf{Eval}\rangle$ with the following properties:*

- $\rho_y \leftarrow \mathsf{Protect}(1^\lambda, y)$*. On input a point $y \in \{0,1\}^n$, the quantum protection algorithm outputs a quantum state ρ_y.*
- $b \leftarrow \mathsf{Eval}(1^\lambda, \rho, x)$*. On input a quantum state ρ and an input $x \in \{0,1\}^n$, the quantum evaluation algorithm outputs a bit $b \in \{0,1\}$.*

Correctness. A point functions copy-protection scheme has *correctness* if the quantum protection of any point function PF_y computes PF_y on every x with overwhelming probability.

$$\forall y \in \{0,1\}^n,\ \forall x \in \{0,1\}^n,\ \Pr\left[\mathsf{Eval}(1^\lambda, \rho_y, x) = \mathsf{PF}_y(x)\ :\ \rho_y \leftarrow \mathsf{Protect}(1^\lambda, y)\right] = 1 - \mathsf{negl}(\lambda)$$

Anti-piracy security. We now define anti-piracy security of a point functions copy-protection scheme. This notion is defined through a piracy game, in which the adversary is given a quantum copy-protection of a point function PRF_y and must split it such that both shares can be used to evaluate the function correctly. We consider two variants of this security notion, namely *anti-piracy security with respect to the product distribution* and *anti-piracy security with respect to the identical distribution.* In the first variant, the challenge in the test of each share is either the point y or another freshly sampled random point; in the second variant, a random point x is sampled before the tests, and either both shares are tested with y or both are tested with x.

Definition 20 (Piracy game for copy-protection of point functions). *We define below a* piracy game for copy-protection of point functions*, parameterized by a copy-protection scheme $\mathsf{CP} = \langle\mathsf{Protect}, \mathsf{Eval}\rangle$ and a security parameter λ. This game is between a challenger and an adversary $\mathcal{A}$. As the two variants of the game differ only in the challenge phase, we describe below a different challenge phase for each variant.*

- ***Setup phase:***
  - *The challenger samples $y \in \{0,1\}^n$ and computes $\rho_y \leftarrow \mathsf{Protect}(1^\lambda, y)$.*
  - *The challenger then sends ρ_y to $\mathcal{A}$.*
- ***Splitting phase:*** *$\mathcal{A}$ outputs a bipartite quantum state σ_{12} as well as two quantum circuits U_1 and U_2.*
- ***Challenge phase (product distribution):*** *For $i \in \{1,2\}$, the challenger performs the following test on (σ_i, U_i):*
  - *sample $b \leftarrow\!\!\$\ \{0,1\}$;*

[3] We actually present a more general version of the construction of [CHV23].

- *set $x_0 := y$ and sample $x_1 \leftarrow\!\!\$\ \{0,1\}^n$;*
- *run U_i on $\sigma_i \otimes |x_b\rangle\langle x_b|$ and check whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.*

- ***Challenge phase (identical distribution):*** *The challenger samples a bit $b \leftarrow\!\!\$\ \{0,1\}$ and an input $x_1 \leftarrow\!\!\$\ \{0,1\}^n$ and performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:*
  - *set $x_0 := y$;*
  - *run U_i on $\sigma_i \otimes |x_b\rangle\langle x_b|$ and check whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.*

$\mathcal{A}$ wins the game if both (σ_1, U_1) and (σ_2, U_2) pass the test. We denote the random variable that indicates whether an adversary $\mathcal{A}$ wins the game or not as $\mathsf{CP} - \mathsf{AP}_{PD}^{\langle \mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A})$ or as $\mathsf{CP} - \mathsf{AP}_{ID}^{\langle \mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A})$ depending on which variant is considered in the game.

Remark 5. As said in Remark 2, we assume that the tests of the challenge phase are done in a projective way.

Definition 21 (Anti-piracy security). *A point functions copy-protection scheme $\langle \mathsf{Protect}, \mathsf{Eval}\rangle$ has* anti-piracy security *with respect to the product distribution if no QPT adversary can win the piracy game above with probability significantly greater than $1/2$. More precisely, for any QPT adversary $\mathcal{A}$:*

$$\Pr\Big[\mathsf{CP} - \mathsf{AP}_{PD}^{\langle \mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A}) = 1\Big] \leq 1/2 + \mathsf{negl}(\lambda).$$

Furthermore, we say that a point functions copy-protection scheme $\langle \mathsf{Protect}, \mathsf{Eval}\rangle$ has anti-piracy security *with respect to the identical distribution if no QPT adversary can win the piracy game above with probability significantly greater than $1/2$. More precisely, for any QPT adversary $\mathcal{A}$:*

$$\Pr\Big[\mathsf{CP} - \mathsf{AP}_{ID}^{\langle \mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A}) = 1\Big] \leq 1/2 + \mathsf{negl}(\lambda).$$

5.2 Construction

In this subsection, we present a construction for copy-protection of point functions. This construction uses a pseudorandom functions copy-protection scheme $\mathsf{PRF}.\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval}\rangle$.

Construction 2: Copy-Protection of Point Functions

- $\mathsf{Protect}(1^\lambda, y)$:
  - Sample $\mathsf{k} \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$.
  - Compute $\rho_\mathsf{k} \leftarrow \mathsf{PRF.Protect}(\mathsf{k})$.
  - Compute $z := \mathsf{PRF}(\mathsf{k}, y)$.
  - Return (ρ_k, z).
- $\mathsf{Eval}(1^\lambda, (\rho, z), x)$:
  - Compute $z' \leftarrow \mathsf{PRF.Eval}(\rho, x)$.
  - If $z' = z$: return 1.
  - Otherwise: return 0.

Theorem 8. *Assuming the underlying pseudorandom functions copy-protection scheme has anti-piracy security in the product challenge bits style, Construction 2 has correctness and anti-piracy security with respect to the product distribution.*

Theorem 9. *Assuming the underlying pseudorandom functions copy-protection scheme has anti-piracy security in the identical distribution style, Construction 2 has correctness and anti-piracy security with respect to the identical distribution.*

Proof of Theorems 8 and 9. (Correctness) for any y, running the evaluation algorithm on the point y yields 1 with probability close to 1 from the correctness of the underlying copy-protection of pseudorandom functions. Running the evaluation algorithm on a point $x \neq y$ yields 1 only if $\mathsf{PRF.Eval}(\rho_{\mathsf{k}}, x) = \mathsf{PRF}(\mathsf{k}, y)$, which happens with negligible probability over k from the security of the underlying pseudorandom function.

(Anti-piracy security) the anti-piracy security with respect to the product distribution (resp. identical distribution) comes directly from the anti-piracy security in the single image style (resp. identical distribution style) of the underlying copy-protection of pseudorandom function scheme. In both cases, the reduction is simply the identity. □

Corollary 2. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, there exists a point functions copy-protection scheme with correctness and anti-piracy security with respect to both the product and the identical distribution.*

Proof. This result follows directly from Theorem 7. □

6 Unclonable Encryption in the Plain Model

In this section, we introduce the notion of unclonable encryption and present a construction in the plain model. Our construction uses a pseudorandom function copy-protection scheme with anti-piracy security in the identical distribution style (Definition 18) as a black box. Our construction is a symmetric one-time unclonable encryption scheme, which implies the existence of a reusable public key encryption scheme using the transformation of [AK21].

6.1 Definitions

In this section, we define unclonable encryption as well as its correctness and indistinguishable anti-piracy security.

Definition 22 (Symmetric one-time unclonable encryption scheme). *An symmetric one-time unclonable encryption scheme with message space $\mathcal{M}$ is a tuple of algorithms $\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ with the following properties:*

- $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$. *On input a security parameter, the key generation algorithm outputs a key* k.
- $\rho \leftarrow \mathsf{Enc}(\mathsf{k}, m)$. *On input a key* k *and a message* $m \in \mathcal{M}$, *the encryption algorithm outputs quantum ciphertext* ρ.
- $m \leftarrow \mathsf{Dec}(\mathsf{k}, \rho)$. *On input a key* k *and a quantum ciphertext* ρ, *the decryption algorithm outputs a message* m.

Correctness. An unclonable encryption scheme has *correctness* if decrypting a quantum encryption of any message m yields m with overwhelming probability. More precisely:

$$\forall m \in \mathcal{M},\ \Pr\left[\mathsf{Dec}(\mathsf{k}, \rho) = m \ : \ \begin{array}{l} \mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \rho \leftarrow \mathsf{Enc}(\mathsf{k}, m) \end{array}\right] = 1 - \mathsf{negl}(\lambda)$$

Indistinguishable anti-piracy security. We now define indistinguishable anti-piracy security of a symmetric one-time unclonable encryption scheme. This notion is defined through a game in which an adversary is given a quantum encryption of either m_0 or m_1 - two messages that parametrize the game - and is asked to split it such that both shares can be used to guess which message has been encrypted.[4]Note that that although our definition holds for *one-time* unclonable encryption schemes, we can similarly define this notion for *reusable* unclonable encryption schemes by giving the adversary access to an encryption oracle, and then letting them choose the pair of messages (m_0, m_1).

[4] As in the other unclonable primitives' definitions, the decryption procedure used on the share is not necessarily the original one Dec but is provided by the adversary as a quantum circuit.

Definition 23 (Piracy game for a symmetric one-time unclonable encryption scheme). *We define below a* piracy game for symmetric one-time unclonable encryption*, parameterized by a symmetric one-time unclonable encryption scheme* $\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$*, a security parameter* λ*, and a pair of messages* (m_0, m_1)*. This game is between a challenger and an adversary* $\mathcal{A}$*.*

- ***Setup phase:***
  - *The challenger samples* $\mathsf{k} \in \mathsf{KeyGen}(1^\lambda)$ *and* $b \leftarrow\!\!\$\ \{0,1\}$*, and computes* $\rho \leftarrow \mathsf{Enc}(\mathsf{k}, m_b)$*.*
  - *The challenger sends* ρ *to* $\mathcal{A}$*.*
- ***Splitting phase:*** $\mathcal{A}$ *outputs a bipartite quantum state* σ_{12}*, as well as two quantum circuits* U_1 *and* U_2*.*
- ***Challenge phase:*** *For* $i \in \{1,2\}$*, the challenger performs the following test on* (σ_i, U_i)*:*
  - *Run the circuit* U_i *on* $\sigma_i \otimes |\mathsf{k}\rangle\langle\mathsf{k}|$ *and checks whether the outcome is* b*. If so, we say that* (σ_i, U_i) *passes the test, otherwise that it does not.*

$\mathcal{A}$ *wins the game if both* (σ_1, U_1) *and* (σ_2, U_2) *pass the test.*

We denote the random variable that indicates whether an adversary $\mathcal{A}$ *wins the game or not as* $\mathsf{UncEnc} - \mathsf{AP}^{\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle}_{(m_0,m_1)}(1^\lambda, \mathcal{A})$*.*

Definition 24 (Indistinguishable anti-piracy security of an unclonable encryption scheme). *A symmetric one-time unclonable encryption scheme* $\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ *has* indistinguishable anti-piracy security *if, for any pair of messages* (m_0, m_1)*, no QPT adversary can win the piracy game above with probability significantly greater than* $1/2$*.*

More precisely, for any QPT adversary $\mathcal{A}$

$$\Pr\left[\mathsf{UncEnc} - \mathsf{AP}^{\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle}_{(m_0,m_1)}(1^\lambda, \mathcal{A}) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

6.2 Construction

In this subsection, we present a construction of a symmetric one-time unclonable encryption scheme for single-bit messages. Through all the subsection, λ denotes a security parameter and $n(\cdot), m(\cdot)$ are polynomials; whenever it is clear from the context, we note n and m instead of $n(\lambda)$ and $m(\lambda)$. Let $\mathsf{PRF}.\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ be a pseudorandom function copy-protection scheme with input space $\{0,1\}^n$ and output space $\{0,1\}^m$. In addition, we ask the copy-protected pseudorandom function to be extracting with error $2^{-\lambda-1}$ for min-entropy n. Note that the copy-protected pseudorandom function presented in Appendix A has this property.

Construction 3: Unclonable Encryption

$\mathsf{KeyGen}(1^\lambda)$:
- Sample a key $\mathsf{k}_S \leftarrow\!\!\$\ \{0,1\}^n$.
- Return k_S.

$\mathsf{Enc}(\mathsf{k}_S, b)$:
- Sample $\mathsf{k}_P \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$ and compute $\rho_{\mathsf{k}_P} \leftarrow \mathsf{PRF.Protect}(\mathsf{k}_P)$.
- Sample $r \leftarrow\!\!\$\ \{0,1\}^n$; let $c_0 \leftarrow \mathsf{PRF}(\mathsf{k}_P, k_S \oplus r)$ and $c_1 \leftarrow\!\!\$\ \{0,1\}^m$.
- Return $(r, c_b, \rho_{\mathsf{k}_P})$.

$\mathsf{Dec}(\mathsf{k}_S, (r, c, \rho_{\mathsf{k}_P}))$:
- Compute $c^* \leftarrow \mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$.
- Return 0 if $c^* = c$ and 1 otherwise.

Theorem 10. *Assume* $\mathsf{PRF}.\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ *has indistinguishable anti-piracy security in the identical distribution style security (Definition 18). Then Construction 3 has correctness and indistinguishable anti-piracy security.*

Proof of correctness. The correctness comes directly from the correctness and security of the underlying PRF copy-protection scheme. More precisely, $\mathsf{Dec}(\mathsf{k}_S, \mathsf{Enc}(\mathsf{k}_S, 0)) = 1$ means that $\mathsf{PRF.Eval}(\rho_{\mathsf{k}_P}, \mathsf{k}_S \oplus r) \neq \mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$ which happens with negligible probability from the correctness of the PRF copy-protection scheme. And $\mathsf{Dec}(\mathsf{k}_S, \mathsf{Enc}(\mathsf{k}_S, 1)) = 0$ means that $\mathsf{PRF.Eval}(\rho_{\mathsf{k}_P}, \mathsf{k}_S \oplus r) = y$ for a uniformly random y happens with non-negligible probability, which contradicts PRF security.

Proof of indistinguishable anti-piracy security. We proceed in the proof through a sequence of hybrids. For any pair of hybrids (G_i, G_j), we say that G_i is *negligibly close to* G_j if for every QPT adversary $\mathcal{A} \coloneqq (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, the probability that $\mathcal{A}$ wins G_i is negligibly close to the probability that they win G_j.

Game G_0: The first hybrid is the piracy game for our construction.

- **Setup phase:**
  - The challenger samples $\mathsf{k}_S \leftarrow\!\!\$\ \{0,1\}^n$;
  - The challenger samples $\mathsf{k}_P \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$ and computes $\rho_{\mathsf{k}_P} \leftarrow \mathsf{PRF.Protect}(\mathsf{k}_P)$;
  - The challenger samples $r \leftarrow\!\!\$\ \{0,1\}^n$, then sets $c_0 \leftarrow \mathsf{PRF}(\mathsf{k}_P, k_S \oplus r)$ and samples $c_1 \leftarrow\!\!\$\ \{0,1\}^m$;
  - The challenger samples $b \leftarrow\!\!\$\ \{0,1\}$ and sends $(r, c_b, \rho_{\mathsf{k}_P})$ to $\mathcal{A}$.
- **Splitting phase:** $\mathcal{A}$ outputs a bipartite quantum state σ_{12}, as well as two quantum circuits U_1 and U_2.
- **Challenge phase:** For $i \in \{1,2\}$, the challenger performs the following test on (σ_i, U_i):
  - Run the circuit U_i on $\sigma_i \otimes |\mathsf{k}_S\rangle\langle\mathsf{k}_S|$ and checks whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

$\mathcal{A}$ wins the game if both (σ_1, U_1) and (σ_2, U_2) pass the test.

Game G_1: In the second hybrid, we replace c_1 by the pseudorandom function evaluation of a random input. More formally, in the setup phase, we replace $c_1 \leftarrow\!\!\$\ \{0,1\}^m$ by $c_1 \coloneqq \mathsf{PRF}(\mathsf{k}_P, x)$ where $x \leftarrow\!\!\$\ \{0,1\}^m$. As x is sampled uniformly at random, from the extracting property of the underlying pseudorandom function, G_0 is negligibly close to G_1.

Game G_2: In this third hybrid, we replace the random x by $\mathsf{k}'_S \oplus r$ where k'_S is sampled uniformly at random from $\{0,1\}^n$. As $\mathsf{k}'_S \oplus r$ is still uniformly random, this does not change the overall distribution of the game. Thus, G_2 is negligibly close to G_1 (more precisely, it is exactly the same game).

Game G_3: For the third hybrid, instead of sending either $\mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$ or $\mathsf{PRF}(\mathsf{k}_P, \mathsf{k}'_S \oplus r)$ - depending on b - to the adversary in the setup phase, and testing each (U_i, σ_i) on k_S, we send only $\mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$ to the adversary in the setup phase and perform the tests on either k_S or k'_S - still depending on b. Note that this is actually only relabelling, hence the distribution of the game is not changed either. Thus the G_3 has exactly the same distribution as G_2. We describe G_3 more precisely below:

- **Setup phase:**
  - The challenger samples $\mathsf{k}_S, \mathsf{k}'_S \leftarrow\!\!\$\ \{0,1\}^n$;
  - The challenger samples $\mathsf{k}_P \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$ and computes $\rho_{\mathsf{k}_P} \leftarrow \mathsf{PRF.Protect}(\mathsf{k}_P)$;
  - The challenger samples $r \leftarrow\!\!\$\ \{0,1\}^n$, and sends $(r, \mathsf{PRF}(\mathsf{k}_P, k_S \oplus r), \rho_{\mathsf{k}_P})$ to $\mathcal{A}$.
- **Splitting phase:** $\mathcal{A}$ outputs a bipartite quantum state σ_{12}, as well as two quantum circuits U_1 and U_2.
- **Challenge phase:** The challenger samples $b \leftarrow\!\!\$\ \{0,1\}$, sets $\mathsf{k}_0 \coloneqq \mathsf{k}_S$ and $\mathsf{k}_1 \coloneqq \mathsf{k}'_S$, then performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  - Run the circuit U_i on $\sigma_i \otimes |\mathsf{k}_b\rangle\langle\mathsf{k}_b|$ and checks whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

$\mathcal{A}$ wins the game if both (σ_1, U_1) and (σ_2, U_2) pass the test.

We now reduce the game G_3 from the piracy game of the underlying pseudorandom function copy-protection scheme in the identical distribution style. Assume that there exists a QPT adversary $\mathcal{A}$ who wins G_3 with advantage δ. We construct a QPT adversary $\mathcal{B}$ who wins the piracy game of the underlying pseudorandom function copy-protection scheme in the identical distribution style with the same advantage δ. $\mathcal{B}$ behaves in the following way; on input a quantum protected pseudorandom function key ρ_{k} and a pseudorandom function image $y := \mathsf{PRF}(\mathsf{k}, x)$:

- $\mathcal{B}$ sends ρ_{k} and y to $\mathcal{A}$; let $\sigma_{12}, \mathsf{U}_1, \mathsf{U}_2$ denote the outcome.
- $\mathcal{B}$ prepares the circuits $(\mathsf{U}'_1, \mathsf{U}'_2)$ described below using U_1 and U_2 respectively.
- Finally, $\mathcal{B}$ outputs σ_{12} as the shares and $\mathsf{U}'_1, \mathsf{U}'_2$ as the quantum circuits.

Hardcoded: Circuit U_i.
On input quantum state $\sigma_i \otimes |x\rangle\langle x|$ and PRF input x, do the following:

1. Compute $\mathsf{k} := r \oplus x$.
2. Run U_i on (σ_1, k).
3. Return the outcome.

Fig. 3. Circuit U'_i.

The inputs of U_1 and U_2 in circuits U'_1 and U'_2 follow the same distribution as the inputs for U_1 and U_2 in G_3. Thus $\mathcal{B}$ wins the game with the same advantage as $\mathcal{A}$, which concludes the proof.

Corollary 3. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, there exists a symmetric one-time unclonable encryption scheme with correctness and indistinguishable anti-piracy security.*

6.3 Extension to Multi-bits Messages

We describe below a way to extend our scheme to any message space of the form $\{0,1\}^\ell$ where $\ell(\cdot)$ is a polynomial in λ. Our construction encrypts the message bit by bit, but not in an independent way. Indeed, we use the same pseudorandom function key for encrypting all the bits (and hence the same copy-protected pseudorandom function key); and show that this is enough to achieve indistinguishable anti-piracy security.

Construction 4: Unclonable Encryption with Message Space $\{0,1\}^\ell$

$\mathsf{KeyGen}(1^\lambda)$:
- For $i \in [\![1, \ell]\!]$: sample a key $\mathsf{k}_{S,i} \leftarrow\!\!\$\ \{0,1\}^n$.
- Return $\mathsf{k}_S := \left(\mathsf{k}_{S,i}\right)_{i \in [\![1,\ell]\!]}$.

$\mathsf{Enc}(\mathsf{k}_S, m)$:
- Sample $\mathsf{k}_P \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$ and compute $\rho_{\mathsf{k}_P} \leftarrow \mathsf{PRF.Protect}(\mathsf{k}_P)$.
- For $i \in [\![1, \ell]\!]$: sample $r_i \leftarrow\!\!\$\ \{0,1\}^n$ and compute $y_i := \mathsf{k}_{S,i} \oplus r_i$.
- Let $c_{0,i} \leftarrow \mathsf{PRF}(\mathsf{k}_P, y_i)$ and $c_{1,i} \leftarrow\!\!\$\ \{0,1\}^m$.
- Let $r := (r_i)_{i \in [\![1,\ell]\!]}$ and $c_m := (c_{m_i,i})_{i \in [\![1,\ell]\!]}$.
- Return $(r, c_m, \rho_{\mathsf{k}_P})$.

$\mathsf{Dec}(\mathsf{k}_S, (r, c, \rho_{\mathsf{k}_P}))$:
- For $i \in [\![1, \ell]\!]$: compute $y_i := \mathsf{k}_{S,i} \oplus r_i$ and $c_i^* \leftarrow \mathsf{PRF}(\mathsf{k}_P, y_i)$.

– Set $m \in \{0,1\}^\ell$ such that $m_i := 0$ if $c_i^* = c_i$ and $m_i := 1$ otherwise.

– Return m

Theorem 11. *Assume* $\mathsf{PRF}.\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ *has indistinguishable anti-piracy security in the identical distribution style. Then Construction 4 has correctness and indistinguishable anti-piracy security.*

Proof. The correctness proof is the same as for the single-bit version: it relies on correctness and security of the underlying pseudorandom function copy-protection scheme.

We give a short summary of the proof of anti-piracy security, as it uses a usual hybrids argument. By doing small hops, we show that if no adversary can distinguish between the encryption of two messages differing on only one index, then no adversary can distinguish between the encryption of two messages differing on only two indices, and so on and so forth until finally showing that no adversary can distinguish between the encryption of two messages differing on all indices. It then remains to show that no adversary can distinguish between the encryption of two messages differing only on one index, which follows from the anti-piracy security of the pseudorandom function copy-protection scheme. □

Corollary 4. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, there exists a public-key reusable unclonable encryption scheme with correctness and indistinguishable anti-piracy security.*

Proof. In [AK21, Section 5], the authors present a way to construct a public-key reusable one-time unclonable encryption scheme from any symmetric one-time unclonable encryption scheme with indistinguishable anti-piracy security, using a (post-quantum) symmetric encryption scheme with pseudorandom ciphertexts and a (post-quantum) single-key public-key functional encryption scheme. We refer the interested reader to this paper for a description of the construction. □

References

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

AB23. Prabhanjan Ananth and Amit Behera. A modular approach to unclonable cryptography, 2023. `https://arxiv.org/abs/2311.11890`.

AK21. Prabhanjan Ananth and Fatih Kaleoglu. Unclonable encryption, revisited. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 299–329. Springer, Heidelberg, November 2021.

AKL+22. Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. On the feasibility of unclonable encryption, and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 212–241. Springer, Heidelberg, August 2022.

AKL23. Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. Cloning games: A general framework for unclonable primitives. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 66–98. Springer, Heidelberg, August 2023.

AL21. Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 501–530. Springer, Heidelberg, October 2021.

BB20. Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *arXiv preprint arXiv:2003.06557*, 2020.

BGI+01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

BGI14. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Heidelberg, March 2014.

BJL+21. Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram. Secure software leasing without assumptions. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 90–120. Springer, Heidelberg, November 2021.

BL20. Anne Broadbent and Sébastien Lord. Uncloneable Quantum Encryption via Oracles. 158:4:1–4:22, 2020.
BW13. Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Heidelberg, December 2013.
CG23. Andrea Coladangelo and Sam Gunn. How to use quantum indistinguishability obfuscation. Cryptology ePrint Archive, Paper 2023/1756, 2023. `https://eprint.iacr.org/2023/1756`.
CHV23. Céline Chevalier, Paul Hermouet, and Quoc-Huy Vu. Semi-quantum copy-protection and more. Cryptology ePrint Archive, Report 2023/244, 2023. `https://eprint.iacr.org/2023/244`.
CLLZ21. Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 556–584, Virtual Event, August 2021. Springer, Heidelberg.
CMP20. Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. Cryptology ePrint Archive, Report 2020/1194, 2020. `https://eprint.iacr.org/2020/1194`.
CV22. Eric Culf and Thomas Vidick. A monogamy-of-entanglement game for subspace coset states. *Quantum*, 6:791, September 2022.
GGM84. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.
HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
JMRW16. Nathaniel Johnston, Rajat Mittal, Vincent Russo, and John Watrous. Extended non-local games and monogamy-of-entanglement games. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472(2189), 2016.
KPTZ13. Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 669–684. ACM Press, November 2013.
LLQZ22. Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 294–323. Springer, Heidelberg, November 2022.
TFKW13. Marco Tomamichel, Serge Fehr, Jedrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, oct 2013.
Wie83. Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983.
Wil11. Mark M Wilde. From classical to quantum shannon theory. *arXiv preprint arXiv:1106.1445*, 2011.
WZ82. William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

A Construction of Pseudorandom Function Copy-Protection

We present below the construction of pseudorandom function copy-protection scheme of [CLLZ21], and show that it has anti-piracy security both in the product challenge bits style and in the identical distribution style.

Construction. Let n be a polynomial in λ; we define ℓ_0, ℓ_1, ℓ_2 such that $n = \ell_0 + \ell_1 + \ell_2$ and $\ell_2 - \ell_0$ is large enough. For this construction, we need three pseudorandom functions:

- A puncturable extracting pseudorandom function $\mathsf{PRF}_1 : \mathcal{K}_1 \times \{0,1\}^n \to \{0,1\}^m$ with error $2^{-\lambda-1}$ for min-entropy n, where m is a polynomial in λ and $n \geq m + 2\lambda + 4$.
- A puncturable injective pseudorandom function $\mathsf{PRF}_2 : \mathcal{K}_2 \times \{0,1\}^{\ell_2} \to \{0,1\}^{\ell_1}$ with failure probability $2^{-\lambda}$, with $\ell_1 \geq 2\ell_2 + \lambda$.
- A puncturable pseudorandom function $\mathsf{PRF}_3 : \mathcal{K}_3 \times \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_2}$.

Construction 5: Pseudorandom Function Copy-Protection
$\mathsf{Protect}(1^\lambda, \mathsf{k})$:

– Sample ℓ_0 random coset states $\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}$, where each subspace $A_i \subseteq \mathbb{F}_2^n$ if of dimension $\frac{n}{2}$.
– For each coset state $|A_{i,s_i,s'_i}\rangle$, prepare the obfuscated membership programs $\mathsf{P}_i^0 = \mathsf{iO}(A_i + s_i)$ and $\mathsf{P}_i^1 = \mathsf{iO}(A_i^\perp + s'_i)$.
– Sample $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$.
– Prepare the program $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$, where P is described in Figure 4.
– Return $\rho_\mathsf{k} := \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \widehat{\mathsf{P}}\right)$.

$\mathsf{Eval}(1^\lambda, \rho_\mathsf{k}, x)$:
– Parse $\rho_\mathsf{k} = \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \widehat{\mathsf{P}}\right)$.
– Parse x as $x := x_0\|x_1\|x_2$.
– For each $i \in [\![1,\ell_0]\!]$, if $x_{0,i} = 1$, apply $\mathsf{H}^{\otimes n}$ to $|A_{i,s_i,s'_i}\rangle$; if $x_{0,i} = 0$, leave the state unchanged.
– Let σ be the resulting state (which can be interpreted as a superposition over tuples of ℓ_0 vectors). Run $\widehat{\mathsf{P}}$ coherently on input x and σ, and measure the final output register to obtain y.
– Return y.

Hardcoded: Keys $(\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3) \in \mathcal{K}_1 \times \mathcal{K}_2 \times \mathcal{K}_3$, programs $\mathsf{P}_i^0, \mathsf{P}_i^1$ for all $i \in [\![1,\ell_0]\!]$.
On input $x = x_0\|x_1\|x_2$ and vectors $v_0, v_1, \cdots, v_{\ell_0}$ where each $v_i \in \mathbb{F}_2^n$, do the following:

1. (**Hidden Trigger Mode**) If $\mathsf{PRF}_3(\mathsf{k}_3, x_1) \oplus x_2 = x_0\|\mathsf{Q}'$ and $x_1 = \mathsf{PRF}_2(\mathsf{k}_2, x_0\|\mathsf{Q}')$: treat Q' as a classical circuit and output $\mathsf{Q}'(v_1, \cdots, v_{\ell_0})$.
2. (**Normal Mode**) If for all $i \in [\![1,\ell_0]\!]$, $\mathsf{P}_i^{x_i}(v_i) = 1$, then output $\mathsf{PRF}_1(\mathsf{k}_1, x)$. Otherwise, output $\perp$.

Fig. 4. Program P.

A.1 Proof of Indistinguishable Anti-Piracy Security in the Product Challenge Bits Style

In this subsection, we prove that the construction above has indistinguishable anti-piracy security in the product challenge bits style. This proof and the next one (for the identical distribution style) both adapt the proof of [CLLZ21, Theorem 7.12] to our settings; some parts are taken verbatim. We first introduce some notations, a procedure and a lemma that we use in the two proofs.

Notations. In the proof, we sometimes parse $x \in \{0,1\}^n$ as (x_0, x_1, x_2) such that $x = x_0\|x_1\|x_2$ (where $\cdot\|\cdot$ is the concatenation operator) and the length of x_i is ℓ_i for $i \in \{0,1,2\}$.

We proceed with both proofs through a sequence of hybrids. For any pair of hybrids (G_i, G_j), we say that G_i is *negligibly close to* G_j if for every QPT adversary $\mathcal{A} := (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, the probability that $\mathcal{A}$ wins G_i is negligibly close to the probability that they win G_j.

Procedure. We define the $\mathsf{GenTrigger}$ procedure (Figure 5) which, given an input's prefix x_0 and a pseudorandom function image y returns a so-called *trigger input* x' that: passes the "Hidden Trigger" condition of the program P. Although this procedure also takes as input pseudorandom function keys $\mathsf{k}_2, \mathsf{k}_3$ and coset states descriptions, we will abuse notation and only write $\mathsf{GenTrigger}(x_0, y)$ when it is clear from

the context. We will also write $\mathsf{GenTrigger}(x_0; \mathsf{Q})$ - where Q is a program - to denote the same procedure using Q instead of the program normally defined in step 1.

Given as input $x_0 \in \{0,1\}^{\ell_0}$, $y \in \{0,1\}^m$, $\mathsf{k}_2, \mathsf{k}_3 \in \mathcal{K}_2 \times \mathcal{K}_3$ and cosets $\{A_{i,s_i,s'_i}\}_{i\in[\![1,\ell_0]\!]}$:

1. Let Q be the program which, given $v_0, \ldots, v_{\ell_0}$, returns y if $R_i^{x_{0,i}}(v_i) = 1$ for all i or $\perp$ otherwise.
2. $x'_1 \leftarrow \mathsf{PRF}_2(\mathsf{k}_2, x_0 \| \mathsf{Q})$;
3. $x'_2 \leftarrow \mathsf{PRF}_3(\mathsf{k}_3, x'_1) \oplus (x_0 \| \mathsf{Q})$;
4. Return $x_0 \| x'_1 \| x'_2$.

Fig. 5. GenTrigger procedure.

Trigger's Inputs Lemma. The following lemma is taken from [CLLZ21, Lemma 7.17].

Lemma 7. *Assuming post-quantum* iO *and one-way functions, any efficient QPT algorithm* $\mathcal{A}$ *cannot win the following game with non-negligible advantage:*

- *A challenger samples* $\mathsf{k}_1 \leftarrow \mathsf{Setup}(1^\lambda)$ *and prepares a quantum key* $\rho_\mathsf{k} \coloneqq (\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,l_0]\!]}, \mathsf{iO}(\mathsf{P}))$ *(recall that* P *has keys* $\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3$ *hardcoded).*
- *The challenger then samples a random input* $x_1 \leftarrow \{0,1\}^n$*; let* $y_1 \leftarrow \mathsf{PRF}_1(\mathsf{k}_1, x_1)$ *and computes* $x'_1 \leftarrow \mathsf{GenTrigger}(x_{1,0}, y_1)$.
- *Similarly, the challenger samples a random input* $x_2 \leftarrow \{0,1\}^n$*; let* $y_2 \leftarrow \mathsf{PRF}_1(\mathsf{k}_1, x_2)$ *and computes* $x'_2 \leftarrow \mathsf{GenTrigger}(x_{2,0}, y_1)$.
- *The challenger flips a coin* b*, and sends either* $(\rho_\mathsf{k}, x_1, x_2)$ *or* $(\rho_\mathsf{k}, x'_1, x'_2)$ *to* $\mathcal{A}$*, depending on the value of the coin.*

$\mathcal{A}$ *wins if it guesses* b *correctly.*

Game G_0: This is the piracy game in the product challenge bits style of the pseudorandom function copy-protection protocol.

- **Setup phase:**
  - The challenger samples ℓ_0 random cosets $\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}$, and prepares the associated coset states $\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}$ and the obfuscated membership programs $\{(\mathsf{P}_i^0, \mathsf{P}_i^1)\}_{i\in[\![1,\ell_0]\!]}$.
  - The challenger samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$ and generates the obfuscated program $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$.
  - The challenger samples $x \leftarrow\!\!\$\ \{0,1\}^n$ and computes $y := \mathsf{PRF}_1(\mathsf{k}_1, x)$.
  - Finally, the challenger sends $\rho_\mathsf{k} := \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \widehat{\mathsf{P}}\right)$ and y to $\mathcal{A}$.
- **Splitting phase:** $\mathcal{A}$ outputs a bipartite quantum state σ_{12}, and two quantum circuits U_1 and U_2.
- **Challenge phase (product challenge bits style):** The challenger performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  - sample $b \leftarrow\!\!\$\ \{0,1\}$;
  - set $x_0 := x$ and sample $x_1 \leftarrow\!\!\$\ \{0,1\}^n$.
  - run the circuit U_i on $\sigma_i \otimes |x_b\rangle\langle x_b|$ and check whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

- **Challenge phase (identical distribution style):** The challenger samples a bit $b \leftarrow\!\!\$\ \{0,1\}$ and a fake input $x_1 \leftarrow\!\!\$\ \{0,1\}^n$, then performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  - set $x_0 := x$;
  - run the circuit U_i on $\sigma_i \otimes |x_b\rangle\langle x_b|$ and check whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

  $\mathcal{A}$ wins the game if both tests pass.

Game G_1: In this game, we replace the challenges x_0, x_1 by their trigger inputs in both tests. More precisely, the challenge phases becomes the following.

- **Challenge phase (product challenge bits style):** The challenger performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  - sample $b \leftarrow\!\!\$\ \{0,1\}$;
  - set $x_0 := x$ and sample $x_1 \leftarrow\!\!\$\ \{0,1\}^n$;
  - set $y_0 := y$ and $y_1 := \mathsf{PRF}(\mathsf{k}_1, x_1)$;
  - compute $x'_0 \leftarrow \mathsf{GenTrigger}(x_0; y_0)$ and $x'_1 \leftarrow \mathsf{GenTrigger}(x_1; y_1)$;
  - run the circuit U_i on $\sigma_i \otimes |x'_b\rangle\langle x'_b|$ and check whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.
- **Challenge phase (identical distribution style):** The challenger samples a bit $b \leftarrow\!\!\$\ \{0,1\}$ and a fake input $x_1 \leftarrow\!\!\$\ \{0,1\}^n$, then performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  - set $y_0 := y$ and $y_1 := \mathsf{PRF}(\mathsf{k}_1, x_1)$;
  - compute $x'_0 \leftarrow \mathsf{GenTrigger}(x; y_0)$ and $x'_1 \leftarrow \mathsf{GenTrigger}(x_1; y_1)$;
  - run the circuit U_i on $\sigma_i \otimes |x'_b\rangle\langle x'_b|$ and check whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

The trigger's inputs lemma (Lemma 7) implies that G_1 is negligibly close to G_0.

Game G_2: In this game, we replace y (in setup phase) and y_1 (in both tests) by uniformly random strings. Since all the inputs have enough min-entropy $\ell_1 + \ell_2 \geq m + 2\lambda + 4$ and PRF_1 is extracting, the outcomes are statistically close to independently random outcomes. Thus G_2 is negligibly close to G_1.

Game G_3: This game has exactly the same distribution as that of G_2. We only change the order in which some values are sampled, and recognize that certain procedures become identical to encryptions in the single-decryptor encryption scheme $\langle \mathsf{SD.Setup}, \mathsf{SD.QKeyGen}, \mathsf{SD.Enc}, \mathsf{SD.Dec}\rangle$ from Construction 1. Thus, the probability of winning in G_3 is the same as in G_2.

- **Setup phase:**
  - The challenger runs $\mathsf{SD.Setup}(1^\lambda)$ to obtain ℓ_0 random cosets $\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}$, the associated coset states $\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}$ and the obfuscated membership programs $\{(\mathsf{P}^0_i, \mathsf{P}^1_i)\}_{i\in[\![1,\ell_0]\!]}$. Let $\rho_{\mathsf{sk}} := \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}$.
  - The challenger samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$ and generates the obfuscated program $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$.
  - The challenger samples $y \leftarrow\!\!\$\ \{0,1\}^m$ and sends $\rho_{\mathsf{k}} := \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \widehat{\mathsf{P}}\right)$ and y to $\mathcal{A}$.
- **Splitting phase:** $\mathcal{A}$ outputs a bipartite quantum state σ_{12}, and two quantum circuits U_1 and U_2.
- **Challenge phase (product challenge bits style):** The challenger samples a random bitstring r to be used in the encryption and performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  - sample $b \leftarrow\!\!\$\ \{0,1\}$
  - set $y_0 := y$ and sample $y_1 \leftarrow\!\!\$\ \{0,1\}^m$;
  - compute $(x_b, \mathsf{Q}) \leftarrow \mathsf{SD.Enc}(\mathsf{pk}, y_b; r)$;

– compute $x'_b \leftarrow \mathsf{GenTrigger}(x_{b,0}; \mathsf{Q})$ where $x_{b,0}$ are the first ℓ_0 bits of x_b;

– run the circuit U_i on $\sigma_i \otimes |x'_b\rangle\langle x'_b|$ and checks whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

- **Challenge phase (identical distribution style):** The challenger samples a bit $b \leftarrow\!\!\$\ \{0,1\}$ and a random bitstring r to be used in the encryption, then performs the following test on (σ_i, U_i) for $i \in \{1,2\}$:
  – set $y_0 := y$ and sample $y_1 \leftarrow\!\!\$\ \{0,1\}^m$;
  – compute $(x_b, \mathsf{Q}) \leftarrow \mathsf{SD.Enc}(\mathsf{pk}, y_b; r)$;
  – compute $x'_b \leftarrow \mathsf{GenTrigger}(x_{b,0}; \mathsf{Q})$ where $x_{b,0}$ are the first ℓ_0 bits of x_b;
  – run the circuit U_i on $\sigma_i \otimes |x'_b\rangle\langle x'_b|$ and checks whether the outcome is b. If so, we say that (σ_i, U_i) passes the test, otherwise that it does not.

  $\mathcal{A}$ wins the game if both tests pass.

Reduction from single-decryptor's piracy game for the product challenge bits style. We reduce the game G_3 in the product challenge bits style to the piracy game of the underlying single-decryptor in the identical randomness style. Assume that there exists a QPT adversary $\mathcal{A}$ who wins the last hybrid G_3 in the product challenge bits style with advantage δ. We construct a QPT adversary $\mathcal{B}$ who wins the piracy game of the single-decryptor scheme of Construction 1 in the identical randomness style with the same advantage δ. $\mathcal{B}$ behaves in the following way; on input a quantum key ρ_{sk} and the associated public key pk:

- $\mathcal{B}$ samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$ and use these keys and pk to prepare the obfuscated program $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$.
- Then $\mathcal{B}$ samples $y, y_1, y_2 \leftarrow\!\!\$\ \{0,1\}^m$ and runs $\mathcal{A}$ on $(\rho_{\mathsf{sk}}, \widehat{\mathsf{P}}, y)$; let $(\sigma_{12}, \mathsf{U}_1, \mathsf{U}_2)$ denote the outcome.
- $\mathcal{B}$ prepares the circuits $(\mathsf{U}'_1, \mathsf{U}'_2)$ described below using U_1 and U_2 respectively.
- Finally, $\mathcal{B}$ outputs $\sigma'_1 := \sigma_1 \otimes |\mathsf{k}_2\rangle\langle\mathsf{k}_2| \otimes |\mathsf{k}_3\rangle\langle\mathsf{k}_3|$ and $\sigma'_2 := \sigma_2 \otimes |\mathsf{k}_2\rangle\langle\mathsf{k}_2| \otimes |\mathsf{k}_3\rangle\langle\mathsf{k}_3|$ as the shares, (y, y_1) and (y, y_2) as the two message pairs, and $(\mathsf{U}'_1, \mathsf{U}'_2)$ as the quantum circuits.

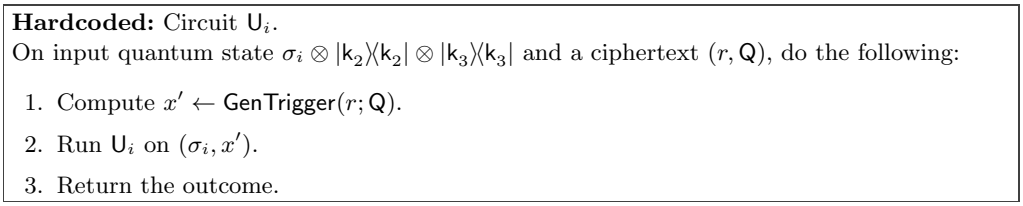

Fig. 6. Circuit U'_i.

The adversary $\mathcal{B}$ perfectly simulates $\mathcal{A}$, and thus $\mathcal{B}$ breaks the anti-piracy security of the single-decryptor scheme with the same probability δ, which completes the proof.

Reduction from single-decryptor's piracy game for the identical distribution style. We reduce the game G_3 in the identical distribution style to the piracy game of the underlying single-decryptor with respect to the identical distribution. Assume that there exists a QPT adversary $\mathcal{A}$ who wins the last hybrid G_3 in the identical distribution style with advantage δ. We construct a QPT adversary $\mathcal{B}$ who wins the piracy game of the single-decryptor scheme of Construction 1 with respect to the identical distribution with the same advantage δ. $\mathcal{B}$ behaves in the following way; on input a quantum key ρ_{sk} and the associated public key pk:

- $\mathcal{B}$ samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$ and use these keys and pk to prepare the obfuscated program $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$.

- Then $\mathcal{B}$ samples $y, y_1 \leftarrow\!\!\$\ \{0,1\}^m$ and runs $\mathcal{A}$ on $(\rho_{\mathsf{sk}}, \widehat{\mathsf{P}}, y)$; let $(\sigma_{12}, \mathsf{U}_1, \mathsf{U}_2)$ denote the outcome.
- $\mathcal{B}$ prepares the circuits $(\mathsf{U}'_1, \mathsf{U}'_2)$ described below using U_1 and U_2 respectively.
- Finally, $\mathcal{B}$ outputs $\sigma'_1 := \sigma_1 \otimes |\mathsf{k}_2\rangle\langle\mathsf{k}_2| \otimes |\mathsf{k}_3\rangle\langle\mathsf{k}_3|$ and $\sigma'_2 := \sigma_2 \otimes |\mathsf{k}_2\rangle\langle\mathsf{k}_2| \otimes |\mathsf{k}_3\rangle\langle\mathsf{k}_3|$ as the shares, (y, y_1) and (y, y_1) as the two message pairs, and $(\mathsf{U}'_1, \mathsf{U}'_2)$ as the quantum circuits.

The adversary $\mathcal{B}$ perfectly simulates $\mathcal{A}$, and thus $\mathcal{B}$ breaks the anti-piracy security of the single-decryptor scheme with the same probability δ, which completes the proof.