Towards 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. Most known constructions rely on the quantum random oracle model (as opposed to the plain model), in which all parties have access in superposition to a powerful random oracle. 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. A core ingredient of these protocols is the so-called monogamy-of-entanglement property. Such games allow quantifying the correlations between the outcomes of multiple non-communicating parties sharing entanglement in a particular context. Specifically, we define the games between a challenger and three players in which the first player is asked to split and share a quantum state between the two others, who are then simultaneously asked a question and need to output the correct answer.
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 progress towards the aforementioned goals. However, it is not sufficient on its own, and we present two conjectures that would allow first to show that copy-protection of point functions exists in the plain model, with different challenge distributions (including arguably the most natural ones), and then that unclonable encryption with unclonable indistinguishability security exists in the plain model. We believe that our new monogamy-of-entanglement to be of independent interest, and it could be useful in other applications as well.

1 Introduction

1.1 Unclonable Cryptography

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 states that, given a copy-protected quantum program, no adversary can produce two (possibly entangled) states that both can be used to compute this program. Testing whether these two states can compute the program is done by sampling two challenges input for the program from a certain *challenge distribution*. Then, informally, each state is used as a quantum program and run on the corresponding challenge to produce some outcome; and the test passes if this outcome is the one that would be output by the program on this input.

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 almost all in the quantum random oracle model. The only known copy-protection for point functions scheme in the plain model (without random oracle or another setup assumption) 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] based on a previous work of Gottesman [Got02], 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?*

1.2 Monogamy Games

In order to understand better the difficulty of achieving such goals, we first recall the security definitions of these primitives, called anti-piracy security. This notion is defined through a piracy game, in which Alice is given a certain quantum state. Alice must then split this state and share it between two other adversaries, Bob and Charlie. Then, Bob and Charlie receive a challenge and must guess the correct answer.

This security can be proven through the use of monogamy games, which are games whose winning probability is restricted by the monogamy-of-entanglement; in order to win the game with the highest probability, the players have to leverage the power of entanglement in the best possible way, but monogamy-of-entanglement prevent them to win with probability 1. As a simple example, consider the following game, studied in particular in [TFKW13]. This game is between a challenger and two players, Bob and Charlie. Bob and Charlie are first asked to prepare a tripartite quantum state ρ_{ABC}; then to send the register A to the challenger; and finally to share the remaining registers between themselves. From this step, Bob and Charlie cannot communicate anymore. Then, the challenger measures each qubit of this register in a random basis - either computational or diagonal - and reveal the bases to Bob and Charlie. Bob and Charlie are now both asked to guess the outcome of the challenger's measurement. The maximum winning probability of this game is $\frac{1}{2} + \frac{1}{2\sqrt{2}}$.

In the following, we consider games with a slightly different structure: the games are between a challenger and three players, Alice, Bob, and Charlie - where Bob and Charlie cannot communicate. The challenger first sends a quantum state to Alice who has to split it and share it between Bob and Charlie. Bob and Charlie are then asked a question and both need to return the correct answer. Interestingly, in these games, the questions asked Bob and Charlie would have been easily answered by Alice before she splits the state. We are indeed interested in how well she can split the state to preserve as much as possible the information necessary to answer correctly in each share.

In [CLLZ21], the authors defined the *coset states*: quantum states of the form $|A_{s,s'}\rangle \coloneqq \sum_{x \in A} (-1)^{\langle x, s' \rangle} |x + s\rangle$ (up to renormalization) 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 regular coset $A + s$ and its dual coset $A^\perp + s'$, in the sense that measuring a coset state in the computational basis yields a random vector in the regular coset; and measuring it in the diagonal basis yields a vector in the dual coset. The coset states feature a so-called *strong monogamy-of-entanglement property* (proven in [CV22]). This property states that no adversaries Alice, Bob and Charlie can win the following monogamy game with non-negligible probability. Given a random coset state $|A_{s,s'}\rangle$, Alice has to split the state and share it between Bob and Charlie. Bob and Charlie then receive the description of the subspace A as the question, and are asked to return a vector in the regular coset $A + s$ for Bob, and a vector in the dual coset $A^\perp + s'$ for Charlie.

1.3 Our contributions

Unfortunately, these monogamy games are not adapted to some distributions, specifically the identical and product distributions, where the elements drawn can be equal. To solve this issue, we present in this game a monogamy game that we call *monogamy game with identical basis.*

Informally, in this game, Bob and Charlie are not asked to return a vector belonging to different cosets (the regular coset for Bob and the dual coset for Charlie), but are instead instructed to return a vector belonging to the same coset (both in the regular coset or both in the dual coset).

Of course, without any additional constraints, Alice could simply measure the coset state in, say, the computational basis, and forward the outcome to both Bob and Charlie. The latter could in turn simply return this outcome and always answer correctly. To prevent such a trivial strategy, the challenger instructs Bob and Charlie on the basis in which the vectors they return must belong. More precisely, the challenger sends as a question the subspace description A as for the [CLLZ21] game, but also a bit b. The expected vectors must then both belong to the regular coset if $b = 0$, or in the dual coset if $b = 1$. Crucially, this basis b is sampled and revealed to the adversaries *after* Alice splits the state. Otherwise, she could simply measure the state in the computational or the diagonal basis depending on the value of b, and forward the outcome to Bob and Charlie.

We prove that the winning probability of this game is at most negligibly higher than $1/2$, which corresponds to the trivial strategy in which Alice always measures the coset state in the computational basis and forwards the outcome to Bob and Charlie, who in turn return it. An illustration of this game is depicted in Figure 1. This new monogamy-of-entanglement (MoE) property of coset states might be of independent interest.[1]

Unfortunately, for reasons detailed in Section 4, this new monogamy game is not sufficient to answer the two questions above affirmatively. We also need that the existence of a compute-and-compare obfuscator [CLLZ21] is still true in a non-local setting. Admitting this conjecture, 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.

Concurrent and independent work. The first version of this paper appeared concurrently and independently with two other works considering 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.

After posting the first version of our paper online, we have had discussions with the authors of [AB23]. We acknowledge that the idea of introducing conjectures was inspired by the work of Ananth and Behera [AB23]. We compare [AB23]'s conjectures with ours in Section 4.

1.4 Technical Overview

A new monogamy-of-entanglement game of coset states. In the heart of our results is a new monogamy-of-entanglement property of coset states, drawing inspiration from previous works [CLLZ21,TFKW13]. A coset state is a quantum state of the form $|A_{s,s'}\rangle := \frac{1}{\sqrt{|A|}} \sum_{x \in A} (-1)^{\langle x,s' \rangle} |x+s\rangle$ 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 [CLLZ21]: given a random coset state $|A_{s,s'}\rangle$, no adversary - Alice - can split the state and share it to two other non-communicating adversaries - Bob and Charlie - such that, given the description of the subspace A, Bob returns a vector in the coset $A+s$ and Charlie a vector in the dual $A^\perp + s'$.

In this paper, we introduce a new variant of the monogamy-of-entanglement property of coset states. In this variant, Bob and Charlie 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 Alice. Crucially Alice 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 Figure 1. We will 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 Alice always measures the coset state in the computational basis and forwards the outcome to both Bob and Charlie, who in turn output it. We will also show that the winning probability of this game can be made negligible by parallel repetition (see Section 3.6).

We now give a sketch of the proof for this new monogamy-of-entanglement game. For simplicity, we describe the proof of the *BB84 version* of our new monogamy-of-entanglement game, since 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 Alice, and Bob and Charlie are given the basis θ and a random bit b. To win the game, Bob and Charlie 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 be 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

[1] Recently, [CGLZR23] also presented a new version of monogamy-of-entanglement game, using a similar idea. We discuss the differences between their version and ours in Section 3.5.

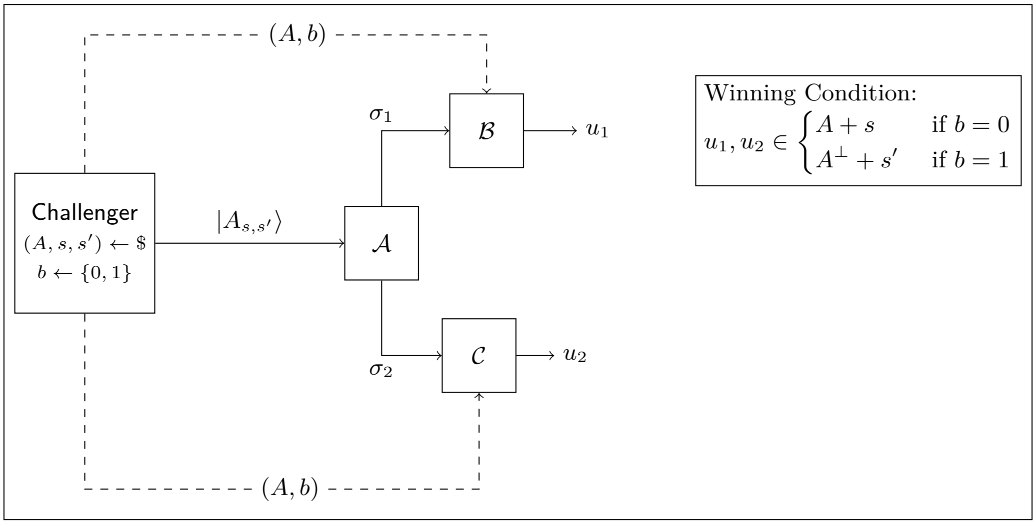


Fig. 1. Monogamy-of-Entanglement Game with Identical Basis (Coset Version). Remark that, in the original monogamy-of-entanglement game for coset states [CLLZ21], the challenger does not sample b, hence there is no b sent to $\mathcal{B}$ and $\mathcal{C}$, and the winning condition is $(u_1 \in A + s) \wedge \left(u_2 \in A^\perp + s'\right)$.

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 $p_{win} = \mathbb{E}_{\theta,b}\left[\mathsf{Tr}\left(\Pi_{\theta,b}\ \rho_{012}\right)\right]$. Then, we show the following upper-bound:

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

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.

Anti-piracy security. We now describe how this new monogamy-of-entanglement property might allow us to obtain new constructions of copy-protection and unclonable encryption. We note that we did not manage to prove security of our constructions from standard assumptions, and they are indeed based on this new monogamy game and a conjecture that we also introduce in this work. 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.

A piracy game is formalized as a security experiment against a triple of cloning adversaries Alice, Bob, and Charlie. Alice receives a copy-protected program $\rho_f := \mathsf{Protect}(f)$, which can be used to evaluate a classical function f, prepares a bipartite state, and sends each half of the state to the two other non-communicating adversaries Bob and Charlie. In the challenge phase, Bob and Charlie receive inputs c_1, c_2, sampled from a challenge distribution and are asked to output b_1, b_2. The adversaries 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:* Bob and Charlie 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:* Bob and Charlie 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:* Bob and Charlie 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 techniques that might 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 5 and Section 6 for more details.

Challenges when proving anti-piracy security w.r.t the product distribution. 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 compute-and-compare 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 adversaries are 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 first adversary Alice 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 adversaries can win the game trivially.

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 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 $i \in [1, n]$. In the proof of [CLLZ21], where the challenge instances given to the two non-communicating adversaries Bob and Charlie are sampled independently, this step can be done by using extracting compute-and-compare obfuscation technique. However, in our case, we face a new problem that now the extraction needs to be done simultaneously where the two challenges are correlated. To remedy the issue, we propose a new conjecture on simultaneous extracting from compute-and-compare obfuscation. We note that weaker version of this conjecture has been proven in [CLLZ21].

Simultaneously extracting from compute-and-compare obfuscation conjecture. Assuming iO and (sub-exponentially) hardness of LWE, for (sub-exponentially) unpredictable distribution $\mathcal{D}$, there exists a compute-and-compare obfuscator [CLLZ21]. We are interested in whether this result still holds in a non-local context. More precisely, consider the two following tasks, which we call *simultaneous distinguishing* and *simultaneous predicting*. Simultaneous predicting asks two players, Bob and Charlie, given a function associated to a compute-and-compare program, and a quantum state as auxiliary information on the program, to output the associated lock value. Crucially, the challenge given to Bob and Charlie might be correlated. In simultaneous distinguishing, Bob and Charlie are given either an obfuscated compute-and-compare program, or the outcome of a simulator on this program's parameters. As in simultaneous predicting, they are also given a quantum state each, but here, they are asked to tell whether they received the obfuscated program, or the simulated one. Our conjecture essentially says that simultaneous predicting *implies* simultaneous distinguishing, for certain challenge distributions.

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 Alice receives a quantum encryption of a bit b, prepares a bipartite state, and sends each half of the state to two non-communicating adversaries Bob and Charlie. In the challenge phase, Bob and Charlie both receive the decryption key k and are asked to output b_1, b_2. Alice, Bob, and Charlie 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 Bob and Charlie, 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.

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 $c > 0$. The notation $\mathsf{subexp}(\lambda)$ denotes a sub-exponential function.

When sampling uniformly at random a value x from a set $\mathcal{S}$, we employ the notation $x \leftarrow\$ \mathcal{S}$. When sampling a value x from a probabilistic algorithm $\mathcal{A}$, or from a distribution $\mathcal{D}$, we employ the notation $a \leftarrow \mathcal{A}$, or $a \leftarrow \mathcal{D}$.

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. When we write that an algorithm $\mathcal{A}$ is "efficient", we mean that $\mathcal{A}$ is QPT. We note $\mathcal{A}(x; r)$ to denote that we run $\mathcal{A}$ on input x with random coins $r \in \{0,1\}^{\mathsf{poly}(\lambda)}$ as the random tape. In the context of security games, we abuse the notations and sometimes write (QPT) adversary instead of (QPT) algorithm. We also sometimes write that a QPT algorithm is run on a classical input x instead of writing that it is run on $|x\rangle\langle x|$.

2.2 Distributions

We define two families of distributions that we often consider in this paper.

Definition 1 (Uniform Distribution). *Let S_λ be any set, and $\lambda \in \mathbb{N}$. We write that a distribution $\mathcal{D}_\lambda$ over $S_\lambda \times S_\lambda$ is* uniform *if it yields pairs of the form (x_1, x_2) where x_1 and x_2 are independently and uniformly sampled from S_λ.*

Similarly, we write that a family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ is uniform if all the $\mathcal{D}_\lambda$ are uniform.

Definition 2 (Identical Distribution). *Let S_λ be any set, and $\lambda \in \mathbb{N}$. We write that a distribution $\mathcal{D}_\lambda$ over $S_\lambda \times S_\lambda$ is* identical *if it yields pairs of the form (x, x) where x uniformly sampled from S_λ.*

Similarly, we write that a family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ is identical if all the $\mathcal{D}_\lambda$ are identical.

2.3 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 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.

2.4 Indistinguishable Obfuscation

Definition 3 (Indistinguishability Obfuscator [BGI+01]). *A uniform* PPT *machine* iO *is called an indistinguishability obfuscator for a classical circuit class* $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *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*
$$\mathsf{Adv}^{\mathsf{io}}(\lambda, \mathcal{A}) \coloneqq |\Pr[\mathcal{D}(\mathsf{iO}(\lambda, C_0)) = 1] - \Pr[\mathcal{D}(\mathsf{iO}(\lambda, C_1)) = 1]| \leq \mathsf{negl}(\lambda).$$

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

2.5 Compute-and-Compare Obfuscation

Definition 4 (Compute-and-Compare Programs). *Given a function* $f : \{0,1\}^n \to \{0,1\}^m$ *along with a lock value* $y \in \{0,1\}^m$ *and a message* $m \in \mathcal{M}$*, we define the compute-and-compare program:*
$$\mathsf{CC}[f, y, m](x) \coloneqq \begin{cases} m & \textit{if } f(x) = y, \\ \bot & \textit{otherwise} . \end{cases}$$

When the function, lock value, and message of a compute-and-compare program are not useful in the context, we will sometimes simply write CC *in lieu of* $\mathsf{CC}[f, y, m]$*.*

Definition 5 (Unpredictable Distribution). *Let* $\mathcal{D} \coloneqq \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a family of distributions 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*
$$\Pr\big[\mathcal{A}(1^\lambda, f, \mathsf{aux}) = y \;:\; (\mathsf{CC}[f, y, m], \mathsf{aux}) \leftarrow \mathcal{D}_\lambda\big] \leq \mathsf{negl}(\lambda).$$
Note that, in this paper, we abuse the notation and write f *to denote indifferently the function* f *or an efficient description of* f*.*

Definition 6 (Sub-Exponentially Unpredictable Distribution). *Let* $\mathcal{D} \coloneqq \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a family of distributions 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 a* sub-exponentially unpredictable distribution *if for all QPT algorithm* $\mathcal{A}$*, we have that*
$$\Pr\big[\mathcal{A}(1^\lambda, f, \mathsf{aux}) = y \;:\; (\mathsf{CC}[f, y, m], \mathsf{aux}) \leftarrow \mathcal{D}_\lambda\big] \leq \frac{1}{\mathsf{subexp}(\lambda)}.$$
Note that, in this paper, we abuse the notation and write f *to denote indifferently the function* f *or an efficient description of* f*.*

Definition 7 (Compute-and-Compare Obfuscator). *A* PPT *algorithm* CC-Obf *is said to be a compute-and-compare obfuscator for a family of unpredictable distributions* $\mathcal{D} \coloneqq \{\mathcal{D}_\lambda\}$ *if:*

- CC-Obf *is functionality preserving: for all* x*,*
$$\Pr\big[\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC})(x) = \mathsf{CC}(x)\big] \geq 1 - \mathsf{negl}(\lambda)$$
- CC-Obf *has distributional indistinguishability: there exists a QPT simulator* Sim *such that*
$$\big\{\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}), \mathsf{aux}\big\} \approx_c \big\{\mathsf{Sim}(1^\lambda, \mathsf{CC.param}), \mathsf{aux}\big\},$$
where $(\mathsf{CC}, \mathsf{aux}) \leftarrow \mathcal{D}_\lambda$*, and* CC.param *denotes the input size, output size, and circuit size of* CC*, that are not required to be obfuscated.*

Theorem 1 ([CLLZ21]). *Assuming post-quantum indistinguishable obfuscation, and the hardness of LWE, there exist compute-and-compare obfuscators for sub-exponentially unpredictable distributions.*

2.6 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 8 (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* $\mathsf{Puncture}$ *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* $x \in \mathcal{X}$*,*

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

- ***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:*

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

  where the probability is taken over the randomness of KeyGen*,* $\mathsf{Puncture}$*, 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 9 (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 10 (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 11 (Monogamy-of-Entanglement Game with Identical Basis (Coset Version)). *This game is between a challenger and a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ - where $\mathcal{B}$ and $\mathcal{C}$ 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 $\mathcal{A}$.*
- *$\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$ and σ_2 to $\mathcal{C}$.*
- *The challenger samples $b \leftarrow \{0,1\}$, then sends (A, b) to both $\mathcal{B}$ and $\mathcal{C}$.*
- *$\mathcal{B}$ returns u_1 and $\mathcal{C}$ returns u_2.*

We say that $\mathcal{B}$ makes a correct guess if $(b = 0 \ \wedge \ u_1 \in A + s)$ or if $(b = 1 \ \wedge \ u_1 \in A^{\perp} + s')$. Similarly, we say that $\mathcal{C}$ makes a correct guess if $(b = 0 \ \wedge \ u_2 \in A + s)$ or if $(b = 1 \ \wedge \ u_2 \in A^{\perp} + s')$. We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win the game if both $\mathcal{B}$ and $\mathcal{C}$ makes a correct guess. For any triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and any security parameter $\lambda \in \mathbb{N}$ for this game, we note $\mathsf{MoE}_{coset}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ the random variable indicating whether $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ 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. $\mathcal{A}$ samples a random bit b^*. $\mathcal{A}$ measures $|A_{s,s'}\rangle$ in the computational basis if $b^* = 0$, or in the Hadamard basis if $b^* = 1$. In both cases, $\mathcal{A}$ sends the outcome u to both $\mathcal{B}$ and $\mathcal{C}$. Regardless of the value of A and b, $\mathcal{B}$ and $\mathcal{C}$ 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 2. *There exists a negligible function $\mathsf{negl}(\cdot)$ such that, for any triple of algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and any security parameter $\lambda \in \mathbb{N}$, $\Pr\left[\mathsf{MoE}_{coset}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C}) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda)$.*

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 2, 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 12 (Monogamy-of-Entanglement Game with Identical Basis (BB84 Version)). *This game is between a challenger and a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ - where $\mathcal{B}$ and $\mathcal{C}$ are non-communicating, and is parametrized by a security parameter λ. An illustration of this game is depicted in Figure 2.*

- *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 $\mathcal{A}$.*
- *$\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$ and σ_2 to $\mathcal{C}$.*
- *The challenger samples $b \leftarrow \{0,1\}$, then sends (θ, b) to both $\mathcal{B}$ and $\mathcal{C}$.*
- *$\mathcal{B}$ returns x_1 and $\mathcal{C}$ returns x_2.*

Let $x_{T_b} := \{x_i \mid \theta_i = b\}$. We say that $(\mathcal{A},\mathcal{B},\mathcal{C})$ win the game if $x_1 = x_2 = x_{T_b}$. For any triple of adversaries $(\mathcal{A},\mathcal{B},\mathcal{C})$ and any security parameter $\lambda \in \mathbb{N}$ for this game, we note $\mathsf{MoE}_{BB84}(1^\lambda, \mathcal{A},\mathcal{B},\mathcal{C})$ the random variable indicating whether $(\mathcal{A},\mathcal{B},\mathcal{C})$ 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.

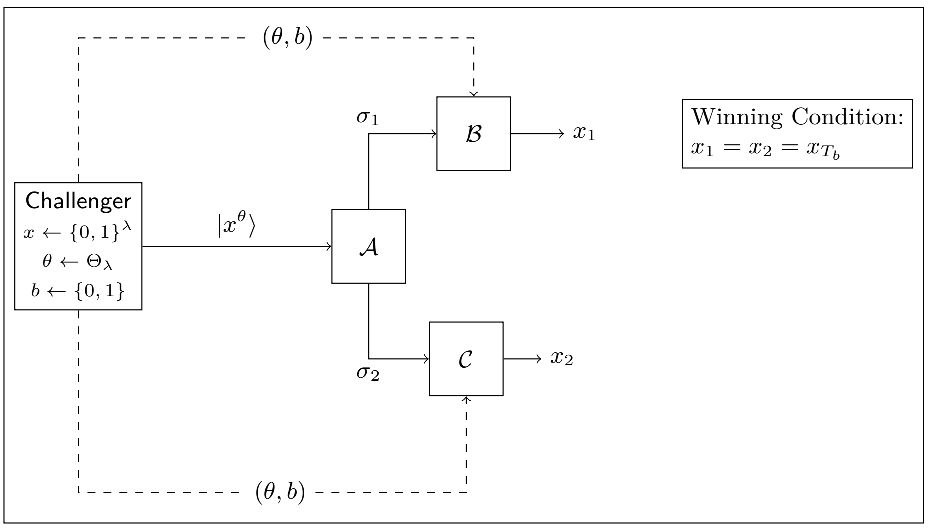


Fig. 2. Monogamy-of-Entanglement Game with Identical Basis (BB84 Version)

Theorem 3. *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that, for any triple of algorithms* $(\mathcal{A},\mathcal{B},\mathcal{C})$ *and any security parameter* $\lambda \in \mathbb{N}$, $\Pr\left[\mathsf{MoE}_{BB84}(1^\lambda, \mathcal{A},\mathcal{B},\mathcal{C}) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda)$.

Proof of Theorem 2 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 3 below.

3.3 Proof of Theorem 3

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 13 (Extended Non-Local Game). *This game is between a challenger and two adversaries $\mathcal{A}$ and $\mathcal{B}$, and is parametrized by a security parameter λ.*

- *$\mathcal{B}$ and $\mathcal{C}$ jointly prepare a quantum state ρ_{012} - where ρ_0 is a λ-qubits quantum state, then send ρ_0 to the challenger. $\mathcal{B}$ and $\mathcal{C}$ keep ρ_1 and ρ_2 respectively. From this step $\mathcal{B}$ and $\mathcal{C}$ 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 $\mathcal{B}$ and $\mathcal{C}$.*
- *$\mathcal{B}$ returns m_1 and $\mathcal{C}$ returns m_2.*

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

Lemma 1. *Let $n \in \mathbb{N}$ and $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ a triple of adversaries for the monogamy-of-entanglement game (Definition 12) 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 13) that win with the same probability p_n.*

Proof. Consider a triple of adversaries for the monogamy-of-entanglement game (Definition 12), 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. $\mathcal{B}$ and $\mathcal{C}$ 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, $\mathcal{B}$ keeps ρ_1 and $\mathcal{C}$ keeps ρ_2. Later, when $\mathcal{B}$ receives (θ, b), from the challenger, $\mathcal{B}$ applies the POVM $B^{\theta,b}$ to ρ_1 and returns the outcome. $\mathcal{C}$ 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.

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

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 $\Pi_1, \dots, \Pi_n$ 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*

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

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

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

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:

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

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 permutation's 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 14 (Computational Monogamy-of-Entanglement Game with Identical Basis (Coset Version)). *This game is between a challenger and a triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *- where* $\mathcal{B}$ *and* $\mathcal{C}$ *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* $\mathcal{A}$*.*
- $\mathcal{A}$ *prepares a bipartite quantum state* σ_{12}*, then sends* σ_1 *to* $\mathcal{B}$ *and* σ_2 *to* $\mathcal{C}$*.*
- *The challenger samples* $b \leftarrow \{0,1\}$*, then sends* (A, b) *to both* $\mathcal{B}$ *and* $\mathcal{C}$*.*
- $\mathcal{B}$ *returns* u_1 *and* $\mathcal{C}$ *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* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *win the game if both* $\mathcal{B}$ *and* $\mathcal{C}$ *makes a correct guess. For any triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *and any security parameter* $\lambda \in \mathbb{N}$ *for this game, we note* $\mathsf{MoE}_{coset(comp)}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ *the random variable indicating whether* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *win the game or not.*

Theorem 4. *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that, for any triple of QPT algorithms* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *and any security parameter* $\lambda \in \mathbb{N}$*,* $\Pr\left[\mathsf{MoE}_{coset(comp)}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C}) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda)$*.*

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 supposed 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 15 (κ-Parallel Computational Monogamy-of-Entanglement Game with Identical Basis (Coset Version)). *This game is between a challenger and a triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *- where* $\mathcal{B}$ *and* $\mathcal{C}$ *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* $\mathcal{A}$*.*
- $\mathcal{A}$ *prepares a bipartite quantum state* σ_{12}*, then sends* σ_1 *to* $\mathcal{B}$ *and* σ_2 *to* $\mathcal{C}$*.*
- *The challenger samples* $r \leftarrow \{0,1\}^\kappa$*, then sends* $\{A_i\}_{i \in [\![1,\kappa]\!]}$ *and* r *to both* $\mathcal{B}$ *and* $\mathcal{C}$*.*
- $\mathcal{B}$ *returns* κ *vectors* $\{u_i\}_{i \in [\![1,\kappa]\!]}$ *and* $\mathcal{C}$ *returns* κ *vectors* $\{u_i'\}_{i \in [\![1,\kappa]\!]}$*.*

We say that $\mathcal{B}$ *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* $\mathcal{C}$ *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* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *win the game if both* $\mathcal{B}$ *and* $\mathcal{C}$ *makes a correct guess. For any triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *and any security parameter* $\lambda \in \mathbb{N}$ *for this game, we note* $\kappa - \mathsf{MoE}_{coset(comp)}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ *the random variable indicating whether* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *win the game or not.*

Theorem 5. *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that, for any triple of QPT algorithms* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *and any security parameter* $\lambda \in \mathbb{N}$*,* $\Pr\left[\kappa - \mathsf{MoE}_{coset(comp)}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C}) = 1\right] \leq \mathsf{negl}(\lambda)$*.*

Comparison with [CGLZR23]. In [CGLZR23], the authors also present a new monogamy-of-entanglement game for coset states. Their game is similar too our parallel version except that, instead of receiving the same challenge bitstring r, $\mathcal{B}$ and $\mathcal{C}$ receive respectively r_1 and r_2, two independently sampled challenge bitstrings, and must answer accordingly. Note that the hardness of the parallel version of our game can be proven using lemma 18 of [AKL23] on their game[2]. We still provide a direct proof for this theorem in Section 3.6 for completeness. We emphasize that for the single-instance version, however, the same lemma cannot be applied.

3.6 Proof of Parallel Version of the Monogamy Game

In this subsection, we prove Theorem 5. 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 3, 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 $\mathcal{A}$ and $\mathcal{B}$, and is parametrized by a security parameter λ and a number of repetitions $\kappa := \mathsf{poly}(\lambda)$.

- $\mathcal{B}$ and $\mathcal{C}$ jointly prepare a quantum state ρ_{012} - where ρ_0 is composed of κ λ-qubits registers, denoted as $\rho_0^1, \ldots, \rho_0^\kappa$ - then send ρ_0 to the challenger. $\mathcal{B}$ and $\mathcal{C}$ keep ρ_1 and ρ_2 respectively. From this step $\mathcal{B}$ and $\mathcal{C}$ 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 $\theta := (\theta^1, \ldots, \theta^\kappa)$ and r to $\mathcal{B}$ and $\mathcal{C}$.
- $\mathcal{B}$ returns $\{m_1^j\}_{j \in [\![1,\kappa]\!]}$ and $\mathcal{C}$ 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 $(\mathcal{B},\mathcal{C})$ 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 3 that

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

where $\{\pi_{k,\alpha}\}$ is a family of mutually orthogonal permutations indexed by $k = k_1\| \ldots \|k_\kappa$ - 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 $S^j := \{i \in R \ : \ \theta_i^j = r_j$ and $\theta_i'^j = 1 - r_j\}$. We define $\bar{\mathrm{P}}$ and $\bar{\mathrm{Q}}$ as

[2] We thank Alper Çakan and Vipul Goyal for pointing out this shorter proof.

follows:

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

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

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 3 (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 3. 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 Conjectures on Simultaneous Compute-and-Compare Obfuscation

In this section, we present our conjectures. We first give an overview of the conjectures, then we define them formally, and finally we discuss their relation to similar conjectures in a recent work [AB23].

4.1 Overview

Recall that for (sub-exponentially) unpredictable distribution $\mathcal{D}$, there exists a compute-and-compare obfuscator (Section 2.5). We are interested in whether this result still holds in a non-local context. More precisely, consider the two following tasks, which we call *simultaneous distinguishing* and *simultaneous predicting*. Simultaneous predicting asks two players, Bob and Charlie, given a function associated to a compute-and-compare program, and a quantum state as auxiliary information on the program, to output the associated lock value. Note that the function given to Bob and the one given to Charlie are not necessarily the same, and that the same goes for the quantum states they are given. Also, crucially, Bob's and Charlie's quantum states can be entangled. In simultaneous distinguishing, Bob and Charlie are given either an obfuscated compute-and-compare program, or the outcome of a simulator on this program's parameters[3]. As in simultaneous predicting, they are also given a quantum state each, but here, they are asked to tell whether they received the obfuscated program, or the simulated one.

These two tasks are parameterized by a distribution over triple of the form $(\mathsf{CC}_1, \mathsf{CC}_2, \sigma_{12})$ - where the two first elements are compute-and-compare programs used to create the challenges in the challenge phase and the last one is the bipartite quantum state shared by Bob and Charlie We say that such a distribution is simultaneously unpredictable if no adversaries can succeed in the associated simultaneous predicting task; and that simultaneous compute-and-compare obfuscation exists for this distribution if there is a compute-and-compare obfuscator with respect to which no adversaries can succeed in the associated simultaneous distinguishing task. The question we ask now is:

Question. *Is there simultaneous compute-and-compare obfuscation for any simultaneous unpredictable distribution ?*

As discussed in [CLLZ21], this question is far from trivial. Indeed, consider its contraposition: *if all candidate algorithms for simultaneous compute-and-compare obfuscation fail to obfuscate the programs as desired, does it mean that the distribution is simultaneously predictable for a certain pair of algorithms ?* Intuitively, the difficulty here stems from whether the challenges are independent or not: if they are, then one can analyze the two adversaries in the distinguishing game independently, and thus say that if the first adversary succeeds in their part of the task, then they can predict their lock value, and that same goes for the second adversary. If the challenges are not independent in the other hand, it is not clear what happens when the first adversary predicts the lock value: as, concretely, the prediction is a measurement, perhaps this measurement perturbs the other register in a way that prevents the other adversary to predict their lock value.

In this work, we break down this question in the following way: we parameterize the distinguishing task by a distribution over pairs of coins used as random tape by the compute-and-compare obfuscator, and by a distribution over bits used to determine whether Bob and Charlie receive the obfuscated program or the simulated one. We consider two types of distributions for the coins' distribution and the bits' distribution:

- the uniform distribution, where the pairs (r_1, r_2) (resp. (b_1, b_2)) are such that r_1 and r_2 (resp. b_1 and b_2) are uniformly and independently sampled;
- the identical distribution where the pairs (r_1, r_2) (resp. (b_1, b_2)) are such that r_1 (resp. b_1) is uniformly sampled and $r_2 = r_1$ (resp. $b_2 = b_1$). We then simply write these pairs as (r, r) and (b, b).

In [CLLZ21], the authors show that the answer of the question above is yes when both the coins' and the bits' distributions are uniform. In particular, they use a technique called "threshold projective implementation" to show that, with these parameters, one can analyze the two adversaries independently, hence Bob prediction does not perturb Charlie's one. We conjecture that this is still the case when the coins' distribution is identical and the bits' one is either uniform or also identical.

[3] By parameters, we mean input size, output size, and depth of a given circuit.

Relation with [AB23]. In a recent work of Ananth and Behera [AB23], the authors make a similar conjecture, this time on simultaneous Goldreich-Levin prediction. Roughly, the usual Goldreich-Levin theorem states that if F is a one way function (meaning that a random x is not predictable given $F(x)$), then no adversary can distinguish the dot product $x \cdot r$ from a random bit, given $F(x)$ - where x is a random input and r a random bitstring of same length. The authors of [AB23] consider the simultaneous version of this task, that is, assuming that (x_1, x_2) are simultaneously unpredictable given $(F(x_1), F(x_2))$ (in the same sense as our definitons above), then $x_1 \cdot r_1$ and $x_2 \cdot r_2$ are simultaneously indistinguishable from two random strings - where the pairs (x_1, x_2), (r_1, r_2), and (b_1, b_2), are sampled from different types of distributions (uniform or identical), similarly as in our case - and they finally describe two conjectures. Note that, as there is a construction of compute-and-compare obfuscation [CLLZ21] (based on iO and hardness of LWE assumptions) that ultimately relies on the Goldreich-Levin theorem, then we expect that the conjectures of [AB23], combined with iO and LWE assumptions, imply our conjectures.

4.2 Definitions

We present formally the notions of simultaneous distinguishing and simultaneous predicting games. For ease of reading, we first introduce what we call simultaneous compute-and-compare distributions.

Definition 16 (Simultaneous Compute-and-Compare Distribution). *We call* simultaneous compute-and-compare distribution *a family of distributions $\mathcal{D}_{\mathsf{CC}} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ over triple of the form $(\mathsf{CC}_1[f_1, y_1, m_1], \mathsf{CC}_2[f_2, y_2, m_2], \sigma_{12})$ where $\mathsf{CC}_1[f_1, y_1, m_1]$ and $\mathsf{CC}_2[f_2, y_2, m_2]$ are both compute-and-compare programs, and σ_{12} is a bipartite quantum state representing some auxiliary information. In the following, we denote the first and second registers of σ_{12} as σ_1 and σ_2.*

We are now ready to define simultaneous distinguishing and predicting games.

Definition 17 (Simultaneous Distinguishing Game). *We define below a* simultaneous distinguishing game*, parameterized by a pair of efficient algorithms $(\mathsf{CC\text{-}Obf}, \mathsf{Sim})$ - where $\mathsf{CC\text{-}Obf}$ uses a bitstring $r \in \{0,1\}^\ell$ as random coins for some $\ell \in \mathbb{N}$, a simultaneous compute-and-compare distribution $\mathcal{D}_{\mathsf{CC}}$, a "coins' distribution" $\mathcal{D}_R$ over $\{0,1\}^{2\ell}$, a "bits' distribution" $\mathcal{D}_B$ over $\{0,1\}^2$, and a security parameter λ. This game is between a challenger and a pair of adversaries $\mathcal{B}$ and $\mathcal{C}$.*

- *The challenger samples $(\mathsf{CC}_1, \mathsf{CC}_2, \sigma_{12}) \leftarrow \mathcal{D}_{\mathsf{CC}}$, $(b_1, b_2) \leftarrow \mathcal{D}_B$ and $(r_1, r_2) \leftarrow \mathcal{D}_R$.*
- *The challenger sends σ_1 to $\mathcal{B}$. The challenger also sends $\mathsf{CC\text{-}Obf}(\mathsf{CC}_1; r_1)$ if $b_1 = 0$, or $\mathsf{Sim}(1^\lambda, \mathsf{CC}_1.\mathsf{param})$ if $b_1 = 1$ to $\mathcal{B}$.*
- *Similarly, the challenger sends σ_2 to $\mathcal{C}$. Then they also send $\mathsf{CC\text{-}Obf}(\mathsf{CC}_2; r_2)$ if $b_2 = 0$, or $\mathsf{Sim}(1^\lambda, \mathsf{CC}_2.\mathsf{param})$ if $b_2 = 2$ to $\mathcal{C}$.*

$\mathcal{B}$, and $\mathcal{C}$ win the game if $\mathcal{B}$ returns $b_1' = b_1$ and $\mathcal{C}$ returns $b_2' = b_2$.

We denote the random variable that indicates whether a pair of adversaries $(\mathcal{B}, \mathcal{C})$ wins the game or not as $\mathsf{Simul} - \mathsf{Dist}^{(\mathsf{CC\text{-}Obf}, \mathsf{Sim})}_{\mathcal{D}_{\mathsf{CC}}, \mathcal{D}_R, \mathcal{D}_B}(1^\lambda, \mathcal{B}, \mathcal{C})$.

Definition 18 (Simultaneous Predicting Game). *We define below a* simultaneous predicting game*, parametrized by a simultaneous compute-and-compare distribution $\mathcal{D}_{\mathsf{CC}}$, and a security parameter λ. This game is between a challenger and a pair of adversaries $\mathcal{B}$ and $\mathcal{C}$.*

- *The challenger samples $(\mathsf{CC}_1[f_1, y_1, m_1], \mathsf{CC}_2[f_2, y_2, m_2], \sigma_{12}) \leftarrow \mathcal{D}_{\mathsf{CC}}$.*
- *Then, the challenger sends (f_1, σ_1) to $\mathcal{B}$ and (f_2, σ_2) to $\mathcal{C}$.*

$\mathcal{B}$, and $\mathcal{C}$ win the game if $\mathcal{B}$ returns $y_1' = y_1$ and $\mathcal{C}$ returns $y_2' = y_2$.

We denote the random variable that indicates whether a pair of adversaries $(\mathcal{B}, \mathcal{C})$ wins the game or not as $\mathsf{Simul} - \mathsf{Predict}_{\mathcal{D}_{\mathsf{CC}}}(1^\lambda, \mathcal{B}, \mathcal{C})$.

4.3 Conjectures

We now state our two conjectures. An informal description of the conjectures is illustrated in Figure 3.

Conjecture 1. Let $\mathcal{D}_{\mathsf{CC}}$ a simultaneous compute-and-compare distribution, $\mathcal{D}_R$ the identical distribution over $\{0,1\}^{2\ell}$, and $\mathcal{D}_B$ the uniform distribution over $\{0,1\}^2$. Assume that, for all pair of QPT adversaries $(\mathcal{B},\mathcal{C})$,

$$\Pr\left[\mathsf{Simul}-\mathsf{Predict}_{\mathcal{D}_{\mathsf{CC}}}(1^\lambda,\mathcal{B},\mathcal{C})=1\right]\leq\mathsf{negl}(\lambda)$$

Then, there exists a compute-and-compare obfuscator $\mathsf{CC\text{-}Obf}$ and its associated simulator Sim such that, for all pair of QPT adversaries $(\mathcal{B},\mathcal{C})$,

$$\Pr\left[\mathsf{Simul}-\mathsf{Dist}^{(\mathsf{CC\text{-}Obf},\mathsf{Sim})}_{\mathcal{D}_{\mathsf{CC}},\mathcal{D}_R,\mathcal{D}_B}(1^\lambda,\mathcal{B},\mathcal{C})=1\right]\leq\frac{1}{2}+\mathsf{negl}(\lambda)$$

Conjecture 2. Let $\mathcal{D}_{\mathsf{CC}}$ a simultaneous compute-and-compare distribution, $\mathcal{D}_R$ the identical distribution over $\{0,1\}^{2\ell}$, and $\mathcal{D}_B$ the identical distribution over $\{0,1\}^2$. Assume that, for all pair of QPT adversaries $(\mathcal{B},\mathcal{C})$,

$$\Pr\left[\mathsf{Simul}-\mathsf{Predict}_{\mathcal{D}_{\mathsf{CC}}}(1^\lambda,\mathcal{B},\mathcal{C})=1\right]\leq\mathsf{negl}(\lambda)$$

Then, there exists a compute-and-compare obfuscator $\mathsf{CC\text{-}Obf}$ and its associated simulator Sim such that, for all pair of QPT adversaries $(\mathcal{B},\mathcal{C})$,

$$\Pr\left[\mathsf{Simul}-\mathsf{Dist}^{(\mathsf{CC\text{-}Obf},\mathsf{Sim})}_{\mathcal{D}_{\mathsf{CC}},\mathcal{D}_R,\mathcal{D}_B}(1^\lambda,\mathcal{B},\mathcal{C})=1\right]\leq\frac{1}{2}+\mathsf{negl}(\lambda)$$

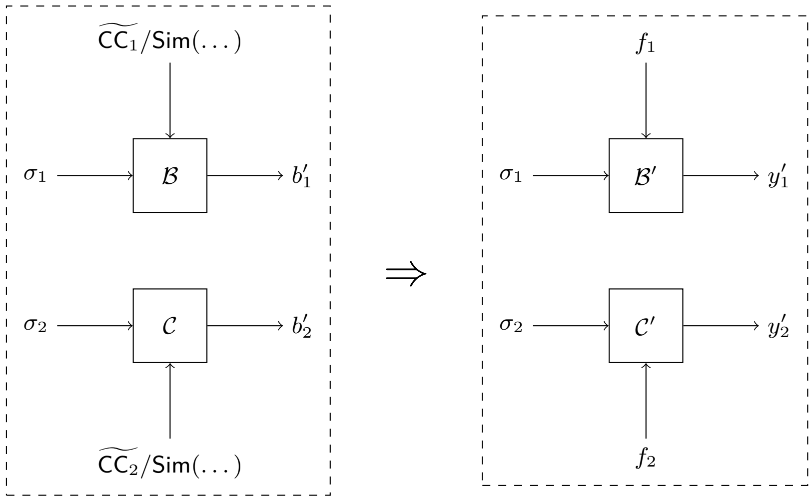
As we in fact use the contrapositions of these conjectures in the following of the paper, we present these contrapositions as the following corollaries.

Corollary 2. *Let $\mathcal{D}_{\mathsf{CC}}$ a simultaneous compute-and-compare distribution, $\mathcal{D}_R$ the identical distribution over $\{0,1\}^{2\ell}$, and $\mathcal{D}_B$ the uniform distribution over $\{0,1\}^2$. Assume that, for all efficient and functionality preserving algorithm $\mathsf{CC\text{-}Obf}$, and for all efficient simulator Sim, there exists a pair of QPT adversaries $(\mathcal{B},\mathcal{C})$ winning the distinguishing game - parametrized by $\mathcal{D}_{\mathsf{CC}},\mathcal{D}_R$, and $\mathcal{D}_B$ - with non-negligible advantage over $1/2$. Then there exists a pair of QPT adversaries $(\mathcal{B},\mathcal{C})$ winning the associated predicting game with non-negligible probability.*

Corollary 3. *Let $\mathcal{D}_{\mathsf{CC}}$ a simultaneous compute-and-compare distribution, $\mathcal{D}_R$ the identical distribution over $\{0,1\}^{2\ell}$, and $\mathcal{D}_B$ the identical distribution over $\{0,1\}^2$. Assume that, for all efficient and functionality preserving algorithm $\mathsf{CC\text{-}Obf}$, and for all efficient simulator Sim, there exists a pair of QPT adversaries $(\mathcal{B},\mathcal{C})$ winning the distinguishing game - parametrized by $\mathcal{D}_{\mathsf{CC}},\mathcal{D}_R$, and $\mathcal{D}_B$ - with non-negligible advantage over $1/2$. Then there exists a pair of QPT adversaries $(\mathcal{B},\mathcal{C})$ winning the associated predicting game with non-negligible probability.*

5 Single-Decryptor and Copy-Protection of Pseudorandom Functions

In this section, we recall the notions of single-decryptor [GZ20] and copy-protection of pseudorandom functions [CLLZ21]. 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.



Distinguishing Game **Predicting Game**

Fig. 3. Contraposition of the conjectures: if $\mathcal{B}$ and $\mathcal{C}$ win the distinguishing game on the left with significant advantage over $1/2$, then there exist $\mathcal{B}'$ and $\mathcal{C}'$ winning the predicting game on the right with non-negligible probability. $\widetilde{\mathsf{CC}}_1$ and $\widetilde{\mathsf{CC}}_2$ represent the compute-and-compare obfuscation of CC_1 and CC_2 with the same random coins.

5.1 Definition of a Single-Decryptor

Definition 19 (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* $\mathsf{QKeyGen}$ *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 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 an 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 notion is defined through a piracy game, in which a first adversary Alice is given a quantum decryption key and must split it and share it between two other adversaries, Bob and Charlie. Bob and Charlie then receive an encryption of either the first, or the second message of a pair (m_0, m_1) - known by the three adversaries - as a challenge, and must guess the encryption of which message they were given. The game (and hence the anti-piracy security) is defined with respect to two distributions: $\mathcal{D}_B$ that yields two bits deciding 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 of each challenge. In order to prove the security of our unclonable encryption and copy-protection schemes, we need to consider the security when $\mathcal{D}_R$ is the identical distribution[4]and $\mathcal{D}_B$ is either the uniform or the identical distribution. We denote the case where $\mathcal{D}_B$ is the uniform distribution as anti-piracy with respect to *product distribution*, and the case where it is the identical distribution as anti-piracy with respect to *identical distribution*.

Remark 2. Note that the original anti-piracy security proposed in [CLLZ21] is simply our definition where $\mathcal{D}_B$ and $\mathcal{D}_R$ are both uniform distributions.

Definition 20 (Piracy Game for Single-Decryptor). *We define below a* piracy game for single-decryptor*, parametrized by a single-decryptor scheme $\mathcal{E} = \langle \mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$, and a security parameter λ. This game is between a challenger and a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. As the two variant of the games (with respect to product or identical distribution) differ only in the challenge phase, we describe below a different challenge phase for each variant.*

- ***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:*** *$\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$, σ_2 to $\mathcal{C}$, and two pairs of messages (m_0^1, m_1^1) and (m_0^2, m_1^2) to the challenger.*
- ***Challenge phase (product distribution):***
  - *The challenger samples* $(b_1, b_2) \leftarrow\$ \{0,1\}$, *and* $(r_1, r_2) \leftarrow\$ \{0,1\}^{\mathsf{poly}(\lambda)}$.
  - *The challenger sends* $\mathsf{Enc}(\mathsf{pk}, m_{b_1}^1; r_1)$ *to* $\mathcal{B}$, *and* $\mathsf{Enc}(\mathsf{pk}, m_{b_2}^2; r_2)$ *to* $\mathcal{C}$.
- ***Challenge phase (identical distribution):***
  - *The challenger samples* $b \leftarrow\$ \{0,1\}$, *and* $r \leftarrow\$ \{0,1\}^{\mathsf{poly}(\lambda)}$.
  - *The challenger sends* $\mathsf{Enc}(\mathsf{pk}, m_b^1; r)$ *to* $\mathcal{B}$, *and* $\mathsf{Enc}(\mathsf{pk}, m_b^2; r)$ *to* $\mathcal{C}$.

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game in the product distribution if $\mathcal{B}$ returns $b'_1 = b_1$ and $\mathcal{C}$ returns $b'_2 = b_2$; and in the identical distribution if both $\mathcal{B}$ and $\mathcal{C}$ return b.

We denote the random variable that indicates whether a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win the game or not as $\mathsf{SD} - \mathsf{AP}^{\mathcal{E}}_{PD}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ or $\mathsf{SD} - \mathsf{AP}^{\mathcal{E}}_{ID}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ depending on which variant of the game we consider (PD and ID respectively denote the product and identical distributions).

Definition 21 (Indistinguishable Anti-Piracy Security). *A single-decryptor scheme $\mathcal{E}$ has* indistinguishable anti-piracy security *with respect to the product distribution if no QPT adversary can win the piracy game above (with the product distribution challenge phase) with probability significantly greater than 1/2. More precisely, for any triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:*

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

Similarly, a single-decryptor scheme $\mathcal{E}$ has indistinguishable anti-piracy security *with respect to the identical distribution if no QPT adversary can win the piracy game above (with the identical distribution challenge phase) with probability significantly greater than 1/2. More precisely, for any triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:*

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

[4] Recall (Section 2.2) that a distribution is identical if it yields a pair of identical elements (x, x) (where x is sampled uniformly at random).

5.2 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 $\left(\mathsf{sk} \coloneqq \{A_i, s_i, s'_i\}_{i\in[\![1,\kappa]\!]}, \mathsf{pk} \coloneqq \{\widehat{\mathsf{P}}_{A_i+si}, \widehat{\mathsf{P}}_{A_i^\perp+s'i}\}_{i\in[\![1,\kappa]\!]}\right)$.
• $\rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$:
– Parse sk as $\{A_i, s_i, s'_i\}_{i\in[\![1,\kappa]\!]}$;
– Return $\bigotimes_{i=1}^{\kappa} \lvert A_{i,s_i,s'_i}\rangle$.
• $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$:
– 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 $\mathsf{iO}(\mathsf{Q_{m,r}})$ of program $\mathsf{Q_{m,r}}$ described in Section 5.2.
– Return $c \coloneqq (r, \mathsf{iO}(\mathsf{Q_{m,r}}))$.
• $m/\perp \leftarrow \mathsf{Dec}(\rho_{\mathsf{sk}}, c)$:
– Parse ρ_{sk} as $\bigotimes_{i=1}^{\kappa} \lvert A_{i,s_i,s'_i}\rangle$ and $c \leftarrow (r, \mathsf{iO}(\mathsf{Q_{m,r}}))$;
– For all $i \in [\![1,\kappa]\!]$: if $r_i = 1$, apply $\mathsf{H}^{\otimes n}$ to $\lvert A_{i,s_i,s'_i}\rangle$;
– Let ρ' be the resulting state, run $\mathsf{iO}(\mathsf{Q_{m,r}})$ 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. 4. Program $\mathsf{Q_{m,r}}$.

Remark 3. Note that the underlying iO algorithm used in the encryption algorithm of Construction 1 might use a random tape. 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 coins r_{iO} and r respectively used for the iO algorithm and for the program $\mathsf{Q_{m,r}}$.

Theorem 6. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions, and Conjecture 1, Construction 1 has* indistinguishable anti-piracy security with respect to the product distribution.

Theorem 7. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions, and Conjecture 2, Construction 1 has* indistinguishable anti-piracy security with respect to the identical distribution*.*

5.3 Proof of Theorem 6

In this section, we prove Theorem 6. 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 triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, the probability that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ 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, with respect to the product distribution.

- **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:** $\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$, σ_2 to $\mathcal{C}$, and two pairs of messages (m_0^1, m_1^1) and (m_0^2, m_1^2) to the challenger.
- **Challenge phase:**
  - The challenger samples random coins $r_{\mathsf{iO}} \leftarrow\$ \{0,1\}^{\mathsf{poly}(\lambda)}$ - to be used in the iO algorithm, and $r \leftarrow\$ \{0,1\}^{\mathsf{poly}(\lambda)}$ - to be used in the encryption algorithm, and two bits uniformly at random $b_1, b_2 \leftarrow\$ \{0,1\}$.
  - The challenger computes $c_1 := (r, \mathsf{Q}_1) \leftarrow \mathsf{Enc}(\mathsf{pk}, m^1_{b_1}; (r, r_{\mathsf{iO}}))$, and $c_2 := (r, \mathsf{Q}_2) \leftarrow \mathsf{Enc}(\mathsf{pk}, m^2_{b_2}; (r, r_{\mathsf{iO}}))$ (note that the programs Q_1 and Q_2 have been obfuscated using r_{iO} as the randomness).
  - The challenger sends c_1 to $\mathcal{B}$, and c_2 to $\mathcal{C}$.

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game if $\mathcal{B}$ returns $b'_1 = b_1$ and $\mathcal{C}$ returns $b'_2 = b_2$.

Game G_1: In this second hybrid, we replace the obfuscated programs Q_1 and Q_2 by obfuscated compute-and-compare programs. More formally, for $i \in [\![1,\kappa]\!]$, we define[5]

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

We similarly define $\mathsf{Can}_r(u_1, ..., u_\kappa) = (\mathsf{Can}_{1,r_1}(u_1), \dots, \mathsf{Can}_{\kappa,r_\kappa}(u_\kappa))$ and $c_r = (c_{1,r_1}, \dots, c_{\kappa,r_\kappa})$ for any $r \in \{0,1\}^\kappa$. Finally, we write CC_1 and CC_2 to denote $\mathsf{CC}[\mathsf{Can}_r, c_r, m^1_{b_1}]$ and $\mathsf{CC}[\mathsf{Can}_r, c_r, m^2_{b_2}]$.

Then, we replace Q_1 by $\mathsf{iO}(\mathsf{CC}_1)$ and Q_2 by $\mathsf{iO}(\mathsf{CC}_2)$. Because the programs Q_1 and Q_2 are respectively functionally equivalent to CC_1 and CC_2, then from iO security, G_0 and G_1 are negligibly close.

Game G_2: In this last hybrid, we replace $\mathsf{iO}(\mathsf{CC}_1)$ by $\mathsf{iO}(\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}_1))$ and $\mathsf{iO}(\mathsf{CC}_2)$ by $\mathsf{iO}(\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}_2))$. Because the programs CC_1 and CC_2 are respectively functionally equivalent to $\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}_1)$ and $\mathsf{CC\text{-}Obf}(1^\lambda, \mathsf{CC}_2)$, then from iO security, G_1 and G_2 are negligibly close.

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

Lemma 5. *Define the simultaneous compute-and-compare distribution* $\mathcal{D}^{\mathcal{A}}_{\mathsf{CC}}$*, parametrized with a QPT algorithm* $\mathcal{A}$ *for the hybrid* G_2 *as follows:*

[5] Recall that for a subspace A and a vector u, $\mathsf{Can}_A(u)$ - defined in Section 2.3 - is the coset representative of $A + u$. Recall also that Can_A can be efficiently implemented given a description of A.

- *sample κ cosets descriptions $(A_i, s_i, s'_i)_{i\in[\![1,\kappa]\!]}$;*
- *run $\mathcal{A}$ on $\otimes_{i=1}^{\kappa} |A_i, s_i, s'_i\rangle$ to get σ_{12} and $(m_0^1, m_1^1), (m_0^2, m_1^2)$;*
- *sample $r \leftarrow\$ [\![1,\kappa]\!]$ and $b_1, b_2 \leftarrow\$ \{0,1\}$;*
- *define the bipartite quantum state σ'_{12} with $\sigma_1 \otimes |b_1\rangle\langle b_1|$ as first register and $\sigma_2 \otimes |b_2\rangle\langle b_2|$ as second register;*
- *return $\left(\mathsf{CC}[\mathsf{Can}_r, c_r, m_{b_1}^1], \mathsf{CC}[\mathsf{Can}_r, c_r, m_{b_2}^2], \sigma'_{12}\right)$.*

Assume in addition that a triple of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win hybrid G_2 with non-negligible advantage over $1/2$. Then there exists a pair of QPT algorithms $(\mathcal{B}', \mathcal{C}')$ that win the simultaneous predicting game, parametrized with $\mathcal{D}_{\mathsf{CC}}^{\mathcal{A}}$, with non-negligible probability.

Proof. The proof follows from the contraposition of Conjecture 1. We construct a pair of QPT adversaries $(\mathcal{B}', \mathcal{C}')$ for the simultaneous distinguishing game parametrized with any efficient and functionality preserving $\mathsf{CC\text{-}Obf}$, any efficient simulator Sim, the simultaneous compute-and-compare distribution $\mathcal{D}_{\mathsf{CC}}^{\mathcal{A}}$, the identical coins' distribution $\mathcal{D}_R$, and the uniform bits' distribution $\mathcal{D}_B$.

- $\mathcal{B}'$ receives the program C_1 from the challenger: C_1 is either a compute-and-compare obfuscation $\mathsf{CC\text{-}Obf}(\mathsf{CC}_1; r)$ - where $\mathsf{CC}_1 := \mathsf{CC}[\mathsf{Can}_r, c_r, m_{b_1}^1]$ - or a simulated program $\mathsf{Sim}(\mathsf{CC}_1.\mathsf{param})$. $\mathcal{B}'$ also receives $\sigma_1 \otimes |b_1\rangle\langle b_1|$.
- $\mathcal{B}'$ runs $\mathcal{B}$ on (σ_1, C_1) to get the outcome b'_1.
- If $b'_1 = b_1$, $\mathcal{B}'$ returns 0, otherwise $\mathcal{B}'$ return 1.

$\mathcal{C}'$ is defined similarly by replacing the “1” indices by “2” indices.

Because, $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win G_2 with non-negligible advantage over $1/2$, and, when C_1 (resp. C_2) is the obfuscated program, the challenge given to $\mathcal{B}$ (resp. $\mathcal{C}$) comes from the same distribution as in G_2, then $\mathcal{B}$ (resp. $\mathcal{C}$) guesses b_1 (resp. b_2) correctly with non-negligible advantage over $1/2$. On the other hand, when C_1 (resp. C_2) is simulated, then it does not hold any information on b_1 (resp. b_2), hence $\mathcal{B}$ (resp. $\mathcal{C}$) guesses correctly only with probability $1/2$. Thus, $\mathcal{B}'$ and $\mathcal{C}'$ succeed in simultaneously distinguishing the simulated programs from the obfuscated ones with non-negligible advantage over $1/2$. Because this reasoning holds for all efficient functionality preserving $\mathsf{CC\text{-}Obf}$ and simulator Sim, then there is no compute-and-compare obfuscator for the distribution $\mathcal{D}_{\mathsf{CC}}$. Using the contraposition of Conjecture 1, completes the proof. ☐

Reduction to monogamy-of-entanglement. We are now ready to proceed to the reduction. Assume that there exists a triple of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ that win the last hybrid G_2 with non-negligible advantage over $1/2$. Then, by Lemma 5, there exists two QPT algorithms $\mathcal{B}'$ and $\mathcal{C}'$ that win the simultaneous predicting game defined above. We construct a triple of QPT algorithms $(\mathcal{A}'', \mathcal{B}'', \mathcal{C}'')$ for the κ-parallel computational version of monogamy-of-entanglement game (Definition 15).

- $\mathcal{A}''$, on input the coset states $\{|A_i, s_i, s'_i\rangle\}_{i\in[\![1,\kappa]\!]}$ and the obfuscated membership programs $\{\mathsf{P}_{A_i+s_i}, \mathsf{P}_{A_i^\perp+s'_i}\}_{i\in[\![1,\kappa]\!]}$:
  - runs $\mathcal{A}$ on these coset states and programs to get σ_{12} and $(m_0^1, m_1^1), (m_0^2, m_1^2)$;
  - sample $b_1, b_2 \leftarrow\$ \{0,1\}$;
  - then prepares the bipartite quantum state σ'_{12} with $\sigma_1 \otimes |b_1\rangle\langle b_1|$ as first register and $\sigma_2 \otimes |b_2\rangle\langle b_2|$ as second register;
  - and finally sends σ'_1 to $\mathcal{B}''$ and σ'_2 to $\mathcal{C}''$.
- $\mathcal{B}''$, on input σ'_1, the subspace descriptions $\{A_i\}_{i\in[\![1,\kappa]\!]}$ and the random basis r:
  - construct a description of Can_r (note that such a description can be computed efficiently given $\{A_i\}_{i\in[\![1,\kappa]\!]}$ and r);
  - runs $\mathcal{B}'$ on $(\sigma_1, \mathsf{Can}_r)$ to get the outcome y'_1;
  - and finally returns y'_1.

- $\mathcal{C}''$ is defined similarly as $\mathcal{B}''$ by replacing the "1" indices by "2" indices.

From Lemma 5, we know that, with non-negligible probability, both y_1' and y_2' are the lock values of the compute-and-compare programs. Then $\mathcal{A}'', \mathcal{B}'', \mathcal{C}''$ win the game with non-negligible probability, contradicting Theorem 5 and concluding the proof.

5.4 Proof of Theorem 7

The proof of Theorem 7 is almost the same as the one of Theorem 6: we proceed with the same sequence of hybrids and use Conjecture 2 instead of Conjecture 1 to finish the proof.

5.5 Copy-Protection of Pseudorandom Functions

In this subsection, we formally define copy-protection of pseudorandom function [CLLZ21] and its correctness and anti-piracy notions.

Definition 22 (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 similarly 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. More precisely, the game is played by a triple of adversaries (Alice, Bob and Charlie): Alice splits the quantum state and, in order to test both shares, they are sent to Bob and Charlie as well as the challenge (image's input of fake input) who are asked to guess which type of input they received. We define two different variants for this security: security with respect to the *product distribution* and security with respect to the *identical distribution*. When considering security with respect to the product distribution, for each share, a challenge is sampled independently to be either the image's input, or a freshly sampled fake input (both with probability 1/2). When considering security with respect to the identical distribution on the other hand, the challenge is still either image's input or a fake input, but it is the same for both Bob and Charlie.

Definition 23 (Piracy Game for Pseudorandom Function Copy-Protection). *We define below a* piracy game for pseudorandom function copy-protection*, parametrized by a pseudorandom function copy-protection scheme* $\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ *and a security parameter* λ*. As the two variants of the game (with respect to the product distribution or to the identical distribution) differ only in the challenge phase, we define a different challenge phase for each variant. This game is between a challenger and a triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*.*

- ***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:*** $\mathcal{A}$ *prepares a bipartite quantum state* σ_{12}*, then sends* σ_1 *to* $\mathcal{B}$ *and* σ_2 *to* $\mathcal{C}$.
- ***Challenge phase (product distribution):***
  - *The challenger samples two bits* $b_1, b_2 \leftarrow\!\!\$\ \{0,1\}$*, and two inputs* $x_1, x_2 \leftarrow\!\!\$\ \mathcal{X}$.
  - *If* $b_1 = 0$*, the challenger sends* x *to* $\mathcal{B}$*; otherwise, the challenger sends* x_1.
  - *Similarly, if* $b_2 = 0$*, the challenger sends* x *to* $\mathcal{C}$*; otherwise, the challenger sends* x_2.
- ***Challenge phase (identical distribution):***
  - *The challenger samples a bit* $b \leftarrow\!\!\$\ \{0,1\}$*, and an input* $x_0 \leftarrow\!\!\$\ \mathcal{X}$.
  - *If* $b = 0$*, the challenger sends* x *to both* $\mathcal{B}$ *and* $\mathcal{C}$*; otherwise, the challenger send them* x_0.

$\mathcal{A}$, $\mathcal{B}$, *and* $\mathcal{C}$ *win the game with respect to the product distribution if* $\mathcal{B}$ *returns* b_1 *and* $\mathcal{C}$ *returns* b_2*; and with respect to the identical distribution if both* $\mathcal{B}$ *and* $\mathcal{C}$ *return* b.

We denote the random variable that indicates whether a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *win the game or not as* $\mathsf{CP}-\mathsf{PRF}-\mathsf{AP}_{PD}^{\langle\mathsf{KeyGen},\mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ *or* $\mathsf{CP}-\mathsf{PRF}-\mathsf{AP}_{ID}^{\langle\mathsf{KeyGen},\mathsf{Protect},\mathsf{Eval}\rangle}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ *depending on if we consider the security with respect to the product distribution (PD) or to the identical distribution (ID).*

Definition 24 (Indistinguishable Anti-Piracy Security). *A pseudorandom function copy-protection scheme* $\langle\mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval}\rangle$ *has* indistinguishable anti-piracy security *with respect to the product distribution if no* QPT *adversary can win the piracy game above (with identical distribution challenge phase) with probability significantly greater than* $1/2$*. More precisely, for any triple of* QPT *adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*:*

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

Furthermore, we say that such a scheme has indistinguishable anti-piracy security *with respect to the product distribution if no* QPT *adversary can win the piracy game above (with identical distribution challenge phase) with probability significantly greater than* $1/2$*. More precisely, for any triple of* QPT *adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*:*

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

Theorem 8. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions, and Conjecture 1 (resp. Conjecture 2), there exists a pseudorandom function copy-protection scheme with indistinguishable anti-piracy security with respect to the product distribution (resp. with respect to the identical distribution).*

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

6 Copy-Protection of Point Functions in the Plain Model

In this section, we present the definition of copy-protection of point functions [Aar09]. 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.[6]Through all this section, λ denotes a security parameter and $n = \mathsf{poly}(\lambda)$.

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

6.1 Definitions

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

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

Definition 25 (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. More precisely, the game is played by a triple of adversaries (Alice, Bob and Charlie): Alice splits the quantum state and, in order to test both shares, they are sent to Bob and Charlie as well as the challenge (a point) who are asked to return the evaluation of the function on this point. 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, for each share, a challenge is sampled independently to be either the point y or another freshly sampled random point; in the second variant, the challenges are either both y, or both x.

Definition 26 (Piracy Game for Copy-Protection of Point Functions). *We define below a* piracy game for copy-protection of point functions*, parametrized 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 a triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*. 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:*** $\mathcal{A}$ *prepares a bipartite quantum state* σ_{12}*, then sends* σ_1 *to* $\mathcal{B}$ *and* σ_2 *to* $\mathcal{C}$*.*
- ***Challenge phase (product distribution):***
  - *The challenger samples* $b_1, b_2 \leftarrow\$ \{0,1\}$*, and* $x_1, x_2 \leftarrow\$ \{0,1\}^n$*.*
  - *If* $b_1 = 0$*, the challenger sends* y *to* $\mathcal{B}$*; otherwise the challenger sends* x_1*.*
  - *Similarly, if* $b_2 = 0$*, the challenger sends* y *to* $\mathcal{C}$*; otherwise the challenger sends* x_2*.*
- ***Challenge phase (identical distribution):***
  - *The challenger samples* $b \leftarrow\$ \{0,1\}$*, and* $x \leftarrow\$ \{0,1\}^n$*.*
  - *If* $b = 0$*, the challenger sends* y *to both* $\mathcal{B}$ *and* $\mathcal{C}$*; otherwise, the challenger send them* x*.*

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game with respect to the product distribution if $\mathcal{B}$ returns $b'_1 = b_1$ and $\mathcal{C}$ returns $b'_2 = b_2$; and with respect to the identical distribution if both $\mathcal{B}$ and $\mathcal{C}$ return b.

We denote the random variable that indicates whether a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win the game or not as $\mathsf{CP-AP}_{PD}^{\langle \mathsf{Protect}, \mathsf{Eval} \rangle}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ or as $\mathsf{CP-AP}_{ID}^{\langle \mathsf{Protect}, \mathsf{Eval} \rangle}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$ depending on which variant of the game we consider (PD and ID respectively denote the product and identical distributions).

Definition 27 (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 triple of QPT adversaries can win the piracy game above (with the product distribution challenge phase) with probability significantly greater than 1/2. More precisely, for any triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:*

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

Similarly, 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 the identical distribution challenge phase) with probability significantly greater than 1/2. More precisely, for any triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:*

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

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

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

Theorem 10. *Assuming the underlying pseudorandom functions copy-protection scheme has anti-piracy security with respect to the identical distribution, Construction 2 has correctness and anti-piracy security with respect to the identical distribution.*

Proof of Theorems 9 and 10. (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 $\mathsf{PRF}.\mathsf{Eval}(\rho_\mathsf{k}, x) = \mathsf{PRF}(\mathsf{k}, y)$, 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 with respect to the product distribution (resp. identical distribution) of the underlying copy-protection of pseudorandom function scheme. In both cases, the reduction is simply the identity. □

Corollary 4. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions, and Conjecture 1 (resp. Conjecture 2), there exists a point functions copy-protection scheme with correctness and anti-piracy security with respect to the product distribution (resp. the identical distribution).*

Proof. This result follows directly from Theorem 8. □

7 Unclonable Encryption in the Plain Model

In this section, we introduce the notion of unclonable encryption [BL20] and present a construction in the plain model. Our construction uses a pseudorandom function copy-protection scheme with anti-piracy security with respect to the product distribution (Definition 24) 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]. Through all this section, we refer to symmetric unclonable encryption simply as unclonable encryption, and use public key unclonable encryption to denote the public key version.

7.1 Definitions

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

Definition 28 (One-Time Unclonable Encryption Scheme). *A 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:

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

Indistinguishable anti-piracy security. We now define indistinguishable anti-piracy security of a 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 chosen by the adversary at the beginning of the game - and is asked to split it such that both shares can be used to guess which message has been encrypted. Note 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, before asking them to choose the pair of messages.

Definition 29 (Piracy Game for a One-Time Unclonable Encryption Scheme). *We define below a* piracy game for one-time unclonable encryption*, parametrized by a one-time unclonable encryption scheme* $\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$*, and a security parameter* λ*. This game is between a challenger and a triple of adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

- ***Setup phase:***
  - $\mathcal{A}$ *sends a message pair* $(m_0, m_1) \in \mathcal{M}^2$ *to the challenger.*

– *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:*** $\mathcal{A}$ *prepares a bipartite quantum state* σ_{12}*, then sends* σ_1 *to* $\mathcal{B}$*, and* σ_2 *to* $\mathcal{C}$*.*
- ***Challenge phase:*** *The challenger sends* k *to both* $\mathcal{B}$ *and* $\mathcal{C}$*.*

$\mathcal{A}$*,* $\mathcal{B}$*, and* $\mathcal{C}$ *win the game if* $\mathcal{B}$ *returns* $b'_1 = b$ *and* $\mathcal{C}$ *returns* $b'_2 = b$*.*

We denote the random variable that indicates whether a triple of adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *win the game or not as* $\mathsf{UncEnc} - \mathsf{AP}^{\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C})$*.*

Definition 30 (Indistinguishable Anti-Piracy Security of an Unclonable Encryption Scheme). *A one-time unclonable encryption scheme* $\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ *has* indistinguishable anti-piracy security *if no triple of QPT adversaries can win the piracy game above with probability significantly greater than* $1/2$*.*

More precisely, for any triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$

$$\Pr\Big[\mathsf{UncEnc} - \mathsf{AP}^{\langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle}(1^\lambda, \mathcal{A}, \mathcal{B}, \mathcal{C}) = 1\Big] \leq 1/2 + \mathsf{negl}(\lambda).$$

7.2 Construction

In this subsection, we present a construction of a 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 $\mathsf{k}_P \leftarrow \mathsf{PRF}.\mathsf{KeyGen}(1^\lambda)$ and compute $\rho_{\mathsf{k}_P} \leftarrow \mathsf{PRF}.\mathsf{Protect}(\mathsf{k}_P)$. – 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 11. *Assume* $\mathsf{PRF}.\langle \mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval} \rangle$ *has indistinguishable anti-piracy security with respect to the identical distribution (Definition 24). 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 $\mathsf{PRF}.\mathsf{Eval}(\rho_{\mathsf{k}_P}, \mathsf{k}_S \oplus r) \neq \mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$ 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 $\mathsf{PRF}.\mathsf{Eval}(\rho_{\mathsf{k}_P}, \mathsf{k}_S \oplus r) = y$ 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 triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, the probability that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ 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:** $\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$, and σ_2 to $\mathcal{C}$.
- **Challenge phase:** The challenger sends k_S to both $\mathcal{B}$ and $\mathcal{C}$.

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game if $\mathcal{B}$ returns $b'_1 = b$ and $\mathcal{C}$ returns $b'_2 = b$.

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 $c_1 := \mathsf{PRF}(\mathsf{k}_P, x)$ 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 $\mathcal{A}$ in the setup phase, and sending k_S to $\mathcal{B}$ and $\mathcal{C}$, we send only $\mathsf{PRF}(\mathsf{k}_P, \mathsf{k}_S \oplus r)$ to $\mathcal{A}$ in the setup phase, and send either k_S or k'_S to $\mathcal{B}$ and $\mathcal{C}$- 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:** $\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$, and σ_2 to $\mathcal{C}$.
- **Challenge phase:** The challenger sends k_S to both $\mathcal{B}$ and $\mathcal{C}$ if $b = 0$, otherwise the challenger sends k'_S.

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game if $\mathcal{B}$ returns $b'_1 = b$ and $\mathcal{C}$ returns $b'_2 = b$.

We now reduce the game G_3 from the piracy game of the underlying pseudorandom function copy-protection scheme with respect to the product distribution. Assume that there exists a triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ who wins G_3 with advantage δ. We construct a triple of QPT adversaries $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ who wins the piracy game of the underlying pseudorandom function copy-protection scheme with respect to the product distribution with the same advantage δ. $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ behave in the following way.

- $\mathcal{A}'$, on input a quantum protected pseudorandom function key ρ_{k} and a pseudorandom function image $y := \mathsf{PRF}(\mathsf{k}, x)$:
  - samples $r \leftarrow\$ \{0,1\}^n$ (note that, by defining $\mathsf{k}_S := x \oplus r$, we can write x as $\mathsf{k}_S \oplus r$);
  - runs $\mathcal{A}$ on $(r, y, \rho_{\mathsf{k}})$ to get σ_{12};
  - prepares the bipartite quantum state σ'_{12} where the first register is $\sigma_1 \otimes |r\rangle\langle r|$ and the second one is $\sigma_2 \otimes |r\rangle\langle r|$;
  - sends σ'_1 to $\mathcal{B}$ and σ'_2 to $\mathcal{C}$.

- $\mathcal{B}'$, on input σ_1' and x:
  - computes $\mathsf{k} := r \oplus x$;
  - runs $\mathcal{B}$ on (σ_1, k);
  - returns the outcome.
- $\mathcal{C}'$ is defined similarly by replacing the "1" indices by "2" indices.

The inputs given to $\mathcal{B}$ and $\mathcal{C}$ follow the same distribution as their inputs in G_3. Thus, $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ win the game with the same advantage as $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, which concludes the proof.

Corollary 5. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions, and Conjecture 2, there exists a one-time unclonable encryption scheme with correctness and indistinguishable anti-piracy security for* 1*-bit long messages.*

7.3 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 12. *Assume* $\mathsf{PRF}.\langle\mathsf{KeyGen}, \mathsf{Protect}, \mathsf{Eval}\rangle$ *has indistinguishable anti-piracy security with respect to the product distribution. 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 hybrid 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 6. *Assuming the existence of post-quantum indistinguishability obfuscation, one-way functions, compute-and-compare obfuscation for the class of unpredictable distributions, and Conjecture 2, there exists a public-key reusable unclonable encryption scheme with correctness and indistinguishable anti-piracy security for* 1*-bit long messages.*

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`.

CGLZR23. Alper Cakan, Vipul Goyal, Chen-Da Liu-Zhang, and João Ribeiro. Unbounded leakage-resilience and intrusion-detection in a quantum world. Cryptology ePrint Archive, Paper 2023/410, 2023. `https://eprint.iacr.org/2023/410`.

CHV23. Céline Chevalier, Paul Hermouet, and Quoc-Huy Vu. Semi-quantum copy-protection and more. TCC 2023, 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.

Got02. Daniel Gottesman. Uncloneable encryption. *arXiv preprint quant-ph/0210062*, 2002.

GZ20. Marios Georgiou and Mark Zhandry. Unclonable decryption keys. Cryptology ePrint Archive, Report 2020/877, 2020. https://eprint.iacr.org/2020/877.

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 with respect to both the product and the identical distributions.

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 5.

– Return $\rho_{\mathsf{k}} \coloneqq \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \widehat{\mathsf{P}}\right)$.

$\mathsf{Eval}(1^\lambda, \rho_{\mathsf{k}}, x)$:

– Parse $\rho_{\mathsf{k}} = \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \widehat{\mathsf{P}}\right)$.

– Parse x as $x \coloneqq x^{(0)}\|x^{(1)}\|x^{(2)}$.

– For each $i \in [\![1, \ell_0]\!]$, if $x_i^{(0)} = 1$, apply $\mathsf{H}^{\otimes n}$ to $|A_{i,s_i,s'_i}\rangle$; if $x_i^{(0)} = 0$, 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. 5. Program P.

A.1 Proof of Indistinguishable Anti-Piracy Security With Respect to the Product Distribution

In this subsection, we prove that the construction above has indistinguishable anti-piracy security with respect to the product distribution. This proof and the next one (for the identical distribution) 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 triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, the probability that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins G_i is negligibly close to the probability that they win G_j.

Procedure. We define the $\mathsf{GenTrigger}$ procedure (Figure 6) 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.

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

Lemma 6. *Assuming post-quantum* iO *and one-way functions, any efficient* QPT *algorithm* $\mathcal{A}$ *cannot win the following game with non-negligible advantage:*

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. 6. $\mathsf{GenTrigger}$ procedure.

- *A challenger samples* $\mathsf{k}_1 \leftarrow \mathsf{Setup}(1^\lambda)$ *and prepares a quantum key* $\rho_\mathsf{k} := (\{|A_{i,s_i,s'_i}\rangle\}_{i \in [\![1,l_0]\!]}, \mathsf{iO}(\mathsf{P}))$ *(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 with respect to the product distribution 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:** $\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$ and σ_2 to $\mathcal{C}$.
- **Challenge phase (product distribution):**
  - The challenger samples two bits $b_1, b_2 \leftarrow\!\!\$\ \{0,1\}$, and two inputs $x_1, x_2 \leftarrow\!\!\$\ \{0,1\}^n$.
  - If $b_1 = 0$, the challenger sends x to $\mathcal{B}$; otherwise, the challenger sends x_1.
  - Similarly, if $b_2 = 0$, the challenger sends x to $\mathcal{C}$; otherwise, the challenger sends x_2.
- **Challenge phase (identical distribution):**
  - The challenger samples a bit $b \leftarrow\!\!\$\ \{0,1\}$, and an input $x_0 \leftarrow\!\!\$\ \{0,1\}^n$.
  - If $b = 0$, the challenger sends x to both $\mathcal{B}$ and $\mathcal{C}$; otherwise, the challenger sends them x_0.

  $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game with respect to the product distribution if $\mathcal{B}$ returns b_1 and $\mathcal{C}$ returns b_2; and with respect to the identical distribution if both $\mathcal{B}$ and $\mathcal{C}$ return b.

Game G_1: In this game, we replace the challenges x, x_0, x_1, and x_2 by their trigger inputs for both $\mathcal{B}$ and $\mathcal{C}$. More precisely, the challenge phases become the following.

- **Challenge phase (product distribution):**

– The challenger samples two bits $b_1, b_2 \leftarrow\$ \{0,1\}$, and two inputs $x_1, x_2 \leftarrow\$ \{0,1\}^n$.

– The challenger computes the two images $y_1 := \mathsf{PRF}(\mathsf{k}_1, x_1)$ and $y_2 := \mathsf{PRF}(\mathsf{k}_1, x_2)$.

– If $b_1 = 0$, the challenger sends $\mathsf{GenTrigger}(x^{(0)}, y)$ to $\mathcal{B}$; otherwise, the challenger sends $\mathsf{GenTrigger}(x_1^{(0)}, y_1)$.

– Similarly, if $b_2 = 0$, the challenger sends $\mathsf{GenTrigger}(x^{(0)}, y)$ to $\mathcal{C}$; otherwise, the challenger sends $\mathsf{GenTrigger}(x_2^{(0)}, y_2)$.

- **Challenge phase (identical distribution):**
  – The challenger samples a bit $b \leftarrow\$ \{0,1\}$, and an input $x_0 \leftarrow\$ \{0,1\}^n$.

  – The challenger computes the image $y_0 := \mathsf{PRF}(\mathsf{k}_1, x_0)$.

  – If $b = 0$, the challenger sends $\mathsf{GenTrigger}(x^{(0)}, y)$ to both $\mathcal{B}$ and $\mathcal{C}$; otherwise, the challenger sends them $\mathsf{GenTrigger}(x_0^{(0)}, y_0)$.

The trigger's inputs lemma (Lemma 6) implies that G_1 is negligibly close to G_0.

Game G_2: In this game, we replace y (in the setup phase) and y_0, y_1, y_2 (in the challenge phases) 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 images are statistically close to independently random bitstrings. 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 encryption 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 $\{(\mathsf{P}_i^0, \mathsf{P}_i^1)\}_{i \in [\![1,\ell_0]\!]}$. 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:** $\mathcal{A}$ prepares a bipartite quantum state σ_{12}, then sends σ_1 to $\mathcal{B}$ and σ_2 to $\mathcal{C}$.
- **Challenge phase (product distribution):**
  – The challenger samples two bits $b_1, b_2 \leftarrow\$ \{0,1\}$, and two inputs $x_1, x_2 \leftarrow\$ \{0,1\}^n$.

  – The challenger also samples a random set of coins $r \leftarrow\$ \{0,1\}^{\mathsf{poly}(\lambda)}$ for the encryption, and two bitstrings $y_1, y_2 \leftarrow\$ \{0,1\}^m$.

  – If $b_1 = 0$, the challenger computes $(x, \mathsf{Q}) \leftarrow \mathsf{SD.Enc}(\mathsf{pk}, y; r)$ and sends $\mathsf{GenTrigger}(x^{(0)}; y)$ to $\mathcal{B}$; otherwise the challenger computes $(x_1, \mathsf{Q}) \leftarrow \mathsf{SD.Enc}(\mathsf{pk}, y_1; r)$ and sends $\mathsf{GenTrigger}(x_1^{(0)}, y_1)$.

  – Similarly, if $b_2 = 0$, the challenger computes $(x, \mathsf{Q}) \leftarrow \mathsf{SD.Enc}(\mathsf{pk}, y; r)$ and sends $\mathsf{GenTrigger}(x^{(0)}, y)$ to $\mathcal{C}$; otherwise the challenger computes $(x_2, \mathsf{Q}) \leftarrow \mathsf{SD.Enc}(\mathsf{pk}, y_2; r)$ and sends $\mathsf{GenTrigger}(x_2^{(0)}, y_2)$.

  $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ win the game with respect to the product distribution if $\mathcal{B}$ returns b_1 and $\mathcal{C}$ returns b_2; and with respect to the identical distribution if both $\mathcal{B}$ and $\mathcal{C}$ return b.

Reduction from single-decryptor's piracy game for the product distribution. We reduce the game G_3 with respect to the product distribution to the piracy game of the underlying single-decryptor with respect to the product distribution. Assume that there exists a triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ who wins the last hybrid G_3 with respect to the product distribution with advantage δ. We construct a QPT adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ who wins the piracy game of the single-decryptor scheme of Construction 1 with respect to the product distribution with the same advantage δ.

- $\mathcal{A}'$, on input a quantum key ρ_{sk} and the associated public key pk:
  - 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 $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$;
  - samples $y, y_1, y_2 \leftarrow\$ \{0,1\}^m$;
  - runs $\mathcal{A}$ on $(\rho_{\mathsf{sk}}, \widehat{\mathsf{P}}, y)$ to get σ_{12};
  - then prepares $\sigma_1' := \sigma_1 \otimes |\mathsf{k}_2, \mathsf{k}_3\rangle\langle\mathsf{k}_2, \mathsf{k}_3|$ and $\sigma_2' := \sigma_2 \otimes |\mathsf{k}_2, \mathsf{k}_3\rangle\langle\mathsf{k}_2, \mathsf{k}_3|$;
  - and finally sends σ_1' to $\mathcal{B}$, σ_2' to $\mathcal{C}$, and the pairs of messages (y, y_1), (y, y_2) to the challenger.
- $\mathcal{B}'$, on input σ_1' and a ciphertext (r, Q):
  - computes $x' \leftarrow \mathsf{GenTrigger}(r; \mathsf{Q})$;
  - runs $\mathcal{B}$ on (σ_1, x') and returns the outcome.
- $\mathcal{C}'$ is defined similarly as $\mathcal{B}'$ by replacing σ_1' by σ_2'.

The adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ perfectly simulates $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, and thus $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ 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. We reduce the game G_3 with respect to the identical distribution to the piracy game of the underlying single-decryptor with respect to the identical distribution. Assume that there exists a triple of QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ who wins the last hybrid G_3 with respect to the identical distribution with advantage δ. We construct a QPT adversary $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ who wins the piracy game of the single-decryptor scheme of Construction 1 with respect to the identical distribution with the same advantage δ.

- $\mathcal{A}'$, on input a quantum key ρ_{sk} and the associated public key pk:
  - 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 $\widehat{\mathsf{P}} \leftarrow \mathsf{iO}(\mathsf{P})$;
  - samples $y, y_0 \leftarrow\$ \{0,1\}^m$;
  - runs $\mathcal{A}$ on $(\rho_{\mathsf{sk}}, \widehat{\mathsf{P}}, y)$ to get σ_{12};
  - then prepares $\sigma_1' := \sigma_1 \otimes |\mathsf{k}_2, \mathsf{k}_3\rangle\langle\mathsf{k}_2, \mathsf{k}_3|$ and $\sigma_2' := \sigma_2 \otimes |\mathsf{k}_2, \mathsf{k}_3\rangle\langle\mathsf{k}_2, \mathsf{k}_3|$;
  - and finally sends σ_1' to $\mathcal{B}$, σ_2' to $\mathcal{C}$, and the pairs of messages (y, y_0), (y, y_0) to the challenger.
- $\mathcal{B}'$, on input σ_1' and a ciphertext (r, Q):
  - computes $x' \leftarrow \mathsf{GenTrigger}(r; \mathsf{Q})$;
  - runs $\mathcal{B}$ on (σ_1, x') and returns the outcome.
- $\mathcal{C}'$ is defined similarly as $\mathcal{B}'$ by replacing σ_1' by σ_2'.

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