One-out-of-Many Unclonable Cryptography: Definitions, Constructions, and More

Fuyuki Kitagawa[†] and Ryo Nishimaki[†]

[†]NTT Social Informatics Laboratories, Tokyo, Japan

{fuyuki.kitagawa.yh,ryo.nishimaki.zk}@hco.ntt.co.jp

February 22, 2023

Abstract

The no-cloning principle of quantum mechanics enables us to achieve amazing unclonable cryptographic primitives, which is impossible in classical cryptography. However, the security definitions for unclonable cryptography are tricky. Achieving desirable security notions for unclonability is a challenging task. In particular, there is no indistinguishable-secure unclonable encryption and quantum copy-protection for single-bit output point functions in the standard model. To tackle this problem, we introduce and study relaxed but meaningful security notions for unclonable cryptography in this work. We call the new security notion *one-out-of-many* unclonable security.

We obtain the following results.

- We show that one-time strong anti-piracy secure secret key single-decryptor encryption (SDE) implies one-out-of-many indistinguishable-secure unclonable encryption.
- We construct a one-time strong anti-piracy secure secret key SDE scheme in the standard model from the LWE assumption.
- We construct one-out-of-many copy-protection for single-bit output point functions from one-out-of-many indistinguishable-secure unclonable encryption and the LWE assumption.
- We construct one-out-of-many unclonable predicate encryption (PE) from one-out-of-many indistinguishable-secure unclonable encryption and the LWE assumption.

Thus, we obtain one-out-of-many indistinguishable-secure unclonable encryption, one-out-of-many copy-protection for single-bit output point functions, and one-out-of-many unclonable PE in the standard model from the LWE assumption. In addition, our one-time SDE scheme is the first SDE scheme that does not rely on any oracle heuristics and strong assumptions such as indistinguishability obfuscation and witness encryption.

Contents

1 Introduction

1.1 Background

Unclonable encryption and quantum copy-protection. Quantum information enables us to achieve new cryptographic primitives beyond classical cryptography. Especially the no-cloning principle of quantum information has given rise to amazing unclonable cryptographic primitives. This includes quantum money [Wie83], quantum copy-protection [Aar09], unclonable encryption [BL20], one-shot signatures [AGKZ20], single-decryptor encryption [GZ20, CLLZ21], and many more. In this work, we mainly focus on unclonable encryption and quantum copy-protection.

Broadbent and Lord [BL20] introduced unclonable encryption. Unclonable encryption is a one-time secure secret key encryption where a plaintext is encoded into a quantum ciphertext that is impossible to clone. More specifically, an unclonable encryption scheme encrypts a plaintext m into a quantum ciphertext ct. The user who has the secret key can recover m from ct. The security notion of unclonable encryption ensures that it is impossible to convert ct into possibly entangled bipartite states ct_1 and ct_2, both of which can be used to recover m when the secret key is given. Ananth and Kaleoglu [AK21] later introduced unclonable public key encryption. Unclonable encryption has interesting applications, such as preventing cloud storage attacks where an adversary steals ciphertexts from cloud storage with the hope that they can be decrypted if the secret key is leaked later.

Quantum copy-protection [Aar09] is a cryptographic primitive that prevents users from creating pirated copies of a program. More specifically, a quantum copy-protection scheme transforms a classical program C into a quantum program ρ that is impossible to copy. We can compute $C(x)$ for any input x using ρ. The security notion of copy-protection ensures that it is impossible to convert ρ into possibly entangled bipartite states ρ_1 and ρ_2, both of which can be used to compute C. As shown by Ananth and La Placa [AL21], it is impossible to have quantum copy-protection for general unlearnable functions. For this reason, recent works have been studying quantum copy-protection for a simple class of functions such as point functions [CMP20, AK21, AKL+22, AK22].[1] Moreover, Coladangelo, Majenz, and Poremba [CMP20] show that quantum copy-protection for point functions can be transformed into quantum copy-protection for a more general class of compute-and-compare programs (C&C programs) that includes conjunctions with wildcards, affine testers, plaintext testers, and so on. We focus on quantum copy-protection for point functions in this work.

Definition of unclonability: one-wayness and indistinguishability. To describe our research questions and contributions, we first explain a general template for unclonable security games played by a tuple of three adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$. The template is common to unclonable encryption and quantum copy-protection. In the first stage, the challenger sends a challenge copy-protected object (such as a quantum ciphertext in unclonable encryption and copy-protected program in quantum copy-protection) to an adversary $\mathcal{A}_0$. Then, $\mathcal{A}_0$ generates possibly entangled bipartite states ρ_1 and ρ_2 and sends ρ_α to $\mathcal{A}_\alpha$ for $\alpha \in \{1,2\}$. In the second phase, the challenger sends extra information (such as secret keys in unclonable encryption and some inputs for the program in quantum copy-protection) to $\mathcal{A}_1$ and $\mathcal{A}_2$, and they try to compute target information about the copy-protected object (such as plaintexts in unclonable encryption and computation results on the given inputs in quantum copy-protection). Here, $\mathcal{A}_1$ and $\mathcal{A}_2$ are not allowed to communicate. If both $\mathcal{A}_1$ and $\mathcal{A}_2$ succeed in computing target information, the adversaries win. Note that if $\mathcal{A}_\alpha$ has the original objects, computing target information is easy.

Using the above game, we can define both one-wayness-based notion and indistinguishability-based notion depending on which one the task of $\mathcal{A}_1$ and $\mathcal{A}_2$ is, recovering entire bits of high min-entropy information or detecting 1-bit information. Similarly to standard security notions, indistinguishability-based one is more general and enables us to have a wide range of applications. Also, indistinguishability-based unclonability usually implies standard cryptographic security notions, but one-wayness-based unclonability does not necessarily imply them.[2] In this work, we focus on

[1] Some lines of works [CLLZ21, LLQZ22] studied quantum copy-protection for cryptographic functionalities that are not captured by C&C programs. Quantum copy-protections for cryptographic functionalities have different names, such as unclonable decryption or single decryptor encryption. In this work, unless stated otherwise, we use the term quantum copy-protection to indicate quantum copy-protection for point functions. For the previous works on quantum copy-protection for cryptographic functionalities, see Section 1.5.

[2] For example, indistinguishability-based unclonability for unclonable encryption implies (one-time) IND-CPA security, but one-wayness-based unclonability does not.

indistinguishability-based unclonability notions.

Toward indistinguishability-based unclonability in the standard model. Unclonable encryption and quantum copy-protection have been studied actively and there are many constructions. Although we have constructions with one-wayness-based unclonability from standard assumptions in the standard model [BL20, AK21], we have constructions with indistinguishability-based unclonability *only in the oracle model*, in both unclonable encryption and quantum copy-protection [AKL+22]. Ananth, Kaleoglu, Li, Liu, and Zhandry [AKL+22] proposed the only indistinguishability-based secure unclonable encryption and quantum copy-protection schemes. Their proof technique is highly specific to the oracle model. Thus, it still remains elusive to achieve unclonable encryption and quantum copy-protection with indistinguishability-based unclonability in the standard model.

Given the above situation, it is natural and reasonable to explore relaxed but meaningful indistinguishability-based unclonability and ask whether the notion can be achieved in the standard model. Such a standard model construction with a relaxed notion would provide new insights toward achieving full-fledged indistinguishability-based unclonability in the standard model.

1.2 Our Result

Our contributions are proposing new definitions for unclonability and constructions satisfying them under the LWE assumption in the standard model.

New definitions: one-out-of-many unclonability notions. We introduce a relaxed indistinguishability-based unclonability for unclonable encryption and copy-protection, called *one-out-of-many* unclonable security. This notion captures a meaningful unclonability, as we argue below. It guarantees that no adversary can generate n copies with probability significantly better than $\frac{1}{n}$ for any n. Thus, roughly speaking, it guarantees that the expected number of successful target objects generated by any copying adversary is less than 1. We define one-out-of-many unclonability by extending the unclonable game played by a tuple of three adversaries into a game played by $n+1$ adversaries, where $n \geq 2$ is arbitrary.

Although one-out-of-many unclonable security looks weaker than existing unclonable security, it is useful in some applications. For example, suppose we publish many quantum objects, say ℓ objects. Then, one-out-of-many security guarantees that no matter what copying attacks are applied to those objects, there are expected to be only ℓ objects on average in this world. Another nice property of one-out-of-many security is that it implies standard cryptographic security notions. For example, one-out-of-many unclonability for unclonable encryption implies (one-time) IND-CPA security. This result contrasts one-wayness-based unclonability notions that do not necessarily imply standard indistinguishability notions.

Unclonable encryption in the standard model via single decryptor encryption. We provide unclonable encryption satisfying one-out-of-many unclonability under the LWE assumption in the standard model. We obtain this result as follows.

We first define one-out-of-many unclonability for (one-time) single decryptor encryption (SDE) [GZ20, CLLZ21]. One-time SDE is a dual of unclonable encryption in the sense that one-time SDE is a one-time secret key encryption scheme where a secret key is encoded into a quantum state, and its security notion guarantees that any adversary cannot copy the quantum secret key. Under appropriate definitions, it is possible to back and forth between unclonable encryption and one-time SDE, as shown by Georgiou and Zhandry [GZ20]. We show that we can transform any one-time SDE with one-out-of-many unclonability to unclonable encryption with one-out-of-many unclonability.

We then show that we can obtain one-time SDE with one-out-of-many unclonability from the LWE assumption. More specifically, assuming the LWE assumption, we construct one-time SDE satisfying strong anti-piracy introduced by Coladangelo, Liu, Liu, and Zhandry [CLLZ21], and show that strong anti-piracy implies one-out-of-many unclonability. Combining this result with the above transformation, we obtain unclonable encryption with one-out-of-many security under the LWE assumption.

Theorem 1.1 (informal). *Assuming the LWE assumption holds, there exists strong anti-piracy secure one-time SDE.*

Theorem 1.2 (informal). *Assuming the LWE assumption holds, there exists one-out-of-many indistinguishable-secure unclonable encryption.*

To achieve one-time SDE satisfying strong anti-piracy, we develop a technique enabling us to use a BB84 [BB14] state as a copy-protected secret key. Our crucial tool is single-key ciphertext-policy functional encryption (CPFE) with a succinct key, which we introduce in this work. We instantiate it with hash encryption (HE) [DGHM18] implied by the LWE assumption. The technique of the post-quantum watermarking by Kitagawa and Nishimaki [KN22] inspired our proof technique. We emphasize that *our one-time SDE scheme is the first SDE scheme that does not require either oracle heuristic or strong assumptions such as indistinguishability obfuscation and witness encryption.*

Quantum copy-protection in the standard model via unclonable encryption. We propose quantum copy protection for single-bit output point functions based on unclonable encryption and the LWE assumption. Known constructions from unclonable encryption [CMP20, AK21] support only multi-bit output point functions.[3] Although we formally prove this result with our new one-out-of-many security notion, our construction also works under standard indistinguishability-based unclonability definitions defined using three adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$.

Theorem 1.3 (informal). *Assuming (resp. one-out-of-many) indistinguishable-secure unclonable encryption and the LWE assumption holds, there exits (resp. one-out-of-many) copy-protection for single-bit output point functions.*

Unclonable predicate encryption. Using the technique proposed by Ananth and Kaleoglu [AK21], we can convert our one-out-of-many secure unclonable encryption into one-out-of-many secure unclonable public-key encryption. We construct a one-out-of-many unclonable predicate encryption (PE) scheme from one-out-of-many unclonable encryption and the LWE assumption. PE is a stronger variant of attribute-based encryption (ABE). ABE is an advanced public-key encryption system where we can generate a user secret key for an attribute x and a ciphertext of a message m under a policy P.[4] We can decrypt a ciphertext and obtain m if $P(x) = 1$. In PE, ciphertexts hide not only plaintexts but also policies.

Theorem 1.4 (informal). *Assuming (resp. one-out-of-many) indistinguishable-secure unclonable encryption and the LWE assumption holds, there exits (resp. one-out-of-many) unclonable PE.*

1.3 Concurrent and Independent Work

Ananth, Kaleoglu, and Liu [AKL23] introduce a new framework called cloning games to study unclonable cryptography. They obtain many implications to unclonable cryptography thanks to the framework. In particular, they obtain information-theoretically secure one-time SDE in the standard model. The scheme is *single-bit* encryption while our computationally secure scheme is multi-bit encryption. *Note that we do not know how to obtain multi-bit encryption from single-bit one via parallel repetition and the standard hybrid argument in unclonable cryptography.* Thus, their results on SDE is incomparable with ours. We also note that it is not clear whether we can obtain one-out-of-many indistinguishable-secure unclonable encryption from their SDE, since it seems that we need strong anti-piracy secure one-time SDE to obtain one-out-of-many secure one, but their scheme is only proved to satisfy (non-strong) indistinguishability-based security. For the detailed reason on the need of strong anti-piracy for the implication, see the "SDE from LWE" paragraph in Section 1.4.

1.4 Technical Overview

We provide a high-level overview of our techniques in this subsection. In this paper, standard math or sans serif font stands for classical algorithms and classical variables. The calligraphic font stands for quantum algorithms and the calligraphic font and/or the bracket notation for (mixed) quantum states.

[3] One might think that copy protection for multi-bit output point functions implies that for single-bit output point functions. However, this is not the case. This is because the security of copy protection for multi-bit output point functions usually relies on the high min-entropy of the multi-bit output string, and it is broken if the output string does not have enough entropy as in the case of single-bit output. Realizing copy protection for single-bit output point function is challenging in the sense that we have to achieve the security without relying on the entropy of the output string.

[4] We focus on ciphertext-policy ABE in this work.

Relaxed definition of unclonable cryptography. We introduce relaxed security notions for unclonable cryptography called *one-out-of-many* unclonable security that roughly guarantees that no adversary can generate n copies with a probability significantly better than $\frac{1}{n}$ for any n. The one-out-of-many unclonability is defined by extending the unclonable game played by $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ explained in Section 1.1. The one-out-of-many unclonability game is played by a tuple of $n+1$ adversaries $(\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$, where n is arbitrary. At the first stage of the game, $\mathcal{A}_0$ is given a single quantum object, generates possibly entangled n-partite states $\rho_1, \ldots, \rho_n$, and sends ρ_k to $\mathcal{A}_k$ for $k \in \{1, \ldots, n\}$. At the second stage, the challenger selects one of $(\mathcal{A}_1, \ldots, \mathcal{A}_n)$ by a random $\alpha \leftarrow \{1, \ldots, n\}$ and sends additional information *only to* $\mathcal{A}_\alpha$, and only $\mathcal{A}_\alpha$ tries to detect the target 1-bit information. Recall that we focus on indistinguishability-based setting. The one-out-of-many unclonability guarantees that the adversary cannot win this game with probability significantly better than the trivial winning probability $\frac{1}{n} \cdot 1 + \frac{n-1}{n} \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{2n}$. The definition captures the above intuition because if $\mathcal{A}_0$ could make n copies with probability $\frac{1}{n} + \delta$ for some noticeable δ, the adversary would win the game with probability at least $(\frac{1}{n} + \delta) \cdot 1 + (1 - \frac{1}{n} - \delta) \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{2n} + \frac{\delta}{2}$.

We consider one-out-of-many unclonable security for the following security notions in this work: (1) (one-time) unclonable-indistinguishable security for unclonable encryption, (2) copy-protection security for single-bit output point functions, (3) (one-time) indistinguishability-based security for one-time SDE, (4) unclonable-simulation security for PE, which is introduced in this work.

The nice property of one-out-of-many security are as follows. One-out-of-many security implies standard cryptographic security notions. For example, one-out-of-many unclonability for unclonable encryption implies (one-time) IND-CPA security, and one for copy-protection implies distributional indistinguishability as virtual black-box obfuscation. Moreover, we can use one-out-of-many secure unclonable cryptographic primitives as a drop-in-replacement of standard indistinguishability-based unclonable cryptographic primitives if our goal is constructing a one-out-of-many secure unclonable cryptographic primitive (and vice versa). For example, the transformation from unclonable encryption to unclonable public-key encryption proposed by Ananth and Kaleoglu [AK21] works also in the one-out-of-many setting. Moreover, all of the generic constructions from an unclonable primitive to another unclonable primitive that we propose work in both standard (three adversary style) setting and one-out-of-many setting.

In addition to the above nice properties, we can prove that *one-out-of-many indistinguishability-based secure one-time SDE is equivalent to one-out-of-many unclonable-indistinguishable secure unclonable encryption*. In this work, we first obtain one-out-of-many indistinguishability-based secure one-time SDE from the LWE assumption, and using this equivalence, we obtain one-out-of-many secure unclonable encryption, copy protection for single-bit output point functions, and unclonable PE.

Before our work, Georgiou and Zhandry [GZ20] showed a transformation from one-time SDE to unclonable encryption under the standard three adversary style setting. Informed readers may think that by combining their result with the result by Coladangelo et al. [CLLZ21] (and the result by Culf and Vidick [CV22]), we can obtain indistinguishability-based unclonable encryption in the standard model based on indistinguishability obfuscation. However, this is not the case due to the fact that those two works used different definitions of the indistinguishability-based security for SDE, and we do not know any relation between them. For more details, see Remark 2.23.

SDE from LWE. We next explain how to obtain one-out-of-many indistinguishability-based secure one-time SDE. In fact, we obtain one-time SDE satisfying much stronger security notion called strong anti-piracy [CLLZ21] from the LWE assumption, and prove that *strong anti-piracy implies one-out-of-many indistinguishability-based security*. Below, we first introduce the definition of strong anti-piracy for one-time SDE, briefly explain the intuition of the implication, and finally present the high level ideas on how to realize strong anti-piracy secure one-time SDE from the LWE assumption.

Recall the general template of the security game for unclonability played by three adversaries $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ explained in Section 1.1. This template also captures the security game of strong anti-piracy for SDE. In strong anti-piracy security for SDE, $\mathcal{A}_0$ receives a copy-protected decryption key dK in the first stage. In the second stage, $\mathcal{A}_1$ and $\mathcal{A}_2$ outputs quantum decryptors $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, and the challenger tests whether both $\mathcal{D}_1$ and $\mathcal{D}_2$ are "good" (or "live") quantum decryptors [Zha20, CLLZ21]. Intuitively, good quantum decryptors can distinguish encryption of m_0 from that of m_1 with probability $\frac{1}{2} + \frac{1}{\mathrm{poly}(\lambda)}$, and do not lose the decryption capability even after its goodness was tested. Strong anti-piracy guarantees that the probability that both $\mathcal{D}_1$ and $\mathcal{D}_2$ are tested as good is negligible. See Definition 2.21 for the precise definition.

The intuition behind the implication from strong anti-piracy to one-out-of-many security is as follows. The one-out-of-many security game for a one-time SDE scheme played by a tuple of $n+1$ adversaries $(\mathcal{A}_0, \cdots, \mathcal{A}_n)$ is defined as follows. The first stage adversary $\mathcal{A}_0$ is given a quantum decryption key dk, generates possibly entangled n-partite states $\rho_1, \cdots \rho_n$, and sends ρ_k to the second stage adversary $\mathcal{A}_k$ for every $k \in \{1, \cdots, n\}$. In the second stage, only randomly chosen single second stage adversary $\mathcal{A}_\alpha$ is given the challenge ciphertext and required to guess the challenge bit. The n-partite state $(\rho_1, \cdots, \rho_n)$ generated by $\mathcal{A}_0$ can be regarded as a tuple of n quantum decryptors. If the one-time SDE scheme is strong anti-piracy secure, all n quantum decryptors *except one* must have success probabilities at most $1/2 + \mathsf{negl}(\lambda)$. Hence, the success probability of $(\mathcal{A}_0, \cdots, \mathcal{A}_{n+1})$ in the one-out-of-many security game is at most $1/n \cdot 1 + (n-1)/n \cdot (1/2 + \mathsf{negl}(\lambda)) = 1/2 + 1/2n + \mathsf{negl}(\lambda)$, which proves the one-out-of-many security. It seems that "strong" anti piracy is required for this argument and it is difficult to prove a similar implication from (non-strong) indistinguishability-based security defined by Coladangelo et al. [CLLZ21]. To formally prove the implication, we have to construct a reduction algorithm that finds two "good" decryptors from n decryptors output by $\mathcal{A}$. If the reduction attacks strong anti-piracy, it is sufficient to randomly pick two decryptors out of n since the reduction's goal is to output two "good" decryptors with inverse polynomial probability. However, if the reduction attacks (non-strong) indistinguishability-based security, the reduction cannot use such random guessing and needs to detect whether each decryptor is "good" since the reduction's goal is to make a distinguishing gap. We are considering the one-time setting where the adversaries are not given the encryption key. Thus, it seems difficult to perform such detection of "good" decryptors.

We next explain how to achieve strong anti-piracy secure one-time SDE based on the LWE assumption in the standard model. We use the monogamy of entanglement property of BB84 states [TFKW13] differently from the previous work on SDE [CLLZ21] that used the monogamy of entanglement property of coset states.

We combine BB84 states and ciphertext-policy FE with succinct key to achieve strong anti-piracy. We first explain the definition of single-key CPFE. A single-key CPFE scheme CPFE consists of three algorithms $(\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$. $\mathsf{FE.Setup}$ takes as input a string x and outputs a public key pk and a decryption key sk_x.[5] Here, we assume that x itself works as a decryption key sk_x for x, thus $\mathsf{FE.Setup}$ outputs only pk. We can achieve such a CPFE (we will explain later). $\mathsf{FE.Enc}$ takes as input pk and a circuit C, and outputs a ciphertext ct. We can decrypt ct with sk_x using $\mathsf{FE.Dec}$, and obtain $C(x)$. The single-key security of CPFE guarantees that $\mathsf{FE.Enc}(\mathsf{pk}, C_0)$ and $\mathsf{FE.Enc}(\mathsf{pk}, C_1)$ are computationally indistinguishable for an adversary who has a decryption key sk_x for x of its choice as long as $C_0(x) = C_1(x)$ holds.

Let $s[i]$ is the i-th bit of a string $s \in \{0,1\}^n$. Our one-time SDE scheme is as follows. The key generation algorithm generate a BB84 state $\left|x^\theta\right\rangle := H^{\theta[1]}|x[1]\rangle \otimes \cdots \otimes H^{\theta[n]}|x[n]\rangle$, where H is the Hadamard gate, and a public key $\mathsf{pk} \leftarrow \mathsf{FE.Setup}(x)$ of CPFE. It outputs an encryption key $\mathsf{ek} := (\theta, \mathsf{pk})$ and decryption key $\mathsf{dk} := \left|x^\theta\right\rangle$. Note that although our one-time SDE scheme is secret key encryption, an encryption key and a decryption key are different. The encryption algorithm takes as input the encryption key and a plaintext m, and generates a ciphertext $\mathsf{fe.ct} \leftarrow \mathsf{FE.Enc}(\mathsf{pk}, C[\mathsf{m}])$, where $C[\mathsf{m}]$ is a constant circuit that outputs m for all inputs. It outputs a ciphertext $(\theta, \mathsf{fe.ct})$. We can decrypt $\mathsf{fe.ct}$ and obtain m by recovering x from $\left|x^\theta\right\rangle$ and θ since x works as a decryption key of CPFE as we assumed. Intuitively, it is hard to copy $\mathsf{dk} = \left|x^\theta\right\rangle$ by the monogamy of entanglement property. The monogamy of entanglement property can be explained by the template of unclonable cryptography. In the first stage, $\mathcal{A}_0$ is given $\left|x^\theta\right\rangle$. In the second stage, $\mathcal{A}_1$ and $\mathcal{A}_2$ receive θ and try to output x. It is proved that the winning probability of the adversaries is exponentially small without any assumptions [TFKW13].

To prove the strong anti-piracy security, we need to extract x both from good decryptors $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively output by $\mathcal{A}_1$ and $\mathcal{A}_2$ to reduce the SDE security to the monogamy of entanglement property. The idea for the extraction is as follows. Let $\widetilde{C}[b, \mathsf{m}_0, \mathsf{m}_1, i]$ be a circuit that takes as input x and outputs $\mathsf{m}_{b \oplus x[i]}$. We estimate the probability that a good decryptor outputs the correct b when we feed $\mathsf{FE.Enc}(\mathsf{pk}, \widetilde{C}[b, \mathsf{m}_0, \mathsf{m}_1, i])$ to it. The security of CPFE guarantees that $\mathsf{FE.Enc}(\mathsf{pk}, \widetilde{C}[b, \mathsf{m}_0, \mathsf{m}_1, i])$ is indistinguishable from $\mathsf{FE.Enc}(\mathsf{pk}, C[\mathsf{m}_{b \oplus x[i]}])$ since $\widetilde{C}[b, \mathsf{m}_0, \mathsf{m}_1, i](x) = \mathsf{m}_{b \oplus x[i]} = C[\mathsf{m}_{b \oplus x[i]}](x)$. Hence, we can analyze the probability as follows.

- If $x[i] = 0$, the distinguishing probability should be greater than $\frac{1}{2}$ since a good decryptor receives $\mathsf{FE.Enc}(\mathsf{pk}, C[\mathsf{m}_b])$ in its view and correctly guesses b with probability $\frac{1}{2} + \frac{1}{\mathrm{poly}(\lambda)}$.

[5] We omit the security parameter for simplicity in this overview. The same is applied to other cryptographic primitives.

- If $x[i] = 1$, the distinguishing probability should be smaller than $\frac{1}{2}$ since a good decryptor receives $\mathsf{FE.Enc}(\mathsf{pk}, C[\mathsf{m}_{1\oplus b}])$ in its view and outputs the flipped bit $1 \oplus b$ with probability $\frac{1}{2} + \frac{1}{\mathrm{poly}(\lambda)}$.

This means that we can decide $x[i] = 0$ or $x[i] = 1$ by estimating the success probability of a good decryptor that receives $\mathsf{FE.Enc}(\mathsf{pk}, \widetilde{C}[b, \mathsf{m}_0, \mathsf{m}_1, i])$. Thus, we can extract x from good decryptors. This extraction technique is based on the post-quantum watermarking extraction technique by Kitagawa and Nishimaki [KN22]. Hence, the extraction succeeds without collapsing good quantum decryptors $\mathcal{D}_1$ and $\mathcal{D}_2$. See Section 4 for the detail.

There is one subtle issue in the argument above. Since pk depends on x, we need leakage information about x to simulate pk in the reduction. More specifically, let $\mathsf{Leak}(\cdot)$ be a leakage function and the reduction needs $\mathsf{Leak}(x)$ to simulate pk of CPFE. We can consider such a leakage variant of the monogamy of entanglement game, where $\mathcal{A}_0$ receives $|x^\theta\rangle$ and $\mathsf{Leak}(x)$ in the first stage. The variant holds if $|\mathsf{Leak}(x)| = \lambda$ that is short enough compared to $n = |x|$ since we can simply guess $\mathsf{Leak}(x)$ with probability $\frac{1}{2^\lambda}$. Although the bound is degraded to $\frac{2^\lambda}{\exp(n)}$, it is still negligible by setting n appropriately. Hence, we use single-key CPFE with *succinct key* to ensure that $\mathsf{Leak}(x)$ does not have much information about x.

A single-key CPFE scheme has a succinct key if it satisfies the following properties. $\mathsf{FE.Setup}$ consists of two algorithms $(\mathsf{HKG}, \mathsf{Hash})$, computes a hash key $\mathsf{hk} \leftarrow \mathsf{HKG}(1^{|x|})$ and a hash value $h \leftarrow \mathsf{Hash}(\mathsf{hk}, x)$ from x, and outputs a public key $\mathsf{pk} := (\mathsf{hk}, h)$ and a decryption key $\mathsf{sk}_x := x$. The length of h should be the same as the security parameter (no matter how large x is). These properties are crucial for our construction since we consider $\mathsf{Leak}(x) := \mathsf{Hash}(\mathsf{hk}, \cdot)$ and a hash value h does not have much information about x.

We can achieve single-key CPFE with succinct key from hash encryption (HE) [DGHM18], which can be achieved from the LWE. We use HE instead of plain PKE in the well-known single-key FE scheme based on PKE [SS10]. Thanks to the compression property of hash encryption, we can achieve the succinct key property. A decryption key of HE is a pre-image of a hash. Hence, we can use x as a decryption key sk_x.

One-out-of-many unclonable encryption. Georgiou and Zhandry [GZ20] showed that under appropriate definitions, it is possible to transform one-time SDE to unclonable encryption. We show that by using the same transformation, we can transform any one-out-of-many secure SDE to one-out-of-many secure unclonable encryption. By combining this transformation with the above one-out-of-many SDE, we can obtain one-out-of-many secure unclonable encryption based on the LWE assumption.

Copy protection for single-bit output point functions. A point function $f_{y,\mathsf{m}}$ is a function that outputs m on input y and outputs $0^{|\mathsf{m}|}$ otherwise. When we say single-bit output, we set $\mathsf{m} = 1$. When we say multi-bit output, we set m as a multi-bit string sampled from some high min-entropy distribution. We denote the family of single-bit output point functions and multi-bit output point functions as $\mathcal{PF}^1$ and $\mathcal{PF}^{\mathsf{mlt}}$, respectively.

We introduce a simplified security game for copy protection for point functions. It follows the template given in Section 1.1. The first stage adversary is given a copy protected program ρ of a randomly generated point function $f_{y,\mathsf{m}}$. In the second stage, $\mathcal{A}_1$ and $\mathcal{A}_2$ are given a challenge input x sampled from some distribution and try to output $f_{y,\mathsf{m}}(x)$ simultaneously. The copy protection security guarantees that the success probability of the adversary is bounded by the trivial winning probability.

Coladangelo et al. [CMP20] proposed a generic construction of copy protection for $\mathcal{PF}^{\mathsf{mlt}}$ using unclonable encryption.[6] The construction is as follows. To copy protect a multi-bit output point function $f_{y,\mathsf{m}}$, it generates a quantum ciphertext of m of an unclonable encryption scheme UE under the key y, that is $ct \leftarrow \mathsf{UE}.\mathcal{E}nc(y, \mathsf{m})$. For simplicity, we assume that given a key y' and a ciphertext ct' of UE, we can efficiently check whether ct' is generated under the key y' or not. Then, to evaluate this copy protected program with input x, it first checks if x and ct match or not, and if so, just output the decryption result of ct under the key x. We see that the construction satisfies the correctness. Coladangelo et al. also show that if UE satisfies one-wayness-based unclonability, the construction satisfies copy protection security.

In this work, we propose a generic construction of copy protection for $\mathcal{PF}^1$ using unclonable-indistinguishable secure unclonable encryption. The above simple construction by Coladangelo et al. does not work if our goal is copy

[6] Ananth and Kaleoglu [AK21] also proposed a similar construction.

protection for $\mathcal{PF}^1$, even if the underlying unclonable encryption is unclonable-indistinguishable secure. The above construction crucially relies on the fact that m is sampled from high min-entropy distribution in $\mathcal{PF}^{\mathsf{mlt}}$. In fact, if m is fixed as the case of $\mathcal{PF}^1$, the construction is completely insecure under the above condition that we can efficiently check the correspondence between a key and a ciphertext of UE, which is required to achieve correctness.

To fix this issue, our construction uses quantum FHE [Mah18] and obfuscation for C&C programs [WZ17, GKW17], both of which can be realized from the LWE assumption. Roughly speaking, in our construction, the above UE-based copy protected program is encrypted by QFHE. The evaluation of the new copy protected program is done by the homomorphic evaluation of QFHE, and we obtain the evaluation result from the QFHE ciphertext by using decryption circuit of QFHE obfuscated by obfuscation for C&C programs. Our construction works in both standard three adversary style setting explained above and one-out-of-many setting.

Unclonable PE. We also define and construct unclonable PE. Our security definition of unclonable PE is simulation-based. It also can be seen as an extension of simulation-based security notion for (not unclonable) PE defined by Gorbunov et al. [GVW15b].

Our construction of unclonable PE is an extension of ABE-to-PE transformation based on obfuscation for C&C programs proposed in classical cryptography [WZ17, GKW17]. The above construction of copy protection for $\mathcal{PF}^1$ can be extended to copy protection for C&C programs by encrypting a C&C program together with the ciphertext of unclonable encryption into the QFHE ciphertext. At a high level, we show that by replacing obfuscation for C&C programs with this copy protection for C&C programs in the ABE-to-PE transformation, we can obtain unclonable PE. To achieve hiding of policies in PE, the construction crucially uses the security notions of the underlying QFHE and obfuscation for C&C programs. Thus, we do not use the abstraction of copy protection for C&C programs, and present our construction directly using ABE and the building blocks of our copy protection construction. Our construction works in both standard three adversary style setting and one-out-of-many setting.

In the above transformation, we use simulation-based secure ABE instead on indistinguishability-based one. As far as we know, simulation-based secure ABE was not studied before and there is no existing construction. Thus, we construct simulation-based secure ABE by ourselves. The construction is based on indistinguishability-based secure ABE and obfuscation for C&C programs, both of which can be based on the LWE assumption. Interestingly, we can also use our simulation-based secure ABE to convert unclonable encryption into the first unclonable ABE via the standard KEM-DEM framework.

1.5 More on Related Work

Copy-protection for C&C programs. Aaronson proposed candidate constructions of copy-protection for point functions [Aar09]. However, he did not provide reduction-based proofs. Coladangelo, Majenz, and Poremba proposed copy-protection for C&C programs in the QROM and copy-protection for multi-bit output point functions based on one-way-secure unclonable encryption [CMP20]. They also show that we can convert copy-protection for point functions into copy-protection for C&C programs. Ananth and Kaleoglu proposed copy-protected point functions based on indistinguishable-secure unclonable encryption [AK21]. Ananth et al. [AKL+22] proposed indistinguishable-secure unclonable encryption and copy-protection for single-bit output point functions in the QROM. Ananth and La Placa [AL21] show that there exists a class of functions that we cannot achieve copy-protection in the plain model. Ananth and Kaleoglu [AK22] extend the impossibility result by Ananth and La Placa [AL21] and show that there exists a class of functions that we cannot achieve copy-protection in the classical-accessible random oracle model (CAROM). CAROM is a model where both constructions and adversaries can only classically access the random oracle.

Unclonable encryption. Broadbent and Lord [BL20] proposed the notion of unclonable encryption based on the idea by Gottesman [Got03].[7] They considered two security definitions for unclonable encryption. One is one-wayness against cloning attacks (one-way-secure unclonable encryption) and they achieve information-theoretic one-wayness by using BB84 states. The other is indistinguishability against cloning attacks (indistinguishable-secure unclonable encryption). However, they did not achieve it. They constructed indistinguishable-secure unclonable encryption only in a very restricted model by using PRFs. Ananth and Kaleoglu [AK21] proposed a transformation from unclonable

[7]The notion of unclonable encryption by Gottesman is slightly diffrent from the one in this paper. His definition focuses on tamper detection.

encryption to public key unclonable encryption. Ananth et al. [AKL+22] proposed the first indistinguishable-secure unclonable encryption in the QROM.

Unclonable decryption. Georgiou and Zhandry [GZ20] proposed the notion of SDE and show the equivalence between indistinguishable-secure unclonable encryption and their SDE.[8], Coladangelo et al. [CLLZ21] proposed new definitions of SDE and constructed a *public key* SDE scheme that satisfies their definitions from IO and the LWE assumption. Although they needed the strong monogamy of entanglement property conjecture for their constructions, the conjecture was proved without any assumptions by Culf and Vidick [CV22]. It is unclear whether SDE under the definitions by Coladangelo et al. [CLLZ21] is equivalent to unclonable encryption. Liu, Liu, Qian, and Zhandry [LLQZ22] achieved bounded collusion-resistant public key SDE, where adversaries can receive many copy-protected decryption keys, from IO and the LWE assumption. They also consider bounded collusion-resistant copy-protection for PRFs and signatures. Sattath and Wyborski [SW22] also extend SDE to unclonable decryptors, where we can generate multiple copy-protected decryption keys from a classical decryption key. They constructed a secret key unclonable decryptors scheme from copy-protection for balanced binary functions. However, they need IO or a quantum oracle to instantiate copy-protection for balanced binary functions.

2 Preliminaries

Notations and conventions. In this paper, standard math or sans serif font stands for classical algorithms (e.g., C or Gen) and classical variables (e.g., x or pk). Calligraphic font stands for quantum algorithms (e.g., $\mathcal{Gen}$) and calligraphic font and/or the bracket notation for (mixed) quantum states (e.g., q or $|\psi\rangle$). For strings x and y, $x\|y$ denotes the concatenation of x and y. Let $[\ell]$ denote the set of integers $\{1,\cdots,\ell\}$, λ denote a security parameter, and $y := z$ denote that y is set, defined, or substituted by z.

In this paper, for a finite set X and a distribution D, $x \leftarrow X$ denotes selecting an element from X uniformly at random, $x \leftarrow D$ denotes sampling an element x according to D. Let $y \leftarrow \mathsf{A}(x)$ and $y \leftarrow \mathcal{A}(\chi)$ denote assigning to y the output of a probabilistic or deterministic algorithm A and a quantum algorithm $\mathcal{A}$ on an input x and χ, respectively. When we explicitly show that A uses randomness r, we write $y \leftarrow \mathsf{A}(x;r)$. PPT and QPT algorithms stand for probabilistic polynomial-time algorithms and polynomial-time quantum algorithms, respectively. Let negl denote a negligible function. Let $\overset{\mathsf{c}}{\approx}$ denote computational indistinguishability.

2.1 Quantum information

We review several quantum information concepts.

Basics. Let $\mathcal{H}$ be a finite-dimensional complex Hilbert space. A (pure) quantum state is a vector $|\psi\rangle \in \mathcal{H}$. Let $\mathcal{S}(\mathcal{H})$ be the space of Hermitian operators on $\mathcal{H}$. A density matrix is a Hermitian operator $\chi \in \mathcal{S}(\mathcal{H})$ with $\mathrm{Tr}(\chi) = 1$, which is a probabilistic mixture of pure states. A quantum state over $\mathcal{H} = \mathbb{C}^2$ is called qubit, which can be represented by the linear combination of the standard basis $\{|0\rangle, |1\rangle\}$. More generally, a quantum system over $(\mathbb{C}^2)^{\otimes n}$ is called an n-qubit quantum system for $n \in \mathbb{N} \setminus \{0\}$.

A Hilbert space is divided into registers $\mathcal{H} = \mathcal{H}^{\mathsf{R}_1} \otimes \mathcal{H}^{\mathsf{R}_2} \otimes \cdots \otimes \mathcal{H}^{\mathsf{R}_n}$. We sometimes write χ^{R_i} to emphasize that the operator χ acts on register $\mathcal{H}^{\mathsf{R}_i}$.[9] When we apply χ^{R_1} to registers $\mathcal{H}^{\mathsf{R}_1}$ and $\mathcal{H}^{\mathsf{R}_2}$, χ^{R_1} is identified with $\chi^{\mathsf{R}_1} \otimes \boldsymbol{I}^{\mathsf{R}_2}$.

A unitary operation is represented by a complex matrix $\boldsymbol{U}$ such that $\boldsymbol{U}\boldsymbol{U}^\dagger = \boldsymbol{I}$. The operation $\boldsymbol{U}$ transforms $|\psi\rangle$ and χ into $\boldsymbol{U}|\psi\rangle$ and $\boldsymbol{U}\chi\boldsymbol{U}^\dagger$, respectively. A projector $\boldsymbol{P}$ is a Hermitian operator ($\boldsymbol{P}^\dagger = \boldsymbol{P}$) such that $\boldsymbol{P}^2 = \boldsymbol{P}$.

For a quantum state χ over two registers $\mathcal{H}^{\mathsf{R}_1}$ and $\mathcal{H}^{\mathsf{R}_2}$, we denote the state in $\mathcal{H}^{\mathsf{R}_1}$ as $\chi[\mathsf{R}_1]$, where $\chi[\mathsf{R}_1] = \mathrm{Tr}_2[\chi]$ is a partial trace of χ (trace out R_2).

[8] Selectively secure secret key SDE in the setting of honestly generated keys. See [GZ20] for the detail.

[9] The superscript parts are gray colored.

Measurement Implementation We review some concepts on quantum measurements.

Definition 2.1 (Projective Implementation [Zha20]). *Let:*

- $\mathcal{D}$ *be a finite set of distributions over an index set* $\mathcal{I}$.
- $\mathcal{P} = \{\boldsymbol{P}_i\}_{i\in\mathcal{I}}$ *be a POVM*
- $\mathcal{E} = \{\boldsymbol{E}_D\}_{D\in\mathcal{D}}$ *be a projective measurement with index set* $\mathcal{D}$.

We consider the following measurement procedure.

1. *Measure under the projective measurement* $\mathcal{E}$ *and obtain a distribution* D.
2. *Output a random sample from the distribution* D.

We say $\mathcal{E}$ *is the projective implementation of* $\mathcal{P}$*, denoted by* $\mathsf{ProjImp}(\mathcal{P})$*, if the measurement process above is equivalent to* $\mathcal{P}$.

Theorem 2.2 ([Zha20, Lemma 1]). *Any binary outcome POVM* $\mathcal{P} = (\boldsymbol{P}, \boldsymbol{I} - \boldsymbol{P})$ *has a unique projective implementation* $\mathsf{ProjImp}(\mathcal{P})$.

Definition 2.3 (Mixture of Projetive Measurement [Zha20]). *Let* $D : \mathcal{R} \to \mathcal{I}$ *where* $\mathcal{R}$ *and* $\mathcal{I}$ *are some sets. Let* $\{(\boldsymbol{P}_i, \boldsymbol{Q}_i)\}_{\in\mathcal{I}}$ *be a collection of binary projective measurement. The mixture of projective measurements associated to* $\mathcal{R}$, $\mathcal{I}$, D, *and* $\{(\boldsymbol{P}_i, \boldsymbol{Q}_i)\}_{\in\mathcal{I}}$ *is the binary POVM* $\mathcal{P}_D = (\boldsymbol{P}_D, \boldsymbol{Q}_D)$ *defined as follows*

$$\boldsymbol{P}_D = \sum_{i\in\mathcal{I}} \Pr[i \leftarrow D(R)]\boldsymbol{P}_i \qquad\qquad \boldsymbol{Q}_D = \sum_{i\in\mathcal{I}} \Pr[i \leftarrow D(R)]\boldsymbol{Q}_i,$$

where R *is uniformly distributed in* $\mathcal{R}$.

Definition 2.4 (Shift Distance). *For two distributions* D_0, D_1*, the shift distance with parameter* ϵ*, denoted by* $\Delta^{\epsilon}_{\mathsf{Shift}}(D_0, D_1)$*, is the smallest quantity* δ *such that for all* $x \in \mathbb{R}$*:*

$$\Pr[D_0 \le x] \le \Pr[D_1 \le x + \epsilon] + \delta, \qquad \Pr[D_0 \ge x] \le \Pr[D_1 \ge x - \epsilon] + \delta,$$
$$\Pr[D_1 \le x] \le \Pr[D_0 \le x + \epsilon] + \delta, \qquad \Pr[D_1 \ge x] \le \Pr[D_0 \ge x - \epsilon] + \delta.$$

For two real-valued measurements $\mathcal{M}$ *and* $\mathcal{N}$ *over the same quantum system, the shift distance between* $\mathcal{M}$ *and* $\mathcal{N}$ *with parameter* ϵ *is*

$$\Delta^{\epsilon}_{\mathsf{Shift}}(\mathcal{M}, \mathcal{N}) := \sup_{|\psi\rangle} \Delta^{\epsilon}_{\mathsf{Shift}}(\mathcal{M}(|\psi\rangle), \mathcal{N}(|\psi\rangle)).$$

Theorem 2.5 ([Zha20, KN22]). *Let* D *be any probability distribution and* $\mathcal{P} = \{(\Pi_i, \boldsymbol{I} - \Pi_i)\}_i$ *be a collection of binary outcome projective measurements. For any* $0 < \epsilon, \delta < 1$*, there exists an algorithm of measurement* $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},\mathcal{D}}$ *that satisfies the following.*

- $\Delta^{\epsilon}_{\mathsf{Shift}}(\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D}, \mathsf{ProjImp}(\mathcal{P}_D)) \le \delta$.
- $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D}$ *is* (ϵ, δ)*-almost projective in the following sense. For any quantum state* $|\psi\rangle$*, we apply* $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D}$ *twice in a row to* $|\psi\rangle$ *and obtain measurement outcomes* x *and* y*, respectively. Then,* $\Pr[|x - y| \le \epsilon] \ge 1 - \delta$.
- $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D}$ *is* (ϵ, δ)*-reverse almost projective in the following sense. For any quantum state* $|\psi\rangle$*, we apply* $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D}$ *and* $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P}^{\mathtt{rev}},D}$ *in a row to* $|\psi\rangle$ *and obtain measurement outcomes* x *and* y*, respectively, where* $\mathcal{P}^{\mathtt{rev}} = \{(\boldsymbol{I} - \Pi_i, \Pi_i)\}_i$*. Then,* $\Pr[|(1 - x) - y| \le \epsilon] \ge 1 - \delta$.
- *The expected running time of* $\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D}$ *is* $T_{\mathcal{P},D} \cdot \mathrm{poly}(1/\epsilon, \log(1/\delta))$ *where* $T_{\mathcal{P},D}$ *is the combined running time of* D*, the procedure mapping* $i \to (\boldsymbol{P}_i, \boldsymbol{I} - \boldsymbol{P}_i)$*, and the running time of measurement* $(\boldsymbol{P}_i, \boldsymbol{I} - \boldsymbol{P}_i)$.

Theorem 2.6 ([Zha20, Corollary 1]). *Let* q *be an efficiently constructible, potentially mixed state, and* D_0, D_1 *efficiently sampleable distributions. If* D_0 *and* D_1 *are computationally indistinguishable, for any inverse polynomial* ϵ *and any function* δ*, we have* $\Delta^{3\epsilon}_{\mathsf{Shift}}(\mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D_0}, \mathcal{API}^{\epsilon,\delta}_{\mathcal{P},D_1}) \le 2\delta + \mathsf{negl}(\lambda)$.

Quantum program with classical inputs and outputs We formalize quantum programs whose inputs and outputs are always classical strings.

Definition 2.7 (Quantum Program with Classical Inputs and Outputs [ALL+21]). *A quantum program with classical inputs is a pair of quantum state q and unitaries $\{\boldsymbol{U}_x\}_{x\in[N]}$ where $[N]$ is the domain, such that the state of the program evaluated on input x is equal to $\boldsymbol{U}_x q \boldsymbol{U}_x^\dagger$. We measure the first register of $\boldsymbol{U}_x q \boldsymbol{U}_x^\dagger$ to obtain an output. We say that $\{\boldsymbol{U}_x\}_{x\in[N]}$ has a compact classical description $\boldsymbol{U}$ when applying $\boldsymbol{U}_x$ can be efficiently computed given $\boldsymbol{U}$ and x.*

2.2 Standard Cryptographic Tools

Commitment. We introduce the notion of injective commitment with equivocal mode. This is an extension of injective commitment introduced by Cohen et al. [CHN+18].

Definition 2.8 (Injective Commitment with Equivocal Mode). *An injective commitment scheme* Com *with equivocal mode for the message space $\mathcal{M}$ and random coin space $\mathcal{R}$ is a tuple of four algorithms* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{EqSetup}, \mathsf{Open})$.

- *The setup algorithm* Setup *takes as input a security parameter 1^λ, and outputs a commitment key* ck.
- *The commitment algorithm* Commit *takes as input the commitment key* ck, *a message $m \in \mathcal{M}$, and a random coin $r \in \mathcal{R}$, and outputs a commitment* com.
- *The equivocation setup algorithms* $\mathsf{EqSetup}$ *takes as input a security parameter 1^λ, and outputs a commitment key* ck^*, *a commitment* com^*, *and a trapdoor* td.
- *The open algorithm* Open *takes as input the trapdoor* td, *a message $m \in \mathcal{M}$, and a commitment* com^*, *and outputs a random coin $r^* \in \mathcal{R}$.*

We say that injective commitment with equivocal mode is secure if it satisfies the following two properties.

Injectivity: *We require that*

$$\Pr\left[\exists (m_1, r_1) \neq (m_2, r_2) \ \ s.t. \ \ \mathsf{Commit}(\mathsf{ck}, m_1; r_1) = \mathsf{Commit}(\mathsf{ck}, m_2, r_2) \mid \mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda)\right] = \mathsf{negl}(\lambda).$$

Trapdoor Equivocality: *For any message $m \in \mathcal{M}$, we have*

$$(\mathsf{ck}, \mathsf{com}, r) \overset{\mathsf{c}}{\approx} (\mathsf{ck}^*, \mathsf{com}^*, r^*),$$

where $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda)$, $r \leftarrow \mathcal{R}$, $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{ck}, m; r)$, $(\mathsf{ck}^*, \mathsf{com}^*, \mathsf{td}) \leftarrow \mathsf{EqSetup}(1^\lambda)$, *and* $r^* \leftarrow \mathsf{Open}(\mathsf{td}, m, \mathsf{com}^*)$.

We do not explicitly require a hiding property since we do not need it in this work.

Theorem 2.9. *If the LWE assumption holds, there exists a secure injective commitment with equivocal mode.*

We can obtain Theorem 2.9 from the construction of injective commitment by Kitagawa and Nishimaki [KN22] since it is Naor's commitment scheme [Nao91] and it is well known that Naor's commitment has a trapdoor equivocal mode. See Appendix C for the construction.

Ciphertext-policy attribute-based encryption. We define ciphertext-policy attribute-base encryption (CP-ABE) and adaptive-indistinguishability (AD-IND security) and adaptive-simulation security (AD-SIM security) for it.

Definition 2.10 (CP-ABE (Syntax)). *A CP-ABE scheme is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *with plaintext space $\mathcal{M}$.*

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** *The setup algorithm takes as input the security parameter 1^λ and outputs a key pair* $(\mathsf{mpk}, \mathsf{msk})$.

$\mathsf{KG}(\mathsf{msk}, x) \to \mathsf{sk}_x$**:** *The key generation algorithm takes as input the master secret key* msk *and an attribute* x*, and outputs a decryption key* sk_x*.*

$\mathsf{Enc}(\mathsf{mpk}, C, \mathsf{m}) \to \mathsf{ct}$**:** *The encryption algorithm takes as input* mpk*, policy* C*, and a plaintext* $\mathsf{m} \in \mathcal{M}$*, and outputs a ciphertext* ct*.*

$\mathsf{Dec}(\mathsf{sk}_x, \mathsf{ct}) \to \mathsf{m}'$ **or** $\bot$**:** *The decryption algorithm takes as input* sk_x *and* ct *and outputs a plaintext* m' *or* $\bot$*.*

Decryption Correctness: *There exists a negligible function* negl *such that for any* $\mathsf{m} \in \mathcal{M}$*,* C*, and* x *such that* $C(x) = 1$*, we have*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_x, \mathsf{ct}) = \mathsf{m} \;\middle|\; \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, C, \mathsf{m}) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

Definition 2.11 (AD-IND Security for CP-ABE). *Let* $\mathsf{CPABE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a CP-ABE scheme. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{ad\text{-}ind}}_{\mathsf{CPABE},\mathcal{A}}(\lambda, \mathsf{coin})$ *for a QPT adversary* $\mathcal{A}$*.*

1. *The challenger computes* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{KG}(1^\lambda)$ *and sends* mpk *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *can get access to the following oracle.*

   $O_{\mathsf{KG},1}(x)$**:** *Given* x*, it returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$*.*
3. $\mathcal{A}$ *sends* C *and* $(\mathsf{m}_0, \mathsf{m}_1) \in \mathcal{M}^2$ *to the challenger, where* C *satisfies* $C(x) = 0$ *for all* x *queried by* $\mathcal{A}$ *in the previous step. The challenger returns* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, C, m_{\mathsf{coin}})$ *to* $\mathcal{A}$*.*
4. $\mathcal{A}$ *can get access to the following oracle.*

   $O_{\mathsf{KG},2}(x)$**:** *Given* x*, if* $C(x) = 0$*, it returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$*. If* $C(x) = 1$*, it returns* $\bot$*.*
5. $\mathcal{A}$ *outputs* $\mathsf{coin}' \in \{0, 1\}$*.*

We say that CPABE *is AD-IND secure if for any QPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{ad\text{-}ind}}_{\mathsf{CPABE},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{ad\text{-}ind}}_{\mathsf{CPABE},\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{ad\text{-}ind}}_{\mathsf{CPABE},\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

Remark 2.12. We can consider the selective variant of Definition 2.11, where $\mathcal{A}$ declares C at the beginning of the games. We can also consider the selective variant of Definition 2.14 introduced below. We denote these notions by SEL-IND and SEL-SIM, respectively.

Theorem 2.13 ([GVW15a, BGG+14]). *If the LWE assumption holds, there exists SEL-IND secure CP-ABE for all boolean circuits. In addition, if the LWE assumption holds against sub-exponential time algorithms, there exists ADA-IND secure CP-ABE for all boolean circuits.*

We define AD-SIM security for CP-ABE, which is a natural adaptation of AD-SIM security for functional encryption [GVW12] to CP-ABE.

Definition 2.14 (AD-SIM Security for CP-ABE). *Let* $\mathsf{CPABE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a CP-ABE scheme. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{S}im,\mathcal{A}}(\lambda, \mathsf{coin})$ *for a QPT simulator* $\mathcal{S}im = (\mathcal{S}im\mathcal{E}nc, \mathcal{S}im\mathcal{K}\mathcal{G})$ *and a QPT adversary* $\mathcal{A}$*.*

1. *The challenger computes* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{KG}(1^\lambda)$ *and sends* mpk *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *can get access to the following oracle.*

   $O_{\mathsf{KG},1}(x)$**:** *Given* x*, it returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$*.*

3. *$\mathcal{A}$ sends C and $m \in \mathcal{M}$ to the challenger, where C satisfies $C(x) = 0$ for all x queried by $\mathcal{A}$ in the previous step. The challenger does the following.*
   - *If* $\mathsf{coin} = 0$, *the challenger generates* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, C, m)$ *and returns* ct *to* $\mathcal{A}$.
   - *If* $\mathsf{coin} = 1$, *the challenger generates* $(\mathsf{ct}, \mathsf{st}) \leftarrow \mathcal{SimEnc}(\mathsf{mpk}, C)$ *and returns* ct *to* $\mathcal{A}$.
4. *$\mathcal{A}$ can get access to the following oracle.*

   $O_{\mathsf{KG},2}(x)$**:** *Given x, if $C(x) = 0$, it returns $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$. If $C(x) = 1$, it does the following.*
   - *If* $\mathsf{coin} = 0$, *the challenger returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$ *to* $\mathcal{A}$.
   - *If* $\mathsf{coin} = 1$, *the challenger returns* $\mathsf{sk}_x \leftarrow \mathcal{SimKG}(\mathsf{msk}, \mathsf{st}, x, m)$ *to* $\mathcal{A}$.
5. *$\mathcal{A}$ outputs* $\mathsf{coin}' \in \{0,1\}$.

We say that CPABE *is AD-SIM secure if there exists $\mathcal{Sim}$ such that for any QPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda,0)=1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda,1)=1\right]\right| \leq \mathsf{negl}(\lambda).$$

AD-SIM security has been widely studied for functional encryption, but as far as we know, there was no previous work studied it for CP-ABE. In Appendix A, we show how to transform any AD-IND secure CP-ABE scheme into AD-SIM secure one by using tools implied by the LWE assumption.

Compute-and-compare obfuscation. We define a class of circuits called compute-and-compare circuits for which we study copy protection and secure software leasing in this work.

Definition 2.15 (Compute-and-Compare Circuits). *A compute-and-compare circuit* $\mathbf{CC}[P, \mathsf{lock}, \mathsf{m}]$ *is of the form*

$$\mathbf{CC}[P, \mathsf{lock}, \mathsf{m}](x) \begin{cases} \mathsf{m} & (P(x) = \mathsf{lock}) \\ 0 & (\textit{otherwise}) \end{cases},$$

where P is a circuit, lock *is a string called lock value, and* m *is a message.*

We introduce the definition of compute-and-compare obfuscation. We assume that a program P has an associated set of parameters pp_P (input size, output size, circuit size) which we do not need to hide.

Definition 2.16 (Compute-and-Compare Obfuscation). *A PPT algorithm* $\mathsf{CC.Obf}$ *is an obfuscator for the family of distributions $D = \{D_\lambda\}$ if the following holds:*

Functionality Preserving: *There exists a negligible function* negl *such that for all program P, all lock value* lock*, and all message* m*, it holds that*

$$\Pr\left[\forall x, \tilde{P}(x) = \mathbf{CC}[P, \mathsf{lock}, \mathsf{m}](x) \mid \tilde{P} \leftarrow \mathsf{CC.Obf}(1^\lambda, P, \mathsf{lock}, \mathsf{m})\right] = 1 - \mathsf{negl}(\lambda).$$

Distributional Indistinguishability: *There exists an efficient simulator* Sim *such that for all message* m*, we have*

$$(\mathsf{CC.Obf}(1^\lambda, P, \mathsf{lock}, \mathsf{m}), \mathit{aux}) \stackrel{\mathsf{c}}{\approx} (\mathsf{Sim}(1^\lambda, \mathsf{pp}_P, |\mathsf{m}|), \mathit{aux}),$$

where $(P, \mathsf{lock}, \mathit{aux}) \leftarrow D_\lambda$.

Theorem 2.17 ([GKW17, WZ17]). *If the LWE assumption holds, there exists compute-and-compare obfuscation for all families of distributions $D = \{D_\lambda\}$, where each D_λ outputs uniformly random lock value* lock *independent of P and aux.*

2.3 Quantum Cryptographic Tools

Unclonable encryption. We introduce the definition of secret key unclonable encryption (SKUE) [BL20] and one-time indistinguishability for it.

Definition 2.18 (SKUE (Syntax)). *A SKUE scheme with the message space $\mathcal{M}$ is a tuple of quantum algorithms $(\mathsf{KG}, \mathcal{Enc}, \mathcal{Dec})$.*

$\mathsf{KG}(1^\lambda) \to \mathsf{uk}$**:** *The key generation algorithm takes as input the security parameter 1^λ and outputs a key* uk.

$\mathcal{Enc}(\mathsf{uk}, m) \to ct$**:** *The encryption algorithm takes as input* uk *and a plaintext* $\mathsf{m} \in \mathcal{M}$ *and outputs a ciphertext ct.*

$\mathcal{Dec}(\mathsf{uk}, ct) \to \mathsf{m}'$**:** *The decryption algorithm takes as input* uk *and ct and outputs a plaintext* $\mathsf{m}' \in \mathcal{M}$ *or* $\bot$.

Decryption correctness: *For any* $\mathsf{m} \in \mathcal{M}$*, it holds that*

$$\Pr\left[\mathcal{Dec}(\mathsf{uk}, ct) = \mathsf{m} \;\middle|\; \begin{array}{l} \mathsf{uk} \leftarrow \mathsf{KG}(1^\lambda) \\ ct \leftarrow \mathcal{Enc}(\mathsf{uk}, \mathsf{m}) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

Definition 2.19 (One-Time Unclonable-Indistinguishable Security for SKUE). *Let $\mathsf{UE} = (\mathcal{Enc}, \mathcal{Dec})$ be an SKUE scheme with the key space $\mathcal{K}$ and the message space $\mathcal{M}$. We consider the following security experiment $\mathsf{Exp}^{\mathsf{ot\text{-}ind\text{-}clone}}_{\mathsf{UE},\mathcal{A}}(\lambda)$, where $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$.*

1. *$\mathcal{A}_0$ sends (m_0, m_1) to the challenger.*
2. *The challenger generates* $\mathsf{coin} \leftarrow \{0,1\}$, $\mathsf{uk} \leftarrow \mathsf{KG}(1^\lambda)$*, and* $ct \leftarrow \mathcal{Enc}(\mathsf{uk}, m_{\mathsf{coin}})$*, and sends ct to $\mathcal{A}_0$.*
3. *$\mathcal{A}_0$ creates a bipartite state q over registers R_1 and R_2. $\mathcal{A}$ sends $q[\mathsf{R}_1]$ and $q[\mathsf{R}_2]$ to $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.*
4. *The challenger sends* uk *to $\mathcal{A}_1$ and $\mathcal{A}_2$. $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively output* coin'_1 *and* coin'_2*. If* $\mathsf{coin}'_i = \mathsf{coin}$ *for* $i \in \{1,2\}$*, the challenger outputs* 1*, otherwise outputs* 0*.*

We say that UE *is one-time unclonable-indistinguishable secure SKUE scheme if for any QPT $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{ot\text{-}ind\text{-}clone}}_{\mathsf{UE},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{ot\text{-}ind\text{-}clone}}_{\mathsf{UE},\mathcal{A}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Single-decryptor encryption. We review the definition of single-decryptor encryption (SDE). We consider a one-time secret key variant of SDE by Coladangelo et al. [CLLZ21] in this work.

Definition 2.20 (Secret Key SDE (Syntax)). *A secret key SDE scheme* SDE *is a tuple of quantum algorithms $(\mathcal{KG}, \mathsf{Enc}, \mathcal{Dec})$ with plaintext space $\mathcal{M}$.*

$\mathcal{KG}(1^\lambda) \to (\mathsf{ek}, d\mathcal{k})$**:** *The key generation algorithm takes as input the security parameter 1^λ and outputs an encryption key* ek *and a quantum decryption key $d\mathcal{k}$.*

$\mathsf{Enc}(\mathsf{ek}, m) \to \mathsf{ct}$**:** *The encryption algorithm takes as input* ek *and a plaintext $m \in \mathcal{M}$ and outputs a ciphertext* ct.

$\mathcal{Dec}(d\mathcal{k}, \mathsf{ct}) \to m'$**:** *The decryption algorithm takes as input $d\mathcal{k}$ and* ct *and outputs a plaintext $m' \in \mathcal{M}$ or $\bot$.*

Decryption correctness: *There exists a negligible function* negl *such that for any $m \in \mathcal{M}$,*

$$\Pr\left[\mathcal{Dec}(d\mathcal{k}, \mathsf{ct}) = m \;\middle|\; \begin{array}{l} (\mathsf{ek}, d\mathcal{k}) \leftarrow \mathcal{KG}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, m) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

Definition 2.21 (One-Time Strong Anti-Piracy Security for Secret Key SDE). *Let $\gamma \in [0,1]$. Let $\mathsf{SDE} = (\mathcal{KG}, \mathsf{Enc}, \mathcal{Dec})$ be a secret key SDE scheme. We consider the one-time strong anti-piracy game $\mathsf{Exp}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, \gamma)$ between the challenger and an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ below.*

1. *The challenger generates* $(\mathsf{ek}, d\mathcal{K}) \leftarrow \mathcal{KG}(1^\lambda)$ *and sends* $d\mathcal{K}$ *to* $\mathcal{A}_0$.
2. $\mathcal{A}_0$ *creates a bipartite state* q *over registers* R_1 *and* R_2. $\mathcal{A}$ *sends* (m_0, m_1), $q[\mathsf{R}_1]$, *and* $q[\mathsf{R}_2]$ *to the challenger,* $\mathcal{A}_1$, *and* $\mathcal{A}_2$, *respectively.*
3. $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *respectively output* $\mathcal{D}_1 = (\rho[\mathsf{R}_1], \boldsymbol{U}_1)$ *and* $\mathcal{D}_2 = (\rho[\mathsf{R}_2], \boldsymbol{U}_2)$.
4. *For* $\alpha \in [2]$, *let* $\boldsymbol{P}_\alpha = (\boldsymbol{P}_{\alpha,b,\mathsf{ct}}, \boldsymbol{I} - \boldsymbol{P}_{\alpha,b,\mathsf{ct}})_{b,\mathsf{ct}}$ *be a collection of binary projective measurements, where*

$$\boldsymbol{P}_{\alpha,b,\mathsf{ct}} = \boldsymbol{U}_{\alpha,\mathsf{ct}}^\dagger |b\rangle \langle b| \boldsymbol{U}_{\alpha,\mathsf{ct}}.$$

We also define D *as the distribution that generates* $b \leftarrow \{0,1\}$ *and* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, m_b)$ *and outputs* (b, ct). *Also, for* $\alpha \in [2]$, *we denote the mixture of* $\boldsymbol{P}_\alpha$ *with respect to* D *as* $\boldsymbol{P}_{\alpha,D}$. *Then, for every* $\alpha \in [2]$, *the challenger applies* $\mathsf{ProjImp}(\boldsymbol{P}_{\alpha,D})$ *to* $\rho[\mathsf{R}_\alpha]$ *and obtains* p_α. *If* $p_\alpha > \frac{1}{2} + \gamma$ *for every* $\alpha \in [2]$, *the challenger outputs* 1. *Otherwise, the challenger outputs* 0.

We say that SDE *is one-time strong anti-piracy secure if for any* $\gamma \in [0,1]$ *and QPT adversary* $\mathcal{A}$, *it satisfies that*

$$\mathsf{Adv}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma) := \Pr\left[\mathsf{Exp}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma) = 1\right] = \mathsf{negl}(\lambda).$$

Remark 2.22. Readers might think the meaning of "one-time" is unclear in Definition 2.21. Here, "one-time" means that $\mathcal{A}_0$ cannot access an encryption oracle that returns a ciphertext for a query m. This naming might sound strange since $\mathcal{A}_0$ does not receive any ciphertext. However, we stick to this naming for correspondence with one-time unclonable-indistinguishable security of unclonable encryption in Definition 2.19.

Remark 2.23 (On the issue in indistinguishability-based definitions). There are two indistinguishability-based security notions for SDE. The first one is defined by Georgiou and Zhandry (denoted by GZ) [GZ20] and the second one is defined by Coladangelo et al. (denoted by CLLZ)[10] [CLLZ21]. Both of them are defined using a security game similar to that in Definition 2.21, except that $\mathcal{A}_1$ and $\mathcal{A}_2$ are given the challenge ciphertexts and required to guess the challenge bits. In the GZ definition, $\mathcal{A}_1$ and $\mathcal{A}_2$ receive the same ciphertext $\mathsf{Enc}(\mathsf{sk}, m_{\mathsf{coin}})$ for the single challenge bit. However, in the CLLZ definition, $\mathcal{A}_1$ and $\mathcal{A}_2$ receive different ciphertexts $\mathsf{Enc}(\mathsf{sk}, m_{\mathsf{coin}_1})$ and $\mathsf{Enc}(\mathsf{sk}, m_{\mathsf{coin}_2})$, respectively, where coin_1 and coin_2 are independent challenge bits. Currently, the relationship between these two security notions for SDE remains elusive.[11] The GZ definition is known to imply unclonable-indistinguishable secure unclonable encryption, but the CLLZ definition is not. Also, strong anti-piracy security defined in Definition 2.21 implies the CLLZ definition but not the GZ definition.

Quantum fully homomorphic encryption. We introduce quantum fully homomorphic encryption (QFHE) with classical ciphretexts.

Definition 2.24 (QFHE with Classical Ciphertexts [Mah18, Bra18]). *A QFHE scheme with classical ciphertext is a tuple of algorithms* $(\mathsf{KG}, \mathsf{Enc}, \mathcal{E}val, \mathsf{Dec})$.

$\mathsf{KG}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** *The key generation algorithm takes as input the security parameter* 1^λ *and outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$**:** *The encryption algorithm takes as input a public key* pk *and a plaintext* m, *and outputs a ciphertext* ct. *Without loss of generality, we can assume that a ciphertext includes the public key* pk.

$\mathcal{E}val(C, \rho, \mathsf{ct}_1, \cdots, \mathsf{ct}_n) \to \mathsf{evct}$**:** *The evaluation algorithm takes as input a quantum circuit* C *with classical outputs, quantum state* ρ, *and ciphertexts* $\mathsf{ct}_1, \cdots, \mathsf{ct}_n$, *and outputs a classical ciphertext* evct.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m'$**:** *The decryption algorithm takes as input a secret key* sk *and a ciphertext* ct, *and outputs a plaintext* m.

[10] They call "CPA-style anti-piracy security".

[11] Ananth et al. [AKL23] show the relationship between *one-wayness-based* security with the same ciphertext and one with the different ciphertexts.

Decryption Correctness: *Let* $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$. *Let* $(m_1,\cdots,m_n)$ *be any* n *messages. For any* $i \in [n]$, *let* $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pk},m_i)$ *for every* $i \in [n]$. *Let* C *be a quantum circuit that takes a quantum state and* n *classical input,* ρ *a quantum state, and* $\mathsf{evct} \leftarrow \mathcal{E}val(C,\rho,\mathsf{ct}_1,\cdots,\mathsf{ct}_n)$. *Then, we have* $\mathsf{Dec}(\mathsf{sk},\mathsf{evct}) = C(\rho,m_1,\cdots,m_n)$.

Semantic Security: *For any two messages of equal length* m_0,m_1, *we have*

$$(\mathsf{pk},\mathsf{Enc}(\mathsf{pk},m_0)) \stackrel{\mathsf{c}}{\approx} (\mathsf{pk},\mathsf{Enc}(\mathsf{pk},m_1)),$$

where $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda)$.

The existing QFHE schemes [Mah18, Bra18] can be seen as QFHE with classical ciphertexts, since they have a property that if the encrypted plaintext is classical, we can make the ciphertext classical. Thus, the following theorem holds.

Theorem 2.25 ([Mah18, Bra18]). *If the LWE assumption holds, there exists QFHE with classical ciphertexts.*

QFHE with classical ciphertexts was previously used in the context of impossibility on copy protection [AL21] and quantum obfuscation [ABDS21]. For the detailed explanation for how to use the existing QFHE schemes as QFHE with classical ciphertexts, please refer to [ABDS21].

3 One-out-of-Many Unclonable Security

This section introduces new unclonable security notions that we call one-out-of-many unclonable security. The definition is roughly as follows. The one-out-of-many unclonable security game is an indistinguishability-style game played by a tuple of $n+1$ adversaries $(\mathcal{A}_0,\mathcal{A}_1,\cdots,\mathcal{A}_n)$, where $2 \leq n$ is arbitrary. At the first stage of the game, $\mathcal{A}_0$ is given a single quantum object (such as ciphertext in unclonable encryption), generates possibly entangled n-partite states $\rho_1,\ldots,\rho_n$, and sends ρ_k to $\mathcal{A}_k$ for $k \in \{1,\ldots,n\}$. At the second stage, the challenger selects one of $(\mathcal{A}_1,\ldots,\mathcal{A}_n)$ by a random $\alpha \leftarrow \{1,\ldots,n\}$ and sends additional information (such as a secret key in unclonable encryption) *only to* $\mathcal{A}_\alpha$, and only $\mathcal{A}_\alpha$ tries to guess the challenge bit $\mathsf{coin} \in \{0,1\}$. The one-out-of-many unclonable security guarantees that the adversary cannot win this game with a probability significantly better than the trivial winning probability $\frac{1}{2}+\frac{1}{2n}$.[12] .

The one-out-of-many unclonable security notion guarantees that no adversary can generate n copies with a probability significantly better than $\frac{1}{n}$ for any n. This is because an adversary who can generate n copies with probability $\frac{1}{n}+\delta$ can win the one-out-of-many game with probability at least $(\frac{1}{n}+\delta)\cdot 1+(1-\frac{1}{n}-\delta)\cdot\frac{1}{2} = \frac{1}{2}+\frac{1}{2n}+\frac{\delta}{2}$, which violates to the one-out-of-many unclonable security. The one-out-of-many unclonable security notion does not rule out a copying process that can generate n copies with probability $\frac{1}{n}$. However, it guarantees that such a process must completely destroy the original object with probability $1-\frac{1}{n}$. In fact, it guarantees that the expected number of successful copies generated by any copying adversary is at most 1.

Although one-out-of-many unclonable security looks weaker than existing unclonable security, it still seems useful in some applications. For example, suppose we publish many quantum objects, say ℓ objects. Then, the one-out-of-many security guarantees that no matter what copying attacks are applied to those objects, there are expected to be only ℓ objects on average in this world.

Below, we define one-out-of-many one-time unclonable-indistinguishable security for SKUE and one-out-of-many one-time anti-piracy for secret key SDE. We prove that one-out-of-many one-time unclonable-indistinguishable security for SKUE implies one-time IND-CPA security. We also prove that one-time strong anti-piracy for secret key SDE implies one-out-of-many one-time anti-piracy for secret key SDE. Then, we show that we can transform secret key SDE with one-out-of-many one-time anti-piracy into one-out-of-many one-time unclonable-indistinguishable secure SKUE.

In Section 5, we define one-out-of-many copy protection security for single-bit output point functions. In Section 6, we define simulation-based security for unclonable PE and introduce its one-out-of-many variant.

[12] Suppose $\mathcal{A}_0$ forwards the given quantum state to $\mathcal{A}_1$ and nothing to $(\mathcal{A}_2,\ldots,\mathcal{A}_n)$. If $\alpha=1$ is chosen, the adversaries win with probability 1 because the additional information, together with the original quantum object, can be used to compute the challenge bit coin correctly. If one of $(\mathcal{A}_2,\ldots,\mathcal{A}_n)$ is chosen, the adversaries win with probability $\frac{1}{2}$ by random guess. Hence, the advantage is $\frac{1}{n}\cdot 1+\frac{n-1}{n}\cdot\frac{1}{2} = \frac{1}{2}+\frac{1}{2n}$, which we consider as the trivial advantage.

3.1 One-out-of-Many Security Notions for SKUE and Secret Key SDE

Definition 3.1 (One-out-of-Many One-Time Unclonable-Indistinguishability for SKUE). *Let* $\mathsf{UE} = (\mathsf{KG}, \mathcal{E}nc, \mathcal{D}ec)$ *be an SKUE scheme with the message space* $\mathcal{M}$*. We consider one-out-of-many one-time unclonable-indistinguishability game* $\mathsf{Exp}^{\mathsf{om\text{-}ot\text{-}clone\text{-}ind}}_{\mathsf{UE},\mathcal{A}}(\lambda, n)$ *between the challenger and an adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$ *below.*

1. $\mathcal{A}_0$ *sends* $(\mathsf{m}_0, \mathsf{m}_1)$ *to the challenger.*
2. *The challenger generates* $\mathsf{coin} \leftarrow \{0,1\}$, $\mathsf{uk} \leftarrow \mathsf{KG}(1^\lambda)$*, and* $ct \leftarrow \mathcal{E}nc(\mathsf{uk}, \mathsf{m}_{\mathsf{coin}})$*, and sends* ct *to* $\mathcal{A}_0$*.*
3. $\mathcal{A}_0$ *creates a quantum state* q *over* n *registers* $\mathsf{R}_1, \cdots, \mathsf{R}_n$*.* $\mathcal{A}_0$ *sends* $q[\mathsf{R}_i]$ *to* $\mathcal{A}_i$ *for every* $i \in [n]$*.*
4. *The challenger generates* $\alpha \leftarrow [n]$*, and gives* uk *to* $\mathcal{A}_\alpha$*.* $\mathcal{A}_\alpha$ *outputs* coin'*. The challenger outputs* 1 *if* $\mathsf{coin}' = \mathsf{coin}$ *and outputs* 0 *otherwise.*

We say that UE *is one-out-of-many one-time unclonable-indistinguishable if for any polynomial* $n = n(\lambda)$ *and QPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$*, it satisfies that*

$$\mathsf{Adv}^{\mathsf{om\text{-}ot\text{-}clone\text{-}ind}}_{\mathsf{UE},\mathcal{A}}(\lambda, n) := \Pr\left[\mathsf{Exp}^{\mathsf{om\text{-}ot\text{-}unclone\text{-}ind}}_{\mathsf{UE},\mathcal{A}}(\lambda, n) = 1\right] \leq \frac{1}{2} + \frac{1}{2n} + \mathsf{negl}(\lambda).$$

Theorem 3.2. *Let* $\mathsf{UE} = (\mathsf{KG}, \mathcal{E}nc, \mathcal{D}ec)$ *be a one-out-of-many one-time unclonable-indistinguishable secure SKUE scheme with the message space* $\mathcal{M}$*. Then,* UE *satisfies one-time IND-CPA security, that is,*

$$\mathcal{E}nc(\mathsf{uk}, \mathsf{m}_0) \overset{\mathsf{c}}{\approx} \mathcal{E}nc(\mathsf{uk}, \mathsf{m}_1)$$

for any $(\mathsf{m}_0, \mathsf{m}_1) \in \mathcal{M}$*, where* $\mathsf{uk} \leftarrow \mathsf{KG}(1^\lambda)$*.*

Proof. Suppose there exists $\mathcal{B}$ who can distinguish $\mathcal{E}nc(\mathsf{uk}, \mathsf{m}_0)$ from $\mathcal{E}nc(\mathsf{uk}, \mathsf{m}_1)$ with probability $\frac{1}{2} + p$ for some $(\mathsf{m}_0, \mathsf{m}_1) \in \mathcal{M}$ and inverse polynomial p, where $\mathsf{uk} \leftarrow \mathsf{KG}(1^\lambda)$. Consider the following $n = \frac{1}{p}$ tuple of adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$ for the one-out-of-many one-time unclonable-indistinguishable security. On input ct, $\mathcal{A}_0$ gives them to $\mathcal{B}$ and obtains $\mathcal{B}$'s guess coin', and sends it to $\mathcal{A}_i$ for every $i \in [n]$. $\mathcal{A}_i$ just outputs coin' if $\alpha = i$ is chosen by the challenger. We have $\mathsf{Adv}^{\mathsf{om\text{-}ot\text{-}unclone\text{-}ind}}_{\mathsf{UE},\mathcal{A}}(\lambda, n) = \frac{1}{2} + p$, which contradicts to the one-out-of-many unclonable-indistinguishable security since $p > \frac{1}{2n} = \frac{p}{2}$. □

Definition 3.3 (One-out-of-Many One-Time Anti-Piracy Security for Secret Key SDE). *Let* $\mathsf{SDE} = (\mathcal{KG}, \mathsf{Enc}, \mathcal{D}ec)$ *be a secret key SDE scheme. We consider one-out-of-many one-time anti-piracy game* $\mathsf{Exp}^{\mathsf{om\text{-}otanti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n)$ *between the challenger and an adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$ *below.*

1. *The challenger generates* $(\mathsf{ek}, dk) \leftarrow \mathcal{KG}(1^\lambda)$ *and sends* dk *to* $\mathcal{A}_0$*.*
2. $\mathcal{A}_0$ *creates a quantum state* q *over* n *registers* $\mathsf{R}_1, \cdots, \mathsf{R}_n$*.* $\mathcal{A}_0$ *sends* $(\mathsf{m}_0, \mathsf{m}_1)$ *to the challenger.* $\mathcal{A}_0$ *also sends* $q[\mathsf{R}_i]$ *to* $\mathcal{A}_i$ *for every* $i \in [n]$*.*
3. $\mathcal{A}_i$ *outputs* $\mathcal{D}_i = (\rho[\mathsf{R}_i], \boldsymbol{U}_i)$ *for every* $i \in [n]$*.*
4. *The challenger generates* $\alpha \leftarrow [n]$ *and* $\mathsf{coin} \leftarrow \{0,1\}$*, and generates* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, \mathsf{m}_{\mathsf{coin}})$*. The challenge runs* $\mathcal{D}_\alpha$ *on input* ct *and obtains* coin'*. The challenger outputs* 1 *if* $\mathsf{coin}' = \mathsf{coin}$ *and outputs* 0 *otherwise.*

We say that SDE *is one-out-of-many one-time anti-piracy secure if for any polynomial* $n = n(\lambda)$ *and QPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$*, it satisfies that*

$$\mathsf{Adv}^{\mathsf{om\text{-}otanti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n) := \Pr\left[\mathsf{Exp}^{\mathsf{om\text{-}otanti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n) = 1\right] \leq \frac{1}{2} + \frac{1}{2n} + \mathsf{negl}(\lambda).$$

We show that one-time strong anti-piracy security for secret key SDE implies one-ouf-of-many one-time anti-piracy for secret key SDE.

Theorem 3.4. *Let* SDE *be a secret key SDE scheme. If* SDE *is one-time strong anti-piracy secure, then* SDE *is also one-out-of-many one-time anti-piracy secure.*

Proof. $\mathsf{Exp}^{\mathsf{om\text{-}anti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n)$ is equivalent to the security game where item 4 is replaced with the following.

- The challenger generates $\alpha \leftarrow [n]$. For every $\alpha' \in [n]$, the challenger applies $\mathsf{ProjImp}(\boldsymbol{P}_{\alpha',D})$ to $\rho[\mathsf{R}_{\alpha'}]$ and obtains $p_{\alpha'}$. The challenger outputs 1 with probability p_α.

This equivalence follows from the definition of ProjImp. Using this version of $\mathsf{Exp}^{\mathsf{om\text{-}anti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n)$, we prove that $\mathsf{Adv}^{\mathsf{om\text{-}anti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n) \leq \frac{1}{2} + \frac{1}{2n} + \gamma$ for any inverse polynomial γ. Since SDE is strong anti-piracy secure, except for some single index i^*, p_i computed by the challenger is smaller than $\frac{1}{2} + \gamma$, with overwhelming probability. Thus, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{om\text{-}anti\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda, n) &\leq \frac{1}{n} \cdot 1 + \frac{n-1}{n} \cdot (\frac{1}{2} + \gamma) + \mathsf{negl}(\lambda) \\ &\leq \frac{1}{2} + \frac{1}{2n} + \gamma.\end{aligned}$$

This completes the proof. □

3.2 From Secret-Key SDE to SKUE: One-out-of-Many Setting

We present a transformation from a secret key SDE scheme that satisfies Definition 3.3 into a SKUE scheme that satisfies Definition 3.1. Georgiou and Zhandry developed this transformation [GZ20]. Note that they do not consider one-out-of-many security for secret key SDE and SKUE. We show that their transformation works even in the one-out-of-many setting.

Let $\mathsf{SDE} = (\mathsf{SDE}.\mathcal{KG}, \mathsf{SDE.Enc}, \mathsf{SDE}.\mathcal{D}ec)$ be a secret key SDE scheme. We also let ℓ be the length of ciphertexts of SDE. We construct a SKUE scheme $\mathsf{UE} = (\mathsf{UE.KG}, \mathsf{UE}.\mathcal{E}nc, \mathsf{UE}.\mathcal{D}ec)$ as follows.

$\mathsf{UE.KG}(1^\lambda)$**:**

- Output $\mathsf{uk} \leftarrow \{0,1\}^\ell$.

$\mathsf{UE}.\mathcal{E}nc(\mathsf{uk}, \mathsf{m})$**:**

- Generate $(\mathsf{sde.ek}, \mathsf{sde}.d\mathcal{K}) \leftarrow \mathsf{SDE}.\mathcal{KG}(1^\lambda)$.
- Compute $\mathsf{sde.ct} \leftarrow \mathsf{SDE.Enc}(\mathsf{sde.ek}, \mathsf{m})$.
- Output $\mathsf{ue}.ct := (\mathsf{sde.ct} \oplus \mathsf{uk}, \mathsf{sde}.d\mathcal{K})$.

$\mathsf{UE}.\mathcal{D}ec(\mathsf{uk}, \mathsf{ue}.ct)$**:**

- Parse $(\mathsf{ct}'_1, \mathsf{sde}.d\mathcal{K}) = \mathsf{ue}.ct$.
- Compute $\mathsf{sde.ct}' := \mathsf{ct}'_1 \oplus \mathsf{uk}$.
- Output $\mathsf{m}' \leftarrow \mathsf{SDE}.\mathcal{D}ec(\mathsf{sde}.d\mathcal{K}, \mathsf{sde.ct}')$.

Theorem 3.5. *If* SDE *is one-out-of-many one-time anti-piracy secure,* UE *is one-out-of-many one-time unclonable-indistinguishable secure.*

Proof. Let n be any polynomial of λ. We construct an adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1, \cdots, \mathcal{B}_n)$ for SDE by using the adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$ for UE.

1. $\mathcal{B}_0$ is given $\mathsf{sde}.d\mathcal{K}$ from its challenger.
2. $\mathcal{B}_0$ runs $\mathcal{A}_0$ and receives $(\mathsf{m}_0, \mathsf{m}_1)$, and passes $(\mathsf{m}_0, \mathsf{m}_1)$ to its challenger.

3. $\mathcal{B}_0$ generates $\mathsf{uk} \leftarrow \{0,1\}^\ell$, sets $\mathsf{ue}.ct := (\mathsf{uk}, \mathsf{sde}.d\mathcal{K})$, and passes $\mathsf{ue}.ct$ to $\mathcal{A}_0$. Then, $\mathcal{A}_0$ create a quantum state $q_{\mathcal{A}}$ over n registers $\mathsf{C}_1, \cdots, \mathsf{C}_n$. $\mathcal{B}_0$ receives them.

4. $\mathcal{B}_0$ creates a quantum state $q_{\mathcal{B}}$ over n registers $\mathsf{R}_1, \cdots, \mathsf{R}_n$ such that $q_{\mathcal{B}}[\mathsf{R}_i] := (\mathsf{uk}, q_{\mathcal{A}}[\mathsf{C}_i])$ for every $i \in [n]$, then $\mathcal{B}_0$ sends $q_{\mathcal{B}}[\mathsf{R}_i]$ to $\mathcal{B}_i$ for every $i \in [n]$.

5. For every $i \in [n]$, $\mathcal{B}_i$ outputs $\mathcal{D}_i = (q_{\mathcal{B}}[\mathsf{R}_1], \boldsymbol{U}_i)$, where $\boldsymbol{U}_i$ is a unitary that takes $\mathsf{sde.ct}$ and $q_{\mathcal{B}}[\mathsf{R}_1]$ as inputs, and outputs $\mathsf{coin}' \leftarrow \mathcal{A}_i(q_{\mathcal{A}}[\mathsf{C}_i], \mathsf{uk} \oplus \mathsf{sde.ct})$.

If the challenger of one-out-of-many security of SDE chooses $\alpha \leftarrow [n]$ and $\mathsf{coin} \leftarrow \{0,1\}$, and generates $\mathsf{sde.ct} \leftarrow \mathsf{SDE.Enc}(\mathsf{sde.ek}, \mathsf{m}_{\mathsf{coin}})$, then the challenger runs $\mathcal{D}_\alpha$ on input $\mathsf{sde.ct}$ and obtains the output $\mathsf{coin}' \leftarrow \mathcal{A}_\alpha(q_{\mathcal{A}}[\mathsf{C}_\alpha], \mathsf{uk} \oplus \mathsf{sde.ct})$. If $\mathsf{sde.ct} = \mathsf{SDE.Enc}(\mathsf{sde.ek}, \mathsf{m}_{\mathsf{coin}})$, then $\mathsf{ue}.ct = (\mathsf{uk}, \mathsf{sde}.d\mathcal{K}) = \mathsf{UE}.\mathcal{E}nc(\mathsf{sde.ct} \oplus \mathsf{uk}, \mathsf{m}_{\mathsf{coin}})$. $\mathcal{B}$ correctly simulates the one-out-of-many security game of SKUE for $\mathcal{A}$ since uk is a uniformly random string. Therefore, the probability that $\mathcal{A}$ succeeds in breaking UE is bounded by the probability that $\mathcal{B}$ succeeds in breaking SDE. This completes the proof. □

4 One-Time Secret Key SDE from LWE

We construct secret key SDE satisfying one-time strong anti-piracy based on the LWE assumption in this section.

4.1 Tools

Ciphertext-policy functional encryption. We introduce ciphertext-policy functional encryption (CPFE). Since we consider single-key setting by default, we use a simplified syntax where the setup algorithm takes as input x and outputs a master public key and a functional decryption key for x. There is no key generation algorithm.

Definition 4.1 (Single-Key Ciphertext-Policy Functional Encryption). *A single-key CPFE scheme for the circuit space $\mathcal{C}$ and the input space $\mathcal{X}$ is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$.

- *The setup algorithm* Setup *takes as input a security parameter* 1^λ *and an input* $x \in \mathcal{X}$*, and outputs a master public key* MPK *and functional decryption key* sk_x.
- *The encryption algorithm* Enc *takes as input the master public key* MPK *and* $C \in \mathcal{C}$*, and outputs a ciphertext* ct.
- *The decryption algorithm* Dec *takes as input a functional decryption key* sk_x *and a ciphertext* ct*, and outputs* y.

Decryption Correctness: *We require* $\mathsf{Dec}(\mathsf{sk}_x, \mathsf{Enc}(\mathsf{MPK}, C)) = C(x)$ *for every* $C \in \mathcal{C}$*,* $x \in \mathcal{X}$*, and* $(\mathsf{MPK}, \mathsf{sk}_x) \leftarrow \mathsf{Setup}(1^\lambda, x)$.

Definition 4.2 (1-Bounded Security). *Let* CPFE *be a single-key CPFE scheme. We define the game* $\mathsf{Expt}^{\mathsf{1\text{-}bounded}}_{\mathsf{CPFE},\mathcal{A}}(\lambda, \mathsf{coin})$ *as follows.*

1. *$\mathcal{A}$ sends $x \in \mathcal{X}$ to the challenger.*
2. *The challenger generates* $(\mathsf{MPK}, \mathsf{sk}_x) \leftarrow \mathsf{Setup}(1^\lambda, x)$ *and sends* $(\mathsf{MPK}, \mathsf{sk}_x)$ *to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs* $(\mathsf{C}_0, \mathsf{C}_1)$ *such that* $\mathsf{C}_0(x) = \mathsf{C}_1(x)$*. The challenger generates* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathsf{C}_{\mathsf{coin}})$*, and sends* ct *to $\mathcal{A}$.*
4. *$\mathcal{A}$ outputs* $\mathsf{coin}' \in \{0,1\}$.

We say that CPFE *is* 1*-bounded secure if for every QPT $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{1\text{-}bounded}}_{\mathsf{CPFE},\mathcal{A}}(\lambda) = \left|\Pr\left[\mathsf{Expt}^{\mathsf{1\text{-}bounded}}_{\mathsf{CPFE},\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{1\text{-}bounded}}_{\mathsf{CPFE},\mathcal{A}}(\lambda, 1) = 1\right]\right| = \mathsf{negl}(\lambda).$$

Definition 4.3 (Succinct Key). *We say that a single-key CPFE scheme satisfies succinct key property if there exist two algorithms* HKG *and* Hash *satisfying the following conditions.*

- $\mathsf{Setup}(1^\lambda, x)$ *runs* $\mathsf{hk} \leftarrow \mathsf{HKG}(1^\lambda, 1^{|x|})$*, compute* $h \leftarrow \mathsf{Hash}(\mathsf{hk}, x)$*, and outputs* $\mathsf{MPK} := (\mathsf{hk}, h)$ *and* $\mathsf{sk}_x := x$*. For the setup of a CPFE scheme with succinct key property, we omit to write* $\mathsf{sk}_x := x$ *from the output of* Setup *and we simply write* $\mathsf{MPK} \leftarrow \mathsf{Setup}(1^\lambda, x)$*.*
- *The length of* h *output by* Hash *is* λ *regardless of the length of the input* x*.*

Theorem 4.4. *If the LWE or exponentially-hard LPN assumption holds, there exists single-key CPFE with succinct key for* P/poly*.*

See Appendix B for the proof of Theorem 4.4.

Monogamy of entanglement. We review the monogamy of entanglement property of of BB84 states [TFKW13] and its variant.

Theorem 4.5 (Monogamy Property of BB84 States [TFKW13]). *Consider the following game between a challenger and an adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*.*

1. *The challenger picks a uniformly random strings* $x \in \{0,1\}^n$ *and* $\theta \in \{0,1\}^n$*. It sends* $\left|x^\theta\right\rangle := H^{\theta[1]} |x[1]\rangle \otimes \cdots \otimes H^{\theta[n]} |x[n]\rangle$ *to* $\mathcal{A}_0$*.*
2. $\mathcal{A}_0$ *creates a bipartite state* q *over registers* R_1 *and* R_2*. Then,* $\mathcal{A}_0$ *sends register* R_1 *to* $\mathcal{A}_1$ *and register* R_2 *to* $\mathcal{A}_2$*.*
3. θ *is then sent to both* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*.* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *return respectively* x'_1 *and* x'_2*.*

Let $\mathsf{MoEBB84}(\mathcal{A}, \lambda)$ *be a random variable which takes the value* 1 *if* $x'_1 = x'_2 = x$*, and takes the value* 0 *otherwise. Then, there exists an exponential function* exp *such that for any adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*, it holds that*

$$\Pr[\mathsf{MoEBB84}(\mathcal{A}, \lambda) = 1] \leq 1/\exp(n).$$

We introduce a variant of the monogamy property above where the adversary can select a leakage function Leak and obtain $\mathsf{Leak}(x)$.

Theorem 4.6 (Monogamy Property of BB84 States with Leakage). *Consider the following game between a challenger and an adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*.*

1. $\mathcal{A}$ *sends a function* Leak *whose output length is* ℓ *to the challenger.*
2. *The challenger picks a uniformly random strings* $x \in \{0,1\}^n$ *and* $\theta \in \{0,1\}^n$*. It sends* $\left|x^\theta\right\rangle := H^{\theta[1]} |x[1]\rangle \otimes \cdots \otimes H^{\theta[n]} |x[n]\rangle$ *and* $\mathsf{Leak}(x)$ *to* $\mathcal{A}_0$*.*
3. $\mathcal{A}_0$ *creates a bipartite state* q *over registers* R_1 *and* R_2*. Then,* $\mathcal{A}_0$ *sends register* R_1 *to* $\mathcal{A}_1$ *and register* R_2 *to* $\mathcal{A}_2$*.*
4. θ *is then sent to both* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*.* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *return respectively* x'_1 *and* x'_2*.*

Let $\mathsf{MoEBB84Leak}(\mathcal{A}, \lambda)$ *be a random variable which takes the value* 1 *if* $x'_1 = x'_2 = x$*, and takes the value* 0 *otherwise. Then, there exists an exponential function* exp *such that for any adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*, it holds that*

$$\Pr[\mathsf{MoEBB84Leak}(\mathcal{A}, \lambda) = 1] \leq 2^\ell/\exp(n).$$

Especially, if ℓ *is independent of* n*, the right hand side is negligible in* λ *by setting* n *appropriately.*

We can reduce Theorem 4.6 to Theorem 4.5 by guessing $\mathsf{Leak}(x)$ with probability $1/2^\ell$.

4.2 Construction

We use a CPFE scheme with succinct key property $\mathsf{CPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ to construct a secret key SDE scheme $\mathsf{SDE} = (\mathsf{SDE}.\mathcal{KG}, \mathsf{SDE.Enc}, \mathsf{SDE}.\mathcal{D}ec)$. The description of SDE is as follows. The plaintext space of SDE is $\{0,1\}^\ell$.

$\mathsf{SDE}.\mathcal{KG}(1^\lambda)$**:**

- Generate $x, \theta \leftarrow \{0,1\}^n$.
- Generate $\left|x^\theta\right\rangle = H^{\theta[1]}\left|x[1]\right\rangle \otimes \dots H^{\theta[n]}\left|x[n]\right\rangle$.
- Generate $\mathsf{MPK} \leftarrow \mathsf{Setup}(1^\lambda, x)$.
- Output $\mathsf{ek} := (\theta, \mathsf{MPK})$ and $\mathit{dk} := \left|x^\theta\right\rangle$.

$\mathsf{SDE.Enc}(\mathsf{ek}, m)$**:**

- Parse $\mathsf{ek} = (\theta, \mathsf{MPK})$.
- Let $C[m]$ be a constant circuit that outputs m on any input. C is padded so that it has the same size as C^* appeared in the security proof.
- Compute $\mathsf{ct} \leftarrow \mathsf{CPFE.Enc}(\mathsf{MPK}, C[m])$.
- Output $\mathsf{sdct} := (\theta, \mathsf{ct})$.

$\mathsf{SDE}.\mathcal{D}ec(\mathit{dk}, \mathsf{sdct})$**:**

- Parse $\mathit{dk} = \left|x^\theta\right\rangle$ and $\mathsf{sdct} = (\theta, \mathsf{ct})$.
- Compute x from $\left|x^\theta\right\rangle$ and θ.
- Output $m \leftarrow \mathsf{Dec}(x, \mathsf{ct})$.

Theorem 4.7. *If* CPFE *is a CPFE scheme that satisfies succinct key property and* 1*-bounded security,* SDE *is a one-time strong anti-piracy secure single-decryptor SKE.*

From Theorems 3.4, 3.5, 4.4 and 4.7, we immediately obtain the following corollary.

Corollary 4.8. *If the LWE assumption holds, there exists one-out-of-many unclonable-indistinguishable secure unclonable encryption.*

Proof of Theorem 4.7. Let $\gamma \in [0,1]$ and $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ be any QPT adversary. $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively output $\mathcal{D}_1 = (\mathsf{R}_1, \boldsymbol{U}_1)$ and $\mathcal{D}_2 = (\mathsf{R}_2, \boldsymbol{U}_2)$, where R_1 and R_2 have a possibly entangled quantum state ρ, and $\boldsymbol{U}_\alpha = (\boldsymbol{U}_{\alpha,\mathsf{ct}})_{\mathsf{ct}}$ for $\alpha \in [2]$. We define the collection of binary projective measurements $\boldsymbol{P}_\alpha = (\boldsymbol{P}_{\alpha,b,\mathsf{ct}}, \boldsymbol{I} - \boldsymbol{P}_{\alpha,b,\mathsf{ct}})_{b,\mathsf{ct}}$ for every $\alpha \in [2]$, the distribution D, and the mixture of $\boldsymbol{P}_\alpha$ with respect to D $\boldsymbol{P}_{\alpha,D}$ for every $\alpha \in [2]$ in the same way as Definition 2.21. Then, we can write

$$\mathsf{Adv}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma) = \Pr\left[p_1 > \frac{1}{2} + \gamma \wedge p_2 > \frac{1}{2} + \gamma\right],$$

where p_α is the result of applying $\mathsf{ProjImp}(\boldsymbol{P}_{\alpha,D})$ to R_α.

We assume that $\mathsf{Adv}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma) = \eta$ for some inverse polynomial η. Using $\mathcal{A}$, we construct $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2)$ that attacks the monogamy property of BB84 states with leakage. Recall that since CPFE satisfies succinct key property, there exist two algorithms HKG and Hash such that $\mathsf{Setup}(1^\lambda, x)$ runs $\mathsf{hk} \leftarrow \mathsf{HKG}(1^\lambda, 1^{|x|})$, compute $h \leftarrow \mathsf{Hash}(\mathsf{hk}, x)$, and outputs $\mathsf{MPK} := (\mathsf{hk}, h)$, where h is a λ-bit string. (See Definition 4.3.)

1. $\mathcal{B}_0$ generates $\mathsf{hk} \leftarrow \mathsf{HKG}(1^\lambda, 1^n)$ and sends a function $\mathsf{Leak}(\cdot) := \mathsf{Hash}(\mathsf{hk}, \cdot)$ to the challenger.
2. $\mathcal{B}_0$ is given $\left|x^\theta\right\rangle$ and $\mathsf{Leak}(x) = h$. $\mathcal{B}_0$ sets $\mathsf{MPK} := (\mathsf{hk}, h)$.
3. By setting $\mathit{dk} := \left|x^\theta\right\rangle$, $\mathcal{B}_0$ simulates $\mathsf{Exp}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma)$ for $\mathcal{A}_0$ and obtains (m_0, m_1) and a quantum state q over registers R_1 and R_2. Then, $\mathcal{B}_0$ sends $(\gamma, \mathsf{MPK}, m_0, m_1, \mathsf{R}_1)$ to $\mathcal{B}_1$ and $(\gamma, \mathsf{MPK}, m_0, m_1, \mathsf{R}_2)$ to $\mathcal{B}_2$.

4. θ is then sent to both $\mathcal{B}_1$ and $\mathcal{B}_2$. $\mathcal{B}_1$ and $\mathcal{B}_2$ behave as follows.

   - $\mathcal{B}_\alpha$ first sets $\mathsf{ek} = (\theta, \mathsf{MPK})$. $\mathcal{B}_\alpha$ simulates $\mathsf{Exp}^{\mathsf{ot\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma)$ for $\mathcal{A}_\alpha$ and obtains $\mathcal{D}_\alpha = (q_\alpha, \boldsymbol{U}_\alpha)$. $\mathcal{B}_\alpha$ computes $x'_\alpha \leftarrow \mathcal{E}\mathit{xtract}(\mathsf{MPK}, m^*_0, m^*_1, \mathcal{D}_\alpha, \gamma)$ and outputs x'_α, where the algorithm $\mathcal{E}\mathit{xtract}$ is described below.

$\mathcal{E}\mathit{xtract}(\mathsf{MPK}, \mathsf{m}^*_0, \mathsf{m}^*_1, \mathcal{D}_\alpha, \epsilon)$**:**

- Let $\epsilon' = \epsilon/4(n+1)$ and $\delta' = 2^{-\lambda}$.
- Parse $(q_\alpha, \boldsymbol{U}_\alpha) \leftarrow \mathcal{D}_\alpha$. Let D_i be the following distribution for every $i \in [\lambda]$.

  D_i**:** Generate $b \leftarrow \{0,1\}$. Generate $\mathsf{ct} \leftarrow \mathsf{CPFE.Enc}(\mathsf{MPK}, C^*[b, m_0, m_1, i])$, where $C^*[b, m_0, m_1, i]$ is a circuit that takes x as input and outputs $m_{b\oplus x[i]}$. Output (b, ct).
- Compute $\widetilde{p}_{\alpha,0} \leftarrow \mathcal{API}^{\epsilon',\delta'}_{\mathcal{P},D}(q_\alpha)$. If $\widetilde{p}_{\alpha,0} < \frac{1}{2} + \epsilon - 4\epsilon'$, return $\bot$. Otherwise, letting $q_{\alpha,0}$ be the post-measurement state, go to the next step.
- For all $i \in [\lambda]$, do the following.
  1. Compute $\widetilde{p}_{\alpha,i} \leftarrow \mathcal{API}^{\epsilon',\delta'}_{\mathcal{P},D_i}(q_{\alpha,i-1})$. Let $q_{\alpha,i}$ be the post-measurement state.
  2. If $\widetilde{p}_{\alpha,i} > \frac{1}{2} + \epsilon - 4(i+1)\epsilon'$, set $x'_\alpha[i] = 0$. If $\widetilde{p}_{\alpha,i} < \frac{1}{2} - \epsilon + 4(i+1)\epsilon'$, set $x'_\alpha[i] = 1$. Otherwise, exit the loop and output $\bot$.
- Output $x'_\alpha = x'_\alpha[1] \| \cdots \| x'_\alpha[\lambda]$.

We will estimate $\Pr[\mathsf{MoEBB84Leak}(\mathcal{A},\lambda) = 1]$. We define the events BadDec_α, and $\mathsf{BadExt}_{\alpha,i}$ for every $i \in [\lambda]$.

BadDec_α**:** When $\mathcal{B}_\alpha$ runs $\mathcal{E}\mathit{xtract}(\mathsf{MPK}, m^*_0, m^*_1, \mathcal{D}_\alpha, \epsilon)$, $\widetilde{p}_{\alpha,0} < \frac{1}{2} + \epsilon - 4\epsilon'$ holds.

$\mathsf{BadExt}_{\alpha,i}$**:** When $\mathcal{B}_\alpha$ runs $\mathcal{E}\mathit{xtract}(\mathsf{MPK}, m^*_0, m^*_1, \mathcal{D}_\alpha, \epsilon)$, the following conditions hold.

- $\widetilde{p}_{\alpha,0} \geq \frac{1}{2} + \epsilon - 4\epsilon'$ holds.
- $x'_\alpha[j] = x[j]$ holds for every $j \in [i-1]$.
- $x'_\alpha[i] \neq x[i]$ holds.

From the assumption that $\mathsf{Adv}^{\mathsf{ek\text{-}santi\text{-}piracy}}_{\mathsf{SDE},\mathcal{A}}(\lambda,\gamma) = \eta$, for $\widetilde{p}_{\alpha,0}$ computed in $\mathcal{E}\mathit{xtract}$, $\widetilde{p}_{\alpha,0} \geq \frac{1}{2} + \epsilon - \epsilon'$ holds with probability $\eta - \mathsf{negl}(\lambda)$ for $\alpha \in [2]$, due to the first item of Theorem 2.5. This means that $\Pr[\mathsf{BadDec}_1 \vee \mathsf{BadDec}_2] = 1 - \eta + \mathsf{negl}(\lambda)$. Then, we have

$$\begin{aligned}\Pr[\mathsf{MoEBB84Leak}(\mathcal{A},\lambda) = 1] &\geq 1 - \left(\Pr[\mathsf{BadDec}_1 \vee \mathsf{BadDec}_2] + \sum_{i\in[\lambda]} \Pr[\mathsf{BadExt}_{1,i}] + \sum_{i\in[\lambda]} \Pr[\mathsf{BadExt}_{2,i}]\right) \\ &= \eta - \mathsf{negl}(\lambda) - \left(\sum_{i\in[\lambda]} \Pr[\mathsf{BadExt}_{1,i}] + \sum_{i\in[\lambda]} \Pr[\mathsf{BadExt}_{2,i}]\right)\end{aligned}$$

Estimation of $\Pr[\mathsf{BadExt}_{\alpha,i}]$ **for every** $\alpha \in [2]$ **and** $i \in [\lambda]$**.** We first estimate $\Pr[\mathsf{BadExt}_{\alpha,1}]$. We first consider the case of $x[1] = 0$. From the first item of the event, we have $\widetilde{p}_{\alpha,0} > \frac{1}{2} + \epsilon - 4\epsilon'$. Let $\widetilde{p}'_{\alpha,0} \leftarrow \mathcal{API}^{\epsilon',\delta'}_{\mathcal{P},D}(q_{\alpha,0})$. From, the almost-projective property of $\mathcal{API}$, we have

$$\Pr\left[\widetilde{p}'_{\alpha,0} > \frac{1}{2} + \epsilon - 4\epsilon' - \epsilon'\right] \geq 1 - \delta'.$$

Lemma 4.9. *When $x[1] = 0$, D_1 is computationally indistinguishable from D.*

Proof. The difference between D_1 and D is that ct is generated as $\mathsf{ct} \leftarrow \mathsf{CPFE.Enc}(\mathsf{MPK}, C^*[b, m_0, m_1, 1])$ in D_1 and it is generated as $\mathsf{ct} \leftarrow \mathsf{CPFE.Enc}(\mathsf{MPK}, C[\mathsf{m}_b])$ in D. From the condition that $x[1] = 0$, we have $C^*[b, m_0, m_1, 1](x) = C[m_b](x) = m_b$. Thus, from the 1-bounded security of CPFE, D_1 and D are computationally indistinguishable when $x[1] = 0$. □

Thus, from Theorem 2.6 and Lemma 4.9, we have

$$1 - \delta' \leq \Pr\left[\widetilde{p}'_{\alpha,0} > \frac{1}{2} + \epsilon - 5\epsilon'\right] \leq \Pr\left[\widetilde{p}_{\alpha,1} > \frac{1}{2} + \epsilon - 8\epsilon'\right] + \mathsf{negl}(\lambda).$$

This means that $\Pr[\mathsf{BadExt}_{\alpha,1}] = \mathsf{negl}(\lambda)$ when $x[1] = 0$. We next consider the case of $x[1] = 1$. We define the following distribution $D^{\mathtt{rev}}$.

$D^{\mathtt{rev}}$**:** Generate $(b, \mathsf{ct}) \leftarrow D$. Output $(1 \oplus b, \mathsf{ct})$.

That is, the first bit of the output is flipped from D. Then, for any random coin r, we have $(\boldsymbol{P}_{D^{\mathtt{rev}}(r)}, \boldsymbol{Q}_{D^{\mathtt{rev}}(r)}) = (\boldsymbol{Q}_{D(r)}, \boldsymbol{P}_{D(r)})$. This is because we have $\boldsymbol{Q}_{b,\mathsf{ct}} = \boldsymbol{I} - \boldsymbol{P}_{b,\mathsf{ct}} = \boldsymbol{P}_{1\oplus b,\mathsf{ct}}$ for any tuple (b, ct). Therefore, $\mathcal{API}^{\epsilon',\delta'}_{\mathcal{P},D^{\mathtt{rev}}}$ is exactly the same process as $\mathcal{API}^{\epsilon',\delta'}_{\mathcal{P}^{\mathtt{rev}},D}$, where $\mathcal{P}^{\mathtt{rev}} = (\boldsymbol{Q}_{b,\mathsf{ct}}, \boldsymbol{P}_{b,\mathsf{ct}})_{b,\mathsf{ct}}$. Let $\widetilde{p}'_{\alpha,0} \leftarrow \mathcal{API}^{\epsilon',\delta'}_{\mathcal{P},D^{\mathtt{rev}}}(q_{\alpha,0})$. From, the reverse-almost-projective property of $\mathcal{API}$, we have

$$\Pr\left[\widetilde{p}'_{\alpha,0} < \frac{1}{2} - \epsilon + 4\epsilon' + \epsilon'\right] \geq 1 - \delta'.$$

Lemma 4.10. *When $x[1] = 1$, D_1 is computationally indistinguishable from $D^{\mathtt{rev}}$.*

Proof. We see that $D^{\mathtt{rev}}$ is identical to the following distribution.

- Generate $b \leftarrow \{0,1\}$ and $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, \mathsf{m}_{1\oplus b})$. Output (b, ct).

Then, the difference between D_1 and $D^{\mathtt{rev}}$ is that ct is generated as $\mathsf{ct} \leftarrow \mathsf{CPFE.Enc}(\mathsf{MPK}, C^*[b, m_0, m_1, 1])$ in D and it is generated as $\mathsf{ct} \leftarrow \mathsf{CPFE.Enc}(\mathsf{MPK}, C[\mathsf{m}_{1\oplus b}])$ in $D^{\mathtt{rev}}$. From the condition that $x[1] = 1$, we have $C^*[b, m_0, m_1, 1](x) = C[m_{1\oplus b}](x) = \mathsf{m}_{1\oplus b}$. Thus, from the 1-bounded security of CPFE, D_1 and $D^{\mathtt{rev}}$ are computationally indistinguishable when $x[1] = 1$. □

Thus, from Theorem 2.6 and Lemma 4.10, we have

$$1 - \delta' \leq \Pr\left[\widetilde{p}'_{\alpha,0} < \frac{1}{2} - \epsilon + 5\epsilon'\right] \leq \Pr\left[\widetilde{p}_{\alpha,1} < \frac{1}{2} - \epsilon + 8\epsilon'\right] + \mathsf{negl}(\lambda).$$

This means that $\Pr[\mathsf{BadExt}_{\alpha,1}] = \mathsf{negl}(\lambda)$ when $x[1] = 1$.

Overall, $\Pr[\mathsf{BadExt}_{\alpha,1}] = \mathsf{negl}(\lambda)$ regardless of the value of $x[1]$. We can similarly show that $\Pr[\mathsf{BadExt}_{\alpha,i}] = \mathsf{negl}(\lambda)$ for $i \in \{2, \cdots, \lambda\}$ using the fact that D_i is computationally indistinguishable from D if $x[i] = 0$ and it is computationally indistinguishable form $D^{\mathtt{rev}}$ if $x[i] = 1$. We omit the details.

From the above discussion, we have $\Pr[\mathsf{MoEBB84Leak}(\mathcal{A}, \lambda) = 1] = \eta - \mathsf{negl}(\lambda)$ for some inverse polynomial η, which contradicts to the monogamy property of BB84 states with leakage. This completes the proof of Theorem 4.7. □

5 Quantum Copy-Protection from Unclonable Encryption

We introduce one-out-of-many copy protection security for single-bit output point functions and present how to achieve it using one-out-of-many secure unclonable encyption in this section.

5.1 Definition

Definition 5.1 (Copy-Protection (Syntax)). *A copy-protection scheme* CP *for a family of circuits* $\mathcal{C}$ *consists of two algorithms* $(\mathit{CopyProtect}, \mathit{Eval})$.

$\mathit{CopyProtect}(1^\lambda, C) \to \rho$**:** *The copy-protection algorithm takes as input the security parameter* 1^λ*, a circuit* $C \in \mathcal{C}$*, and outputs a quantum state* ρ.

$\mathit{Eval}(\rho, x)$**:** *The evaluation algorithm takes as input a quantum state* ρ *and an input* x*, and outputs* y.

Evaluation Correctness: *For every circuit* C *and input* x*, we have*

$$\Pr\left[\mathit{Eval}(\rho, x) = C(x) \mid \rho \leftarrow \mathit{CopyProtect}(1^\lambda, C)\right] = 1 - \mathsf{negl}(\lambda).$$

Remark 5.2. We can assume without loss of generality that a copy protected program ρ output by $\mathit{CopyProtect}$ is reusable, that is, it can be reused polynomially many times. This is because the output of Eval on input ρ and any input x is almost deterministic by correctness, and thus such an operation can be done without almost disturbing ρ by the gentle measurement lemma [Win99].

We focus on copy protection scheme for a family of single-bit output point functions that we denote $\mathcal{PF}^1$. We also define a family of single-bit output point functions $\mathcal{PF}^1_{\ell_{\mathsf{inp}}}$ as $\mathcal{PF}^1_{\ell_{\mathsf{inp}}} = \{f_y\}_{y\in\{0,1\}^{\ell_{\mathsf{inp}}}}$, where f_y outputs 1 if the input is $y \in \{0,1\}^{\ell_{\mathsf{inp}}}$ and 0 otherwise.

We review the widely used copy-protection security for $\mathcal{PF}^1$ originally introduced by Coladangelo et al. [CMP20].

Definition 5.3 (Copy-Protection Security for $\mathcal{PF}^1$). *Let* CP *be a copy protection scheme for* $\mathcal{PF}^1_{\ell_{\mathsf{inp}}}$*. Let* D_Y *be a distribution over* $\{0,1\}^{\ell_{\mathsf{inp}}}$*. Let* $D_X(\cdot)$ *be a distribution over* $\{0,1\}^{\ell_{\mathsf{inp}}} \times \{0,1\}^{\ell_{\mathsf{inp}}}$*, where* $D_X(\cdot)$ *takes as input* $y' \in \{0,1\}^{\ell_{\mathsf{inp}}}$*. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{cp\text{-}pf1}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda)$ *for* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$.

1. *The challenger generates* $y \leftarrow D_Y$*. The challenger generates* $\rho \leftarrow \mathit{CopyProtect}(1^\lambda, f_y)$ *and sends* ρ *to* $\mathcal{A}_0$.
2. $\mathcal{A}_0$ *creates a bipartite state* q *over registers* R_1 *and* R_2*.* $\mathcal{A}_0$ *sends* $q[\mathsf{R}_1]$ *and* $q[\mathsf{R}_2]$ *to* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*, respectively.*
3. *The challenger generates* $(x_1, x_2) \leftarrow D_X(y)$*, and sends* x_1 *and* x_2 *to* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*, respectively.*
4. $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *respectively output* b_1 *and* b_2*. If* $b_i = f_y(x_i)$ *for* $i \in \{1,2\}$*, the challenger outputs* 1*, otherwise outputs* 0.

We define $p^{\mathtt{triv}} = \max_{i\in\{1,2\},\mathcal{S}} p_{i,\mathcal{S}}$*, where*

$$p_{i,\mathcal{S}} = \Pr\left[b_i = f_y(x_i) \;\middle|\; \begin{array}{l} y \leftarrow D_Y, (x_1, x_2) \leftarrow D_X(y) \\ b_i \leftarrow \mathcal{S}(x_i) \end{array}\right]$$

and the maximization is done by all possibly computationally unbounded algorithm $\mathcal{S}$.

We say that CP *satisfies copy-protection security for* $\mathcal{PF}^1$ *with respect to* D_Y *and* D_X *if for any QPT* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{cp\text{-}pf1}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{cp\text{-}pf1}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda) = 1\right] \le p^{\mathtt{triv}} + \mathsf{negl}(\lambda).$$

The following definition is a natural adaptation of Definition 5.3 into one-out-of-many setting.

Definition 5.4 (One-out-of-Many Copy-Protection Security for $\mathcal{PF}^1$). *Let* CP *be a copy protection scheme for* $\mathcal{PF}^1_{\ell_{\mathsf{inp}}}$*. Let* D_Y *and* $D_X(\cdot)$ *be distributions over* $\{0,1\}^{\ell_{\mathsf{inp}}}$*, where* $D_X(\cdot)$ *takes as input* $y' \in \{0,1\}^{\ell_{\mathsf{inp}}}$*. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda, n)$ *for* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$.

1. *The challenger generates* $y \leftarrow D_Y$*. The challenger generates* $\rho \leftarrow \mathit{CopyProtect}(1^\lambda, f_y)$ *and sends* ρ *to* $\mathcal{A}_0$.

2. *$\mathcal{A}_0$ creates a quantum state q over n registers $\mathsf{R}_1, \cdots, \mathsf{R}_n$. $\mathcal{A}_0$ sends $q[\mathsf{R}_i]$ to $\mathcal{A}_i$ for every $i \in [n]$.*

3. *The challenger generates $\alpha \leftarrow [n]$. The challenger generates $x \leftarrow D_X(y)$ and sends x to $\mathcal{A}_\alpha$. $\mathcal{A}_\alpha$ outputs b_α. If $b_\alpha = f_y(x)$, the challenger outputs 1, otherwise outputs 0.*

We define $p^{\mathtt{triv}} = \max_{\mathcal{S}} p_{\mathcal{S}}$, where

$$p_{\mathcal{S}} = \Pr\left[b = f_y(x) \;\middle|\; \begin{array}{l} y \leftarrow D_Y, x \leftarrow D_X(y) \\ b \leftarrow \mathcal{S}(x) \end{array}\right]$$

and the maximization is done by all possibly computationally unbounded algorithm $\mathcal{S}$.

We say that CP *satisfies one-out-of-many copy-protection security for $\mathcal{PF}^1$ with respect to D_Y and D_X if for any polynomial $n = n(\lambda)$ and QPT $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda, n) := \Pr\left[\mathsf{Exp}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda, n) = 1\right] \leq \frac{1}{n} \cdot 1 + \frac{n-1}{n} \cdot p^{\mathtt{triv}} + \mathsf{negl}(\lambda).$$

5.2 Construction

We construct a copy-protection scheme for single-bit output point functions $\mathcal{PF}^1_{\ell_{\mathsf{inp}}}$, where ℓ_{inp} is specified later. We use the following tools:

- SKUE $\mathsf{UE} = (\mathsf{UE.KG}, \mathsf{UE}.\mathcal{E}nc, \mathsf{UE}.\mathcal{D}ec)$. Suppose the plaintext space of UE is $\{0,1\}^\lambda$ and the key length is ℓ_{uk}.
- Injective commitment scheme with equivocal mode $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{EqSetup}, \mathsf{Open})$. Suppose the message space of Com is $\{0,1\}^{\ell_{\mathsf{uk}}}$ and the random coin space is $\{0,1\}^{\ell_{\mathsf{comr}}}$.
- Compute-and-compare obfuscation CC.Obf with the simulator CC.Sim. In this section, the message feed to CC.Obf is fixed to 1. Thus, we omit to write it from the input.
- QFHE with classical ciphertexts $\mathsf{QFHE} = (\mathsf{QFHE.KG}, \mathsf{QFHE.Enc}, \mathsf{QFHE}.\mathcal{E}val, \mathsf{QFHE.Dec})$.

We set $\ell_{\mathsf{inp}} = \ell_{\mathsf{uk}} + \ell_{\mathsf{comr}}$. The description of CP is as follows.

$\mathcal{C}opy\mathcal{P}rotect(1^\lambda, f_y)$**:**

- Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$, $\mathsf{uk} \leftarrow \mathsf{UE.KG}(1^\lambda)$, and $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda)$.
- Generate $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
- Parse $y_{\mathsf{mask}} \| y_{\mathsf{comr}} \leftarrow y$ and generate $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{ck}, y_{\mathsf{mask}}; y_{\mathsf{comr}})$.
- Generate $\mathsf{ct}_{\mathsf{uk}} \leftarrow y_{\mathsf{mask}} \oplus \mathsf{uk}$.
- Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, (\mathsf{com}, \mathsf{ct}_{\mathsf{uk}}))$.
- Generate $\mathsf{ue}.ct \leftarrow \mathsf{UE}.\mathcal{E}nc(\mathsf{uk}, \mathsf{lock})$.
- Generate $\widetilde{D} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock})$.
- Output $\rho = (\mathsf{ck}, \mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D})$.

$\mathcal{E}val(\rho, x)$**:**

- Parse $\rho = (\mathsf{ck}, \mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D})$.
- Compute $\mathsf{evct} \leftarrow \mathsf{QFHE}.\mathcal{E}val(C[\mathsf{ck}, x], \mathsf{ue}.ct, \mathsf{qfhe.ct})$, where the circuit $C[\mathsf{ck}, x]$ is described in Figure 1.
- Output $y \leftarrow \widetilde{D}(\mathsf{evct})$.

Evaluation Correctness. It is easy to see that CP satisfies evaluation correctness from the correctness of CC.Obf, QFHE, and UE, and injectivity of Com.

Quantum circuit $C[\mathsf{ck}, x]$

Constants: Strings ck and x.

Input: A quantum state $\mathsf{ue}.ct$ and strings com and $\mathsf{ct}_{\mathsf{uk}}$.

1. Parse $x_{\mathsf{mask}} \| x_{\mathsf{comr}} \leftarrow x$.
2. If $\mathsf{com} \neq \mathsf{Commit}(\mathsf{ck}, x_{\mathsf{mask}}; x_{\mathsf{comr}})$, output 0^λ. Otherwise, go to the next step.
3. Compute $\mathsf{uk}' \leftarrow x_{\mathsf{mask}} \oplus \mathsf{ct}_{\mathsf{uk}}$.
4. Output $\mathsf{lock}' \leftarrow \mathsf{UE}.\mathcal{D}ec(\mathsf{uk}', \mathsf{ue}.ct)$.

Figure 1: The description of $C[\mathsf{ck}, x]$

Security. For security, we have the following theorems.

Theorem 5.5. *Let $0 \leq w \leq 1$. We define the distributions $U_{\ell_{\mathsf{inp}}}$ and $D_{w\text{-}\mathsf{resamp}}(\cdot)$ as follows.*

- *$U_{\ell_{\mathsf{inp}}}$ is the uniform distribution over $\{0,1\}^{\ell_{\mathsf{inp}}}$.*
- *$D_{w\text{-}\mathsf{resamp}}(\cdot)$ is a distribution such that $D_{w\text{-}\mathsf{resamp}}(y)$ outputs y with probability $1-w$ and outputs a resampled value $z \leftarrow U_{\ell_{\mathsf{inp}}}$ with probability w.*

Let $D = \{D_\lambda\}$ be a family of distributions where each D_λ outputs $(\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock}, aux := \mathsf{pk})$ generated as those in $\mathcal{C}opy\mathcal{P}rotect$. If $\mathsf{CC.Obf}$ is secure with respect to D, QFHE satisfies semantic security, UE satisfies one-out-of-many one-time unclonable-indistinguishable security, and Com satisfies trapdoor equivocality, then CP satisfies one-out-of-many copy protection security for $\mathcal{PF}^1$ with respect to the distributions $D_Y = U_{\ell_{\mathsf{inp}}}$ and $D_X(\cdot) = D_{w\text{-}\mathsf{resamp}}(\cdot)$.

Theorem 5.6. *We define $D^2_{w\text{-}\mathsf{resamp}}(\cdot)$ as follows.*

- *$D^2_{w\text{-}\mathsf{resamp}}(\cdot)$ is a distribution such that $D^2_{w\text{-}\mathsf{resamp}}(y)$ outputs (y,y) with probability $1-w$ and outputs (z,z) for a resampled value $z \leftarrow U_{\ell_{\mathsf{inp}}}$ with probability w.*

In Theorem 5.5, if we use one-time unclonable-indistinguishable secure UE, CP satisfies copy protection security for $\mathcal{PF}^1$ with respect to the distributions $D_Y = U_{\ell_{\mathsf{inp}}}$ and $D_X(\cdot) = D^2_{w\text{-}\mathsf{resamp}}(\cdot)$.

Remark 5.7 (On instantiations). When we instantiate CP based on Theorem 5.6, we need to be careful about the fact that the existing one-time unclonable-indistinguishable secure SKUE scheme uses QROM. The construction of CP evaluates the decryption circuit of UE by QFHE. Thus, to use the QROM based SKUE scheme as the building block of CP, we have to assume that it is secure when we replace QRO with real hash functions so that the decryption algorithm has a concrete description that QFHE can evaluate. Note that we always need this assumption to use QROM-based SKUE schemes in the real world.

When we instantiate CP based on Theorem 5.5, there is no such issue and we can obtain a construction secure in the standard model, since we have one-out-of-many one-time unclonable-indistinguishable secure SKUE based on the LWE assumption in the standard model.

We below prove Theorem 5.5 and omit the proof of Theorem 5.6. It is easy to see that we can similarly prove Theorem 5.6 by using one-time unclonable-indistinguishable security at the transition from Hyb_3 to Hyb_4 in the proof of Theorem 5.5.

Proof of Theorem 5.5. Let n be any polynomial of λ and $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$ any efficient adversary. We consider the case where $1-w \leq w$. The proof when $w \leq 1-w$ is similar. In this case, $p^{\mathtt{triv}} = w$ and our goal is to show that

$$\mathsf{Adv}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP}, D_Y, D_X, \mathcal{A}}(\lambda) = \frac{1}{n} \cdot 1 + \frac{n-1}{n} \cdot w + \mathsf{negl}(\lambda).$$

We prove it by using the following sequence of experiments.

Hyb_1: This is $\mathsf{Exp}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda,n)$ where $D_X(y) = D_{w\text{-}\mathsf{resamp}}(y)$ outputs $x = y$ without doing re-sampling, except that the output of the experiment is set to the adversary's output.

1. The challenger generates $y \leftarrow U_{\ell_{\mathsf{inp}}}$. The challenger sends ρ generated as follows to $\mathcal{A}_0$.
   - Generate $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$, $\mathsf{uk} \leftarrow \mathsf{UE.KG}(1^\lambda)$, and $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda)$.
   - Generate $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
   - Parse $y_{\mathsf{mask}} \| y_{\mathsf{comr}} \leftarrow y$ and generate $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{ck}, y_{\mathsf{mask}}; y_{\mathsf{comr}})$.
   - Generate $\mathsf{ct}_{\mathsf{uk}} \leftarrow y_{\mathsf{mask}} \oplus \mathsf{uk}$.
   - Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, (\mathsf{com}, \mathsf{ct}_{\mathsf{uk}}))$.
   - Generate $\mathsf{ue}.ct \leftarrow \mathsf{UE}.\mathcal{E}nc(\mathsf{uk}, \mathsf{lock})$.
   - Generate $\widetilde{D} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock})$.
   - Set $\rho = (\mathsf{ck}, \mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D})$.
2. $\mathcal{A}_0$ creates a quantum state q over n registers $\mathsf{R}_1, \cdots, \mathsf{R}_n$. $\mathcal{A}_0$ sends $q[\mathsf{R}_i]$ to $\mathcal{A}_i$ for every $i \in [n]$.
3. The challenger generates $\alpha \leftarrow [n]$. The challenger sends $x = y$ to $\mathcal{A}_\alpha$. $\mathcal{A}_\alpha$ outputs b_α. The output of the experiment is b_α.

Hyb_2: This is the same as Hyb_1 except that the tuple $(\mathsf{ck}, \mathsf{com}, y_{\mathsf{comr}})$ is generated as $(\mathsf{ck}, \mathsf{com}, \mathsf{td}) \leftarrow \mathsf{EqSetup}(1^\lambda)$ and $y_{\mathsf{comr}} \leftarrow \mathsf{Open}(\mathsf{td}, y_{\mathsf{mask}}, \mathsf{com})$.

From the trapdoor equivocation property of Com, we have $|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_3: This is the same as Hyb_2 except that y_{mask} is replaced with $y_{\mathsf{mask}} \oplus \mathsf{uk}$. By this change, $\mathsf{qfhe.ct}$ and y_{comr} are generated as $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, (\mathsf{com}, y_{\mathsf{mask}}))$ and $y_{\mathsf{comr}} \leftarrow \mathsf{Open}(\mathsf{td}, y_{\mathsf{mask}} \oplus \mathsf{uk}, \mathsf{com})$. Moreover, $\mathcal{A}_\alpha$ is given $x = (y_{\mathsf{mask}} \oplus \mathsf{uk}) \| y_{\mathsf{comr}}$.

We have $\Pr[\mathsf{Hyb}_2 = 1] = \Pr[\mathsf{Hyb}_3 = 1]$.

Hyb_4: This is the same as Hyb_3 except that $\mathsf{ue}.ct$ is generated as $\mathsf{ue}.ct \leftarrow \mathsf{UE}.\mathcal{E}nc(\mathsf{uk}, 0^\lambda)$.

We consider the following adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1, \cdots, \mathcal{B}_n)$ that attacks the one-out-of-many one-time unclonable-indistinguishable security of UE. $\mathcal{B}_0$ behaves as follows.

1. $\mathcal{B}_0$ sends $(\mathsf{lock}, 0^\lambda)$ to the challenger, where $\mathsf{lock} \leftarrow \{0,1\}^\lambda$ and obtains $\mathsf{ue}.ct$ from the challenger. Then $\mathcal{B}_0$ sends ρ generated as follows to $\mathcal{A}_0$.
   - Generate $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$ and $(\mathsf{ck}, \mathsf{com}, \mathsf{td}) \leftarrow \mathsf{EqSetup}(1^\lambda)$.
   - $y_{\mathsf{mask}} \leftarrow \{0,1\}^{\ell_{\mathsf{uk}}}$.
   - Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, (\mathsf{com}, y_{\mathsf{mask}}))$.
   - Generate $\widetilde{D} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock})$.
   - Set $\rho = (\mathsf{ck}, \mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D})$.
2. When $\mathcal{A}_0$ creates a quantum state q over n registers $\mathsf{R}_1, \cdots, \mathsf{R}_n$, $\mathcal{B}_0$ sends $(\mathsf{com}, \mathsf{td}, y_{\mathsf{mask}}, q[\mathsf{R}_i])$ to $\mathcal{B}_i$ for every $i \in [n]$.

$\mathcal{B}_\alpha$ behaves as follows, where $\alpha \leftarrow [n]$ is chosen by the challenger.

1. $\mathcal{B}_\alpha$ send $q[\mathsf{R}_\alpha]$ to $\mathcal{A}_\alpha$.
2. When $\mathcal{B}_\alpha$ obtains uk from the challenger, it computes $y_{\mathsf{comr}} \leftarrow \mathsf{Open}(\mathsf{td}, y_{\mathsf{mask}} \oplus \mathsf{uk}, \mathsf{com})$ and sends $x = y_{\mathsf{mask}} \oplus \mathsf{uk} \| y_{\mathsf{comr}}$ to $\mathcal{A}_\alpha$.
3. When $\mathcal{A}_\alpha$ outputs b_α, it outputs $\mathsf{coin}_\alpha = b_\alpha$.

Let the challenge bit in the security game played by $\mathcal{B}$ be coin. If $\mathsf{coin} = 0$, $\mathcal{B}$ perfectly simulates Hyb_3 to $\mathcal{A}$. If $\mathsf{coin} = 1$, $\mathcal{B}$ perfectly simulates Hyb_4 to $\mathcal{A}$. Also, $\mathcal{B}$ outputs $\mathcal{A}$'s output. Then, we have

$$\begin{aligned}\Pr[\mathsf{coin}_\alpha = \mathsf{coin}] - \frac{1}{2} &= \frac{1}{2}(\Pr[\mathsf{coin}_\alpha = 1 \mid \mathsf{coin} = 0] - \Pr[\mathsf{coin}_\alpha = 1 \mid \mathsf{coin} = 1]) \\ &= \frac{1}{2}(\Pr[\mathsf{Hyb}_3 = 1] - \Pr[\mathsf{Hyb}_4 = 1]).\end{aligned}$$

Thus, from the one-out-of-many one-time unclonable-indistinguishable security of UE, we have $\frac{1}{2}(\Pr[\mathsf{Hyb}_3 = 1] - \Pr[\mathsf{Hyb}_4 = 1]) \leq \frac{1}{2n} + \mathsf{negl}(\lambda)$.

Hyb_5**:** This is the same as Hyb_4 except that $\widetilde{D}$ is generated as $\widetilde{D} \leftarrow \mathsf{CC.Sim}(1^\lambda, \mathsf{pp}_{\mathsf{QFHE.Dec}})$, where $\mathsf{pp}_{\mathsf{QFHE.Dec}}$ is the parameters of QFHE.Dec.

From the security of CC.Obf, we have $|\Pr[\mathsf{Hyb}_4 = 1] - \Pr[\mathsf{Hyb}_5 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_6**:** This is the same as Hyb_5 except that qfhe.ct is generated as $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, 0^L)$, where $L = |\mathsf{com}| + \ell_{\mathsf{uk}}$.

From the security of QFHE, we have $|\Pr[\mathsf{Hyb}_5 = 1] - \Pr[\mathsf{Hyb}_6 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_7**:** This is the same as Hyb_6 except that y_{mask} is replaced with $y_{\mathsf{mask}} \oplus \mathsf{uk}$. By this change, y_{comr} is generated as $y_{\mathsf{comr}} \leftarrow \mathsf{Open}(\mathsf{ck}, y_{\mathsf{mask}}, \mathsf{com})$. Moreover, $\mathcal{A}_\alpha$ is given $y := y_{\mathsf{mask}} \| y_{\mathsf{comr}}$.

We have $\Pr[\mathsf{Hyb}_6 = 1] = \Pr[\mathsf{Hyb}_7 = 1]$.

Hyb_8**:** This is the same as Hyb_7 except that ck is generated as $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda)$ and y_{comr} is generated uniformly at random.

From the trapdoor equivocation property of Com, we have $|\Pr[\mathsf{Hyb}_7 = 1] - \Pr[\mathsf{Hyb}_8 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_9**:** This is the same as Hyb_8 except that a re-sampled value $x \leftarrow U_{\ell_{\mathsf{inp}}}$ is given to $\mathcal{A}_\alpha$ instead of $y = y_{\mathsf{mask}} \| y_{\mathsf{comr}}$.

In Hyb_8, ρ given to $\mathcal{A}_0$ is independent of $y = y_{\mathsf{mask}} \| y_{\mathsf{comr}}$ and y is uniformly at random, and thus, we have $\Pr[\mathsf{Hyb}_8 = 1] = \Pr[\mathsf{Hyb}_9 = 1]$.

Hyb_{10}**:** This is the same as Hyb_9 except that we generate ρ in the same way as Hyb_1.

We can prove $|\Pr[\mathsf{Hyb}_9 = 1] - \Pr[\mathsf{Hyb}_{10} = 1]| = \mathsf{negl}(\lambda)$ by using the security of CC.Obf, QFHE, UE, and Com, essentially undoing the changes between Hyb_1 and Hyb_9. To make this change, we can rely on one-time IND-CPA security of UE, not one-out-of-many one-time unclonable-indistinguishable security for the following reason. In Hyb_9 and Hyb_{10}, $\mathcal{A}_\alpha$ is given a re-sample value x and not $y = y_{\mathsf{mask}} \| y_{\mathsf{comr}}$. Then, we can ensure that the information of uk is not given to $\mathcal{A}$ except ue.ct in this transition, which allows us to use one-time IND-CPA security of UE. Note that the one-time IND-CPA security of UE is implied by the one-out-of-many one-time unclonable-indistinguishable security of UE as proven in Theorem 3.2.

Hyb_{10} is $\mathsf{Exp}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},D_Y,D_X,\mathcal{A}}(\lambda, n)$ where $D_X(y) = D_{w\text{-}\mathsf{resamp}}(y)$ outputs a resampled $x \leftarrow U_{\ell_{\mathsf{inp}}}$ and the output is set to $\mathcal{A}_\alpha$'s output. Let Resamp be the event that $D_X(y) = D_{w\text{-}\mathsf{resamp}}(y)$ does re-sampling. Then, we have

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},\mathcal{A}}(\lambda) - w \\ &\leq (1-w) \cdot \Pr[b_\alpha = 1 \mid \neg\mathtt{Resamp}] + w \cdot \Pr[b_\alpha = 0 \mid \mathtt{Resamp}] + \mathsf{negl}(\lambda) - w \\ &= (1-w) \cdot \Pr[b_\alpha = 1 \mid \neg\mathtt{Resamp}] - w \cdot (1 - \Pr[b_\alpha = 0 \mid \mathtt{Resamp}]) + \mathsf{negl}(\lambda) \\ &\leq (1-w) \cdot \Pr[b_\alpha = 1 \mid \neg\mathtt{Resamp}] - (1-w) \cdot \Pr[b_\alpha = 1 \mid \mathtt{Resamp}] + \mathsf{negl}(\lambda) \\ &\leq (1-w) \cdot (\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_{10} = 1]) + \mathsf{negl}(\lambda).\end{aligned}$$

The third inequality uses $1 - w \leq w$. From the above discussions, we have $\Pr[\mathsf{Hyb}_1] - \Pr[\mathsf{Hyb}_{10}] \leq \frac{1}{n} + \mathsf{negl}(\lambda)$. Therefore, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cp\text{-}pf1\text{-}om}}_{\mathsf{CP},\mathcal{A}}(\lambda) &\leq w + (1-w)\cdot\frac{1}{n} + \mathsf{negl}(\lambda)\\ &= \frac{1}{n}\cdot 1 + \frac{n-1}{n}\cdot w + \mathsf{negl}(\lambda).\end{aligned}$$

This completes the proof. □

6 Unclonable Predicate Encryption

We introduce unclonable predicate encryption (PE) and present how to achieve it in this section.

6.1 Definition

The definition of unclonable PE is a natural extension of PE to an unclonable variant.

Definition 6.1 (Unclonable Predicate Encryption (Syntax)). *An unclonable predicate encryption scheme is a tuple of quantum algorithms* $(\mathsf{Setup}, \mathsf{KG}, \mathcal{Enc}, \mathcal{Dec})$ *with plaintext space* $\mathcal{M}$.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** *The setup algorithm takes as input the security parameter* 1^λ *and outputs a key pair* $(\mathsf{mpk}, \mathsf{msk})$.

$\mathsf{KG}(\mathsf{msk}, x) \to \mathsf{sk}_x$**:** *The key generation algorithm takes as input the master secret key* msk *and an attribute* x*, and outputs a decryption key* sk_x.

$\mathcal{Enc}(\mathsf{mpk}, C, \mathsf{m}) \to ct$**:** *The encryption algorithm takes as input* mpk*, predicate* C*, and a plaintext* $\mathsf{m} \in \mathcal{M}$*, and outputs a ciphertext* ct.

$\mathcal{Dec}(\mathsf{sk}_x, ct) \to \mathsf{m}'$ **or** $\perp$**:** *The decryption algorithm takes as input* sk_x *and* ct *and outputs a plaintext* m' *or* $\perp$.

Decryption Correctness: *There exists a negligible function* negl *such that for any* $\mathsf{m} \in \mathcal{M}$*,* C*, and* x *such that* $C(x) = 1$*, we have*

$$\Pr\left[\mathcal{Dec}(\mathsf{sk}_x, ct) = \mathsf{m} \;\middle|\; \begin{array}{l}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)\\ \mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)\\ ct \leftarrow \mathcal{Enc}(\mathsf{mpk}, C, \mathsf{m})\end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

We define simulation-based security for unclonable PE, and then discuss its validity.

Definition 6.2 (Adaptive Unclonable-Simulation Security for PE). *Let* $\mathsf{UPE} = (\mathsf{Setup}, \mathsf{KG}, \mathcal{Enc}, \mathcal{Dec})$ *be an unclonable predicate encryption scheme. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathcal{Sim},\mathcal{A}}(\lambda)$*, where* $\mathcal{Sim}$ *is a QPT simulation algorithm and* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$.

1. *The challenger generates* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and sends* mpk *to* $\mathcal{A}_0$.
2. $\mathcal{A}_0$ *can get access to the following oracle.*

   $O_{\mathsf{KG},1}(x)$**:** *Given* x*, it returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$.
3. $\mathcal{A}_0$ *sends* C *and* $\mathsf{m} \in \mathcal{M}$ *to thfe challenger, where* C *satisfies* $C(x) = 0$ *for all* x *queried by* $\mathcal{A}_0$ *in the previous step. The challenger picks* $\mathsf{coin} \leftarrow \{0,1\}$ *and does the following.*
   - *If* $\mathsf{coin} = 0$*, the challenger generates* $ct \leftarrow \mathcal{Enc}(\mathsf{mpk}, C, \mathsf{m})$ *and returns* ct *to* $\mathcal{A}_0$.
   - *If* $\mathsf{coin} = 1$*, the challenger generates* $ct \leftarrow \mathcal{Sim}(1^\lambda, |C|, |\mathsf{m}|)$ *and returns* ct *to* $\mathcal{A}_0$.

Hereafter, $\mathcal{A}_0$ is not allowed to query x such that $C(x) = 1$ to $O_{\mathsf{KG},1}$.

4. *$\mathcal{A}_0$ creates a bipartite state q over registers R_1 and R_2. $\mathcal{A}_0$ sends $q[\mathsf{R}_1]$ and $q[\mathsf{R}_2]$ to $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.*
5. *$\mathcal{A}_1$ and $\mathcal{A}_2$ can get access to the following oracle.*

   $O_{\mathsf{KG},2}(x)$**:** *Given x, it returns $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$. Note that $O_{\mathsf{KG},2}$ accepts a query x such that $C(x) = 1$.*
6. *$\mathcal{A}_1$ and $\mathcal{A}_2$ respectively output coin'_1 and coin'_2. If $\mathsf{coin}'_i = \mathsf{coin}$ for $i \in \{1,2\}$, the challenger outputs 1, otherwise outputs 0.*

We say that UPE *is unclonable-simulation secure if there exists QPT $\mathcal{Sim}$ such that for any QPT $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathcal{Sim},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathcal{A}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Remark 6.3. We select a simulation-based security definition since it intuitively captures security of encryption and stronger than indistinguishability-based one. There are impossibility results of simulation-based secure FE [BSW11, AGVW13]. However, those are not applied to our setting since we consider the single challenge ciphertext setting of PE, where the message (a.k.a payload) part is recovered by a secret key.

Remark 6.4 (On the validity of unclonable-simulation security). We claim that our unclonable-simulation security for PE captures unclonability of both the payload part and the predicate part. To this end, we first argue that by using both the payload part and the predicate part, we can realize SKUE satisfying unclonable-simulation security where $\mathcal{A}_0$ is given a real or simulated challenge ciphertext in the security game. We then discuss about the validity of unclonable-simulation security for SKUE.

It is rather clear that we can achieve unclonable-simulation secure SKUE using the payload part. Then, we show how to construct an unclonable-simulation secure SKUE scheme for the message space $\{1,...N\}$ using the predicate part, where N is a polynomial in λ. Let $C[i]$ be a predicate that takes as input j and output 1 if and only if $i = j$. In this construction, when we encrypt a message i^*, we generate a ciphertext of the payload $\mathsf{m} = 1$ with the predicate $C[i^*]$ by the unclonable PE scheme. The decryption key of this construction is $(\mathsf{sk}_1, \ldots, \mathsf{sk}_N)$, where sk_j is the decryption key for the attribute j. Decryption is done by testing if a ciphertext can be decrypted by sk_j for every j. The unclonable-simulation security of this SKUE scheme is reduced to that of the unclonable PE scheme.

We now discuss the validity of unclonable-simulation security of SKUE. Although we do not see the formal implication from unclonable-simulation security to unclonable-indistinguishable security, we can use any SKUE scheme with the former security notion as that with the latter security notion whose message space is $\{0,1\}$. This is done by encoding 1-bit messages using real ciphertext and simulated ciphertext. Also, we can formally prove that unclonable-simulation security is strictly stronger than one-wayness-based unclonability, similarly to unclonable-indistinguishability. In the proof of the implication from unclonable-indistinguishability to one-wayness-based unclonability, the argument goes though by using the former to switch a real ciphertext into a junk ciphertext (such as a ciphertext of 0). A similar argument can be done by using the power of simulation-based security that is used to switch a real ciphertext into a simulated one. In general, we can use unclonable-simulation security as a drop-in replacement of unclonable-indistinguishable security, if the latter is used to switch a real ciphertext into a junk one. For example, this is the case in our construction of copy-protection for single-bit output point functions presented in Section 5, which proves the usefulness of unclonable-simulation security.

We propose one-out-of-many variant of unclonable-simulation security for PE.

Definition 6.5 (One-out-of-Many Adaptive Unclonable-Simulation Security for PE). *Let* $\mathsf{UPE} = (\mathsf{Setup}, \mathsf{KG}, \mathcal{Enc}, \mathcal{Dec})$ *be an unclonable PE scheme. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathcal{Sim},\mathcal{A}}(\lambda, n)$*, where $\mathcal{Sim}$ is a QPT simulation algorithm and $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$.*

1. *The challenger generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends mpk to $\mathcal{A}_0$.*
2. *$\mathcal{A}_0$ can get access to the following oracle.*

$O_{\mathsf{KG},1}(x)$: *Given x, it returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$.

3. $\mathcal{A}_0$ *sends* C *and* $\mathsf{m} \in \mathcal{M}$ *to thfe challenger, where* C *satisfies* $C(x) = 0$ *for all* x *queried by* $\mathcal{A}_0$ *in the previous step. The challenger picks* $\mathsf{coin} \leftarrow \{0,1\}$ *and does the following.*
   - *If* $\mathsf{coin} = 0$, *the challenger generates* $ct \leftarrow \mathcal{E}nc(\mathsf{mpk}, C, \mathsf{m})$ *and returns* ct *to* $\mathcal{A}_0$.
   - *If* $\mathsf{coin} = 1$, *the challenger generates* $ct \leftarrow \mathcal{S}im(1^\lambda, |C|, |\mathsf{m}|)$ *and returns* ct *to* $\mathcal{A}_0$.

   Hereafter, $\mathcal{A}_0$ *is not allowed to query* x *such that* $C(x) = 1$ *to* $O_{\mathsf{KG},1}$.
4. $\mathcal{A}_0$ *creates a quantum state* q *over* n *registers* $\mathsf{R}_1, \cdots, \mathsf{R}_n$. $\mathcal{A}_0$ *sends* $q[\mathsf{R}_i]$ *to* $\mathcal{A}_i$ *for every* $i \in [n]$.
5. *The challenger generates* $\alpha \leftarrow [n]$. $\mathcal{A}_\alpha$ *can access the following oracle.*

   $O_{\mathsf{KG},2}(x)$: *Given* x, *it returns* $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$. *Note that* $O_{\mathsf{KG},2}$ *accepts a query* x *such that* $C(x) = 1$.
6. $\mathcal{A}_\alpha$ *outputs* coin'. *If* $\mathsf{coin}' = \mathsf{coin}$, *the challenger outputs* 1, *otherwise outputs* 0.

We say that UPE *is one-out-of-many unclonable-simulation secure if there exists QPT* $\mathcal{S}im$ *such that for any polynomial* $n = n(\lambda)$ *and QPT* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$, *it holds that*

$$\mathsf{Adv}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathcal{S}im,\mathcal{A}}(\lambda, n) := \Pr\left[\mathsf{Exp}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathcal{A}}(\lambda, n) = 1\right] \leq \frac{1}{2} + \frac{1}{2n} + \mathsf{negl}(\lambda).$$

Remark 6.6. We can consider selective variants of Definitions 6.2 and 6.5, where $\mathcal{A}_0$ declares C at the beginning of the games. We call selective unclonable-simulation security and one-out-of-many selective unclonable-simulation security, respectively.

6.2 Construction

We construct an unclonable PE scheme UPE using the following tools.

- SKUE $\mathsf{UE} = (\mathsf{UE.KG}, \mathsf{UE}.\mathcal{E}nc, \mathsf{UE}.\mathcal{D}ec)$. Suppose the plaintext space of UE is $\{0,1\}^\lambda$.
- Compute-and-compare obfuscation CC.Obf with the simulator CC.Sim.
- QFHE with classical ciphertexts $\mathsf{QFHE} = (\mathsf{QFHE.KG}, \mathsf{QFHE.Enc}, \mathsf{QFHE}.\mathcal{E}val, \mathsf{QFHE.Dec})$.
- AD-SIM secure CP-ABE $\mathsf{CPABE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with a QPT simulator $\mathcal{ABE}Sim = (\mathcal{S}im\mathcal{E}nc, \mathcal{S}im\mathcal{KG})$.

The description of UPE is as follows.

$\mathsf{UPE.Setup}(1^\lambda)$:

- Output $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.

$\mathsf{UPE.KG}(\mathsf{msk}, x)$:

- Output $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$.

$\mathsf{UPE.Enc}(\mathsf{mpk}, C, \mathsf{m})$:

- Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$ and $\mathsf{uk} \leftarrow \mathsf{UE.KG}(1^\lambda)$.
- Generate $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
- Generate $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, C, \mathsf{uk})$.
- Let $D(\cdot)$ be the decryption circuit $\mathsf{Dec}(\cdot, \mathsf{ct})$ of CPABE that has ct hardwired.
- Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, D)$.

Quantum circuit $C[\mathsf{sk}_x]$

Constants: A string sk_x.

Input: A quantum state $\mathsf{ue}.ct$ and a circuit description D.

1. Compute $\mathsf{uk}' \leftarrow D(\mathsf{sk}_x)$.
2. Output $\mathsf{lock}' \leftarrow \mathsf{UE}.\mathcal{D}ec(\mathsf{uk}', \mathsf{ue}.ct)$.

Figure 2: The description of $C[\mathsf{sk}_x]$

- Generate $\mathsf{ue}.ct \leftarrow \mathsf{UE}.\mathcal{E}nc(\mathsf{uk}, \mathsf{lock})$.
- Generate $\widetilde{D}_{\mathsf{qfhe}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock}, \mathsf{m})$.
- Output $\mathsf{upe}.ct \leftarrow (\mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D}_{\mathsf{qfhe}})$.

$\mathsf{UPE.Dec}(\mathsf{sk}_x, \mathsf{upe}.ct)$**:**

- Parse $\rho = (\mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D}_{\mathsf{qfhe}})$.
- Compute $\mathsf{evct} \leftarrow \mathsf{QFHE}.\mathcal{E}val(C[\mathsf{sk}_x], \mathsf{ue}.ct, \mathsf{qfhe.ct})$, where the circuit $C[\mathsf{sk}_x]$ is described in Figure 2.
- Output $\mathsf{m} \leftarrow \widetilde{D}_{\mathsf{qfhe}}(\mathsf{evct})$.

The correctness of UPE immediately follows from the correctness of the building blocks.

Security. We have the following theorems.

Theorem 6.7. *Let $D = \{D_\lambda\}$ be a family of distributions where each D_λ outputs $(\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock}, aux := \mathsf{pk})$ generated as those in* UPE.Enc*. If* CC.Obf *is secure with respect to D,* QFHE *satisfies semantic security,* UE *satisfies one-out-of-many one-time unclonable-indistinguishable security, and* CPABE *is AD-SIM secure, then* UPE *satisfies one-out-of-many adaptive unclonable-simulation security.*

Theorem 6.8. *In Theorem 6.7, if we use a one-time unclonable-indistinguishable secure SKFE scheme* UE*, then* UPE *satisfies adaptive unclonable-simulation security.*

Similarly to our copy-protection scheme presented in Section 5, when we instantiate UPE based on Theorem 6.8, we need to be careful about the use of QRO by the existing SKUE schemes. See Remark 5.7.

If our goal is selective unclonable-simulation security or one-out-of-many selective unclonable-simulation security, we can use SEL-SIM secure CP-ABE instead of AD-SIM secure CP-ABE in Theorems 6.7 and 6.8.

We below prove Theorem 6.7 and omit the proof of Theorem 6.8. It is easy to see that we can similarly prove Theorem 6.8 by using one-time unclonable-indistinguishable security at the transition from Hyb_2 to Hyb_3 in the proof of Theorem 6.7.

Proof of Theorem 6.7. Let n be any polynomial of λ and $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_n)$ any efficient adversary. Also, let $\mathsf{UPE}.\mathcal{S}im$ be the following algorithm.

$\mathsf{UPE}.\mathcal{S}im(1^\lambda, |C|, |\mathsf{m}|)$**:**

- Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$ and $\mathsf{uk} \leftarrow \mathsf{UE.KG}(1^\lambda)$.
- Generate $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
- Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, 0^L)$, where L is the size of CPABE's decryption circuit $\mathsf{Dec}(\cdot, \mathsf{ct})$ that has hardwired ciphertext of a λ-bit message with a $|C|$-bit predicate.
- Generate $\mathsf{ue}.ct \leftarrow \mathcal{E}nc(\mathsf{uk}, 0^\lambda)$.
- Generate $\widetilde{D}_{\mathsf{qfhe}} \leftarrow \mathsf{CC.Sim}(1^\lambda, \mathsf{pp}_{\mathsf{QFHE.Dec}}, |\mathsf{m}|)$, where $\mathsf{pp}_{\mathsf{QFHE.Dec}}$ is the parameters of QFHE.Dec.

- Output $\mathsf{upe}.ct \leftarrow (\mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D}_{\mathsf{qfhe}})$.

We prove $\mathsf{Adv}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathsf{UPE}.\mathcal{Sim},\mathcal{A}}(\lambda, n) \leq \frac{1}{2} + \frac{1}{2n} + \mathsf{negl}(\lambda)$ for any polynomial n and QPT $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \cdots, \mathcal{A}_n)$ using the following sequence of experiments.

Hyb_1**:** This is $\mathsf{Exp}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathsf{UPE}.\mathcal{Sim},\mathcal{A}}(\lambda, n)$ where $\mathsf{coin} = 0$ and the output of the experiment is set to the adversary's output. The detailed description is as follows.

1. The challenger generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends mpk to $\mathcal{A}_0$.
2. $\mathcal{A}_0$ can get access to the following oracle.

   $O_{\mathsf{KG},1}(x)$**:** Given x, it returns $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$.
3. $\mathcal{A}_0$ sends C and $\mathsf{m} \in \mathcal{M}$ to the challenger, where C satisfies $C(x) = 0$ for all x queried by $\mathcal{A}_0$ in the previous step. The challenger returns $\mathsf{upe}.ct$ generated as follows.
   - Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$ and $\mathsf{uk} \leftarrow \mathsf{UE.KG}(1^\lambda)$.
   - Generate $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
   - Generate $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, C, \mathsf{uk})$.
   - Let $D(\cdot)$ be the decryption circuit $\mathsf{Dec}(\cdot, \mathsf{ct})$ of CPABE that has ct hardwired.
   - Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, D)$.
   - Generate $\mathsf{ue}.ct \leftarrow \mathcal{Enc}(\mathsf{uk}, \mathsf{lock})$.
   - Generate $\widetilde{D}_{\mathsf{qfhe}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock}, \mathsf{m})$.
   - set $\mathsf{upe}.ct \leftarrow (\mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D}_{\mathsf{qfhe}})$.

   Hereafter, $\mathcal{A}_0$ is not allowed to query x such that $C(x) = 1$ to $O_{\mathsf{KG},1}$.
4. $\mathcal{A}_0$ creates a quantum state q over n registers $\mathsf{R}_1, \ldots, \mathsf{R}_n$. $\mathcal{A}_0$ sends $q[\mathsf{R}_i]$ to $\mathcal{A}_i$ for every $i \in [n]$.
5. The challenger generates $\alpha \leftarrow [n]$. $\mathcal{A}_\alpha$ can get access to the following oracle.

   $O_{\mathsf{KG},2}(x)$**:** Given x, it returns $\mathsf{sk}_x \leftarrow \mathsf{KG}(\mathsf{msk}, x)$. Note that $O_{\mathsf{KG},2}$ accepts a query x such that $C(x) = 1$.
6. $\mathcal{A}_\alpha$ outputs coin'_α. The final output of the experiment is coin'_α.

Hyb_2**:** This is the same as Hyb_1 except the following changes.

- ct is generated as $(\mathsf{ct}, \mathsf{st}) \leftarrow \mathcal{SimEnc}(\mathsf{mpk}, C)$ instead of $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, C, \mathsf{uk})$.
- $O_{\mathsf{KG},2}$ returns $\mathsf{sk}_x \leftarrow \mathcal{SimKG}(\mathsf{msk}, \mathsf{st}, x, \mathsf{uk})$ given x if $C(x) = 1$.

From the AD-SIM security of CPABE, we have $|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_3**:** This is the same as Hyb_2 except that $\mathsf{ue}.ct$ is generated as $\mathsf{ue}.ct \leftarrow \mathsf{UE}.\mathcal{Enc}(\mathsf{uk}, 0^\lambda)$.

We consider the following adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1, \cdots, \mathcal{B}_n)$ that attacks the one-out-of-many one-time unclonable-indistinguishable security of UE. $\mathcal{B}_0$ behaves as follows.

1. $\mathcal{B}_0$ generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends mpk to $\mathcal{A}_0$.
2. $\mathcal{B}_0$ simulates $O_{\mathsf{KG},1}(x)$ for $\mathcal{A}_0$ using msk.
3. When $\mathcal{A}_0$ outputs C and $\mathsf{m} \in \mathcal{M}$, where C satisfies $C(x) = 0$ for all x queried by $\mathcal{A}_0$ in the previous step, $\mathcal{B}_0$ outputs $(\mathsf{lock}, 0^\lambda)$, obtains $\mathsf{ue}.ct$, and returns $\mathsf{upe}.ct$ generated as follows, where $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
   - Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{QFHE.KG}(1^\lambda)$ and $\mathsf{uk} \leftarrow \mathsf{UE.KG}(1^\lambda)$.
   - Generate $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
   - Generate $(\mathsf{ct}, \mathsf{st}) \leftarrow \mathcal{SimEnc}(\mathsf{mpk}, C)$.
   - Let $D(\cdot)$ be the decryption circuit $\mathsf{Dec}(\cdot, \mathsf{ct})$ of CPABE that has ct hardwired.

- Generate $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, D)$.
- Generate $\widetilde{D}_{\mathsf{qfhe}} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{QFHE.Dec}(\mathsf{sk}, \cdot), \mathsf{lock}, \mathsf{m})$.
- set $\mathsf{upe}.ct \leftarrow (\mathsf{qfhe.ct}, \mathsf{ue}.ct, \widetilde{D}_{\mathsf{qfhe}})$.

Hereafter, $\mathcal{A}_0$ is not allowed to query x such that $C(x) = 1$ to $O_{\mathsf{KG},1}$.

4. When $\mathcal{A}_0$ outputs a quantum state q over n registers $\mathsf{R}_1, \dots, \mathsf{R}_n$, $\mathcal{B}_0$ sends $(\mathsf{msk}, \mathsf{st}, q[\mathsf{R}_i])$ to $\mathcal{B}_i$ for every $i \in [n]$.

$\mathcal{B}_\alpha$ behaves as follows, where $\alpha \leftarrow [n]$ is chosen by the challenger.

1. Given uk as an input, $\mathcal{B}_\alpha$ first send $q[\mathsf{R}_\alpha]$ to $\mathcal{A}_\alpha$.
2. $\mathcal{B}_\alpha$ simulates $O_{\mathsf{KG},2}(x)$ for $\mathcal{A}_\alpha$ as follows.

   $O_{\mathsf{KG},2}(x)$**:** Given x, $\mathcal{B}_\alpha$ returns $\mathsf{sk}_x \leftarrow \mathcal{SIMKG}(\mathsf{msk}, \mathsf{st}, x, \mathsf{uk})$.
3. When $\mathcal{A}_\alpha$ outputs coin'_α, $\mathcal{B}_\alpha$ outputs $b' = \mathsf{coin}'_\alpha$.

Let the challenge bit in the security game played by $\mathcal{B}$ be b. If $b = 0$, $\mathcal{B}$ perfectly simulates Hyb_2 to $\mathcal{A}$. If $b = 1$, $\mathcal{B}$ perfectly simulates Hyb_3 to $\mathcal{A}$. Also, $\mathcal{B}$ outputs $\mathcal{A}$'s output. Then, we have

$$\begin{aligned}\Pr\left[b' = b\right] - \frac{1}{2} &= \frac{1}{2}(\Pr\left[b' = 1 | b = 0\right] - \Pr\left[b' = 1 | b = 1\right]) \\ &= \frac{1}{2}(\Pr[\mathsf{Hyb}_2 = 1] - \Pr[\mathsf{Hyb}_3 = 1]).\end{aligned}$$

Thus, from the one-out-of-many one-time unclonable-indistinguishable security of UE, we have $\frac{1}{2}(\Pr[\mathsf{Hyb}_2 = 1] - \Pr[\mathsf{Hyb}_3 = 1]) \leq \frac{1}{2n} + \mathsf{negl}(\lambda)$.

Hyb_4**:** This is the same as Hyb_3 except that $\widetilde{D}_{\mathsf{qfhe}}$ is generated as $\widetilde{D}_{\mathsf{qfhe}} \leftarrow \mathsf{CC.Sim}(1^\lambda, \mathsf{pp}_{\mathsf{QFHE.Dec}}, |\mathsf{m}|)$, where $\mathsf{pp}_{\mathsf{QFHE.Dec}}$ is the parameters of $\mathsf{QFHE.Dec}$.

From the security of $\mathsf{CC.Obf}$, we have $|\Pr[\mathsf{Hyb}_3 = 1] - \Pr[\mathsf{Hyb}_4 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_5**:** This is the same as Hyb_4 except that $\mathsf{qfhe.ct}$ is generated as $\mathsf{qfhe.ct} \leftarrow \mathsf{QFHE.Enc}(\mathsf{pk}, 0^L)$, where L is the size of CPABE's decryption circuit $\mathsf{Dec}(\cdot, \mathsf{ct})$ that has hardwired ciphertext of a λ-bit message with a $|C|$-bit predicate.

From the security of QFHE, we have $|\Pr[\mathsf{Hyb}_4 = 1] - \Pr[\mathsf{Hyb}_5 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_5 is $\mathsf{Exp}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathsf{UPE}.\mathcal{Sim},\mathcal{A}}(\lambda, n)$ where $\mathsf{coin} = 1$ and the output of the experiment is set to $\mathcal{A}_\alpha$'s output. Also, we have

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathsf{UPE}.\mathcal{Sim},\mathcal{A}}(\lambda, n) - \frac{1}{2} \\ &= \frac{1}{2}(\Pr\left[\mathsf{coin}' = 1 | \mathsf{coin} = 0\right] - \Pr\left[\mathsf{coin}' = 1 | \mathsf{coin} = 1\right]) \\ &= \frac{1}{2}(\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_5 = 1]).\end{aligned}$$

From the above discussions, we have $\frac{1}{2}(\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_5 = 1]) \leq \frac{1}{2n} + \mathsf{negl}(\lambda)$, which means that $\mathsf{Adv}^{\mathsf{om\text{-}ada\text{-}sim\text{-}clone}}_{\mathsf{UPE},\mathsf{UPE}.\mathcal{Sim},\mathcal{A}}(\lambda, n) \leq \frac{1}{2} + \frac{1}{2n} + \mathsf{negl}(\lambda)$. This completes the proof. □

Remark 6.9. We can also consider a setting where $\mathcal{A}_\alpha$ receives the master secret key msk of unclonable PE. If we use IO, we can achieve the stronger definition. The non-committing ABE scheme based on IO by Hiroka et al. [HMNY21] achieves stronger security where the adversary is given a master secret key after a challenge ciphertext is given. If we use their scheme instead of our simulation-based secure ABE scheme, we can achieve the stronger security for unclonable PE.

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 229–242. IEEE Computer Society, 2009. (Cited on page 3, 9.)

[ABDS21] Gorjan Alagic, Zvika Brakerski, Yfke Dulek, and Christian Schaffner. Impossibility of quantum virtual black-box obfuscation of classical circuits. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 497–525, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 17.)

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 255–268. ACM Press, June 2020. (Cited on page 3.)

[AGVW13] Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 500–518. Springer, Heidelberg, August 2013. (Cited on page 31.)

[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. (Cited on page 3, 4, 5, 6, 8, 9.)

[AK22] Prabhanjan Ananth and Fatih Kaleoglu. A note on copy-protection from random oracles. Cryptology ePrint Archive, Report 2022/1109, 2022. https://eprint.iacr.org/2022/1109. (Cited on page 3, 9.)

[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. (Cited on page 3, 4, 9, 10.)

[AKL23] Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. Cloning games: A general framework for unclonable primitives. *arXiv (CoRR)*, abs/2302.01874, 2023. (Cited on page 5, 16.)

[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. (Cited on page 3, 9, 17.)

[ALL+21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 526–555, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 12.)

[BB14] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.*, 560:7–11, 2014. (Cited on page 5.)

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014. (Cited on page 13.)

[BL20] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via oracles. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 4:1–4:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. (Cited on page 3, 4, 9, 15.)

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Heidelberg, August 2018. (Cited on page 16, 17.)

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011. (Cited on page 31.)

[CHN+18] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. *SIAM Journal on Computing*, 47(6):2157–2202, 2018. (Cited on page 12.)

[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. (Cited on page 3, 4, 6, 7, 10, 15, 16.)

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *arXiv (CoRR)*, abs/2009.13865, 2020. (Cited on page 3, 5, 8, 9, 25.)

[CV22] Eric Culf and Thomas Vidick. A monogamy-of-entanglement game for subspace coset states. *Quantum*, 6:791, Sep 2022. (Cited on page 6, 10.)

[DGHM18] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, and Daniel Masny. New constructions of identity-based and key-dependent message secure encryption schemes. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 3–31. Springer, Heidelberg, March 2018. (Cited on page 5, 8, 41, 42.)

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017. (Cited on page 9, 14.)

[Got03] Daniel Gottesman. Uncloneable encryption. *Quantum Inf. Comput.*, 3(6):581–602, 2003. (Cited on page 9.)

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, August 2012. (Cited on page 13.)

[GVW15a] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *Journal of the ACM*, 62(6):45:1–45:33, 2015. (Cited on page 13.)

[GVW15b] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, August 2015. (Cited on page 9.)

[GZ20] Marios Georgiou and Mark Zhandry. Unclonable decryption keys. Cryptology ePrint Archive, Report 2020/877, 2020. https://eprint.iacr.org/2020/877. (Cited on page 3, 4, 6, 8, 10, 16, 19.)

[HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 606–636. Springer, Heidelberg, December 2021. (Cited on page 35.)

[KN22] Fuyuki Kitagawa and Ryo Nishimaki. Watermarking PRFs against quantum adversaries. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 488–518. Springer, Heidelberg, May / June 2022. (Cited on page 5, 8, 11, 12, 43.)

[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*, LNCS. Springer, 2022. (Cited on page 3, 10.)

[Mah18] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018. (Cited on page 9, 16, 17.)

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991. (Cited on page 12, 44.)

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 463–472. ACM Press, October 2010. (Cited on page 8.)

[SW22] Or Sattath and Shai Wyborski. Uncloneable decryptors from quantum copy-protection. *arXiv (CoRR)*, abs/2203.05866, 2022. (Cited on page 10.)

[TFKW13] Marco Tomamichel, Serge Fehr, Jędrzej 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. (Cited on page 7, 21.)

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983. (Cited on page 3.)

[Win99] A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999. (Cited on page 25.)

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017. (Cited on page 9, 14.)

[Zha20] Mark Zhandry. Schrödinger's pirate: How to trace a quantum decoder. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 61–91. Springer, Heidelberg, November 2020. (Cited on page 6, 11.)

A AD-SIM secure CP-ABE

We show how to transform any AD-IND secure CP-ABE scheme into AD-SIM secure one in this section.

A.1 Additional Tool

We introduce pseudorandom ciphertext secret key encryption (SKE).

Definition A.1 (Pseudorandom Ciphertext SKE). *A SKE scheme* SKE *is a two tuple* (E,D) *of PPT algorithms.*

- *The encryption algorithm* E*, given a key* $k \in \{0,1\}^\lambda$ *and a plaintext* $m \in \mathcal{M}$*, outputs a ciphertext* c*, where* $\mathcal{M}$ *is the plaintext space of* SKE*.*
- *The decryption algorithm* D*, given a key* k *and a ciphertext* c*, outputs a plaintext* $\tilde{m} \in \{\bot\} \cup \mathcal{M}$*. This algorithm is deterministic.*

Correctness: *We require* $\mathsf{D}(k,\mathsf{E}(k,m)) = m$ *for every* $m \in \mathcal{M}$ *and key* $k \in \{0,1\}^\lambda$*.*

Pseudorandom Ciphertext Property: *Let* $\{0,1\}^\ell$ *be the ciphertext space of* SKE*. We define the following experiment* $\mathsf{Exp}^{\mathsf{pr\text{-}ct}}_{\mathsf{SKE},\mathcal{A}}(1^\lambda,\mathsf{coin})$ *between the challenger and an adversary* $\mathcal{A}$*.*

1. *The challenger generates* $k \leftarrow \{0,1\}^\lambda$*. Then, the challenger sends* 1^λ *to* $\mathcal{A}$*.*

2. *$\mathcal{A}$ may make polynomially many encryption queries adaptively. $\mathcal{A}$ sends $m \in \mathcal{M}$ to the challenger. Then, the challenger returns $c \leftarrow \mathsf{E}(k, m)$ if $\mathsf{coin} = 0$, otherwise $c \leftarrow \{0,1\}^{\ell}$.*
3. *$\mathcal{A}$ outputs $\mathsf{coin}' \in \{0,1\}$.*

We require that for any QPT adversary $\mathcal{A}$, we have

$$\mathsf{Adv}^{\mathsf{pr\text{-}ct}}_{\mathsf{SKE},\mathcal{A}}(\lambda) = \left|\Pr\left[\mathsf{Exp}^{\mathsf{pr\text{-}ct}}_{\mathsf{SKE},\mathcal{A}}(1^\lambda, 0) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{pr\text{-}ct}}_{\mathsf{SKE},\mathcal{A}}(1^\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

Theorem A.2. *If OWFs exist, there exists a pseudorandom-secure SKE scheme.*

A.2 Construction

Since the plaintext space of AD-SIM secure ABE can be expanded by parallel repetition using different instances, we focus on constructing a scheme with the plaintext space $\{0,1\}$.

We construct $\mathsf{CPABE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ using the following tools.

- Compute-and-compare obfuscation $\mathsf{CC.Obf}$ with the simulator $\mathsf{CC.Sim}$.
- Ciphertext-policy ABE $\mathsf{IND\text{-}CP\text{-}ABE} = (\mathsf{IND.Setup}, \mathsf{IND.KG}, \mathsf{IND.Enc}, \mathsf{IND.Dec})$.
- Pseudorandom ciphertext SKE $\mathsf{SKE} = (\mathsf{SKE.E}, \mathsf{SKE.D})$.

The description is as follows.

$\mathsf{Setup}(1^\lambda)$**:**

- Generate $(\mathsf{ind.mpk}, \mathsf{ind.msk}) \leftarrow \mathsf{IND.Setup}(1^\lambda)$.
- Generate $R \leftarrow \{0,1\}^\lambda$.
- Output $\mathsf{mpk} := (R, \mathsf{ind.mpk})$ and $\mathsf{msk} := (R, \mathsf{ind.msk})$.

$\mathsf{KG}(\mathsf{msk}, x)$**:**

- Parse $\mathsf{msk} := (R, \mathsf{ind.msk})$.
- Generate $c \leftarrow \{0,1\}^{\ell_{\mathsf{ct}}}$.
- Generate $\mathsf{ind.sk}_{(x,c)} \leftarrow \mathsf{IND.KG}(\mathsf{ind.msk}, x\|c)$.
- Output $\mathsf{sk}_x := \mathsf{ind.sk}_{(x,c)}$.

$\mathsf{Enc}(\mathsf{mpk}, C, x, \mathsf{m} \in \{0,1\})$**:**

- Parse $\mathsf{mpk} := (R, \mathsf{ind.mpk})$.
- If $\mathsf{m} = 1$, generate ct as follows.
  - Generate $k \leftarrow \{0,1\}^\lambda$ and $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
  - Generate $\mathsf{ind.ct} \leftarrow \mathsf{IND.Enc}(\mathsf{ind.mpk}, G[C,k,R], \mathsf{lock})$, where $G[C,k,R]$ takes as input (x,c) and output 1 if and only if $C(x) = 1$ and $\mathsf{SKE.D}(k,c) \neq R$.
  - Generate $\mathsf{ct} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{IND.Dec}(\cdot, \mathsf{ind.ct}), \mathsf{lock}, R)$.
- If $\mathsf{m} = 0$, generate $\mathsf{ct} \leftarrow \mathsf{CC.Sim}(1^\lambda, \mathsf{pp}_{\mathsf{IND.Dec}'}, |R|)$.
- Output ct.

$\mathsf{Dec}(\mathsf{sk}_x, \mathsf{ct})$**:** Output 1 if $\mathsf{ct}(\mathsf{sk}_x) = R$ and 0 otherwise.

Theorem A.3. *Let $D = \{D_{\mathsf{mpk},C}\}$ be a family of distributions where each $D_{\mathsf{mpk},C}$ outputs $(\mathsf{IND.Dec}(\cdot, \mathsf{ind.ct}), \mathsf{lock}, aux_{\mathcal{X}} := k)$ generated as follows.*

- *Parse* $\mathsf{mpk} := (\mathsf{ind.mpk}, R)$
- *Generate* $k \leftarrow \{0,1\}^\lambda$ *and* $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
- *Generate* $\mathsf{ct} \leftarrow \mathsf{IND.Enc}(\mathsf{ind.mpk}, G[C,k,R], 0^\lambda)$.
- *Output* $(\mathsf{IND.Dec}(\cdot, \mathsf{ind.ct}), \mathsf{lock}, aux := k)$.

If CC.Obf *is secure with respect to D,* IND-CP-ABE *is AD-IND secure, and* SKE *is a pseudorandom ciphertext SKE scheme, then* CPABE *satisfies AD-SIM security.*

If our goal is SEL-SIM secure CP-ABE, we can use SEL-IND secure CP-ABE as a building block.

Proof. We first provide the construction of the simulator $\mathcal{Sim} = (\mathcal{SimEnc}, \mathcal{SimKG})$. We see that $\mathcal{SimEnc}(\mathsf{mpk}, C)$ is exactly the same as $\mathsf{Enc}(\mathsf{mpk}, C, \mathsf{m} = 1)$.

$\mathcal{SimEnc}(\mathsf{mpk}, C)$**:**

- Parse $\mathsf{mpk} := (R, \mathsf{ind.mpk})$.
- Generate $k \leftarrow \{0,1\}^\lambda$ and $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
- Generate $\mathsf{ct} \leftarrow \mathsf{IND.Enc}(\mathsf{ind.mpk}, G[C,k,R], \mathsf{lock})$, where $G[C,k,R]$ takes as input (x,c) and output 1 if and only if $C(x) = 1$ and $\mathsf{D}(k,c) \neq R$.
- Generate $\mathsf{ct} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{Dec}(\cdot, \mathsf{ct}), \mathsf{lock}, R)$.
- Output ct and $\mathsf{st} := k$.

$\mathcal{SimKG}(\mathsf{msk}, \mathsf{st}, x, \mathsf{m} \in \{0,1\})$**:**

- Parse $\mathsf{msk} := (R, \mathsf{ind.msk})$ and $\mathsf{st} := k$.
- Generate $c \leftarrow \{0,1\}^{\ell_{\mathsf{ct}}}$ if $\mathsf{m} = 1$ and $c \leftarrow \mathsf{SKE.E}(k, R)$ otherwise.
- Generate $\mathsf{sk}_{(x,c)} \leftarrow \mathsf{IND.KG}(\mathsf{ind.msk}, x \| c)$.
- Output $\mathsf{sk}_x := \mathsf{sk}_{(x,c)}$.

Let $\mathcal{A}$ be any QPT adversary. Conditioned that an adversary $\mathcal{A}$ outputs $\mathsf{m} = 1$, $\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda, 0)$ and $\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda, 1)$ are exactly the same experiment. Thus, it is sufficient to prove $\Pr\left[1 \leftarrow \mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda, 0)\right]$ and $\Pr\left[1 \leftarrow \mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda, 1)\right]$ are negligibly close conditioned that $\mathsf{m} = 0$. We prove this by using the following sequence of experiments.

Hyb_1:This is $\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathcal{Sim},\mathcal{A}}(\lambda, 1)$ where $\mathcal{A}$ outputs $\mathsf{m} = 0$. The detailed description is as follows.

1. The challenger computes $(\mathsf{ind.mpk}, \mathsf{ind.msk}) \leftarrow \mathsf{IND.Setup}(1^\lambda)$ and $R \leftarrow \{0,1\}^\lambda$, and sends $\mathsf{mpk} := (R, \mathsf{ind.mpk})$ to $\mathcal{A}$.
2. $\mathcal{A}$ can get access to the following oracle.

   $O_{\mathsf{KG},1}(x)$**:** Given x, it generates $c \leftarrow \{0,1\}^{\ell_{\mathsf{ct}}}$ and returns $\mathsf{sk}_x \leftarrow \mathsf{IND.KG}(\mathsf{ind.msk}, x \| c)$.
3. $\mathcal{A}$ sends C and $\mathsf{m} = 0$ to the challenger, where C satisfies $C(x) = 0$ for all x queried by $\mathcal{A}$ in the previous step. The challenger does the following.
   - Generate $k \leftarrow \{0,1\}^\lambda$ and $\mathsf{lock} \leftarrow \{0,1\}^\lambda$.
   - Generate $\mathsf{ind.ct} \leftarrow \mathsf{IND.Enc}(\mathsf{ind.mpk}, G[C,k,R], \mathsf{lock})$, where $G[C,k,R]$ takes as input (x,c) and output 1 if and only if $C(x) = 1$ and $\mathsf{SKE.D}(k,c) \neq R$.
   - Return $\mathsf{ct} \leftarrow \mathsf{CC.Obf}(1^\lambda, \mathsf{IND.Dec}(\cdot, \mathsf{ind.ct}), \mathsf{lock}, R)$.
4. $\mathcal{A}$ can get access to the following oracle.

$O_{\mathsf{KG},2}(x)$: Given x, if $C(x) = 0$, it returns $\mathsf{sk}_x \leftarrow \mathsf{IND.KG}(\mathsf{ind.msk}, x\|c)$, where $c \leftarrow \{0,1\}^{\ell_{\mathsf{ct}}}$. Otherwise, it returns sk_x generated as follows.

- Generate $c \leftarrow \mathsf{SKE.E}(k, R)$.
- Returns $\mathsf{sk}_x \leftarrow \mathsf{IND.KG}(\mathsf{ind.msk}, x\|c)$.

5. $\mathcal{A}$ outputs $\mathsf{coin}' \in \{0,1\}$.

Hyb_2: This is the same as Hyb_1 except that ind.ct is generated as $\mathsf{ind.ct} \leftarrow \mathsf{IND.Enc}(\mathsf{ind.mpk}, G[C,k,R], 0^\lambda)$.

In Hyb_1 and Hyb_2, $\mathcal{A}$ can query x such that $C(x) = 1$ to $O_{\mathsf{KG},2}$. However, for such query x, $O_{\mathsf{KG},2}$ returns $\mathsf{sk}_x \leftarrow \mathsf{IND.KG}(\mathsf{ind.msk}, x\|c)$, where $c \leftarrow \mathsf{SKE.E}(k, R)$. We see that if $c \leftarrow \mathsf{SKE.E}(k, R)$, $G[C,k,R](x\|c) = 0$ since $\mathsf{SKE.D}(k, c) = R$. Thus, $\mathcal{A}$ can obtain decryption keys only for an attribute $x\|c$ such that $G[C,k,R](x\|c) = 0$ for the policy $G[C,k,R]$. Then, from the AD-IND security of CPABE, we have $|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_3: This is the same as Hyb_2 except that ct is generated as $\mathsf{ct} \leftarrow \mathsf{CC.Sim}(1^\lambda, \mathsf{pp}_{\mathsf{IND.Dec}}, |R|)$.

From the security of CC.Obf, we have $|\Pr[\mathsf{Hyb}_2 = 1] - \Pr[\mathsf{Hyb}_3 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_4: This is the same as Hyb_3 except that $O_{\mathsf{KG},2}$, given an input x, c is generated as $c \leftarrow \{0,1\}^{\ell_{\mathsf{ct}}}$ even when $C(x) = 1$.

From the security of SKE, we have $|\Pr[\mathsf{Hyb}_3 = 1] - \Pr[\mathsf{Hyb}_4 = 1]| = \mathsf{negl}(\lambda)$.

Hyb_4 is exactly $\mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathit{Sim},\mathcal{A}}(\lambda, 0)$ where $\mathcal{A}$ outputs $\mathsf{m} = 0$. Thus, from the above discussions, we have $\left|\Pr\left[1 \leftarrow \mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathit{Sim},\mathcal{A}}(\lambda, 0)\right] - \Pr\left[1 \leftarrow \mathsf{Exp}^{\mathsf{ad\text{-}sim}}_{\mathsf{CPABE},\mathit{Sim},\mathcal{A}}(\lambda, 1)\right]\right| = \mathsf{negl}(\lambda)$ conditioned that $\mathcal{A}$ outputs $\mathsf{m} = 0$. This completes the proof. □

B Succinct CPFE

We review the definition of hash encryption introduced in [DGHM18].

Definition B.1 (Hash Encryption). *A hash encryption scheme* HE *is a four tuple* $(\mathsf{HKG}, \mathsf{Hash}, \mathsf{HEnc}, \mathsf{HDec})$ *of PPT algorithms.*

- HKG *is the key generation algorithm that takes as input a security parameter* 1^λ *and the input-length* 1^n*. Then, it outputs a hash key* hk*.*
- Hash *is the (deterministic) hashing algorithm that takes a hash key* hk *and a string* $x \in \{0,1\}^n$ *as input, and outputs a hash value* $h \in \{0,1\}^\lambda$*.*
- HEnc *is the encryption algorithm that takes a hash key* hk*, a triple* $(h, j, \alpha) \in \{0,1\}^\lambda \times [n] \times \{0,1\}$*, and a message* $m \in \{0,1\}^*$ *as input, and outputs a ciphertext* hct*.*
- HDec *is the (deterministic) decryption algorithm that takes a hash key* hk*, a string* $x \in \{0,1\}^n$*, and a ciphertext* ct *as input, and outputs a message* m *which could be the special invalid symbol* $\perp$*.*

We require the following properties.

Correctness *Let* $\mathsf{hk} \leftarrow \mathsf{HKG}(1^\lambda, 1^n)$*. We have* $\mathsf{HDec}(\mathsf{hk}, x, \mathsf{HEnc}(\mathsf{hk}, (\mathsf{Hash}(\mathsf{hk}, x), j, x[j]), m)) = m$ *for all strings* $x = (x[1], \ldots, x[n]) \in \{0,1\}^n$*, positions* $j \in [n]$*, and plaintext* $m \in \{0,1\}^*$*. and all messages* m*.*

Security *Consider the following security experiment* $\mathsf{Expt}^{\mathsf{he}}_{\mathsf{HE},\mathcal{A}}(\lambda, \mathsf{coin})$ *between a challenger and an adversary* $\mathcal{A}$*.*

1. $\mathcal{A}$ *sends* $x = (x[1], \ldots, x[n]) \in \{0,1\}^n$ *to the challenger.*
2. *The challenger generates* $\mathsf{hk} \leftarrow \mathsf{HKG}(1^\lambda, 1^n)$ *and sends* hk *to* $\mathcal{A}$*.*

3. *$\mathcal{A}$ sends a position $j \in [n]$ and a pair of messages (m_0, m_1) of the same length to the challenger. The challenger computes $h \leftarrow \mathsf{Hash}(\mathsf{hk}, x)$ and $\mathsf{hct} \leftarrow \mathsf{HE}(\mathsf{hk}, (h, j, 1 \oplus x_j), m_{\mathsf{coin}})$, and returns hct to $\mathcal{A}$.*
4. *$\mathcal{A}$ outputs $\mathsf{coin}' \in \{0,1\}$.*

For any QPT $\mathcal{A}$, we have

$$\mathsf{Adv}^{\mathsf{he}}_{\mathsf{HE},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{\mathsf{he}}_{\mathsf{HE},\mathcal{A}}(\lambda,0)=1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{he}}_{\mathsf{HE},\mathcal{A}}(\lambda,1)=1\right]\right| = \mathsf{negl}(\lambda).$$

Theorem B.2 ([DGHM18]). *If the LWE or exponentially-hard LPN assumption holds, there exists a hash encryption.*

We present a CPFE scheme that satisfies 1-bounded security Definition 4.2 and the succinct key property Definition 4.3.

Building blocks.

- Hash encryption $\mathsf{HE} = (\mathsf{HKG}, \mathsf{Hash}, \mathsf{HEnc}, \mathsf{HDec})$.
- Grabled circuit $(\mathsf{GC.Grbl}, \mathsf{GC.Eval}, \mathsf{GC.Sim})$.

Our CPFE scheme CPFE is as follows.

$\mathsf{Setup}(1^\lambda, x)$**:**

- Generate $\mathsf{hk} \leftarrow \mathsf{HKG}(1^\lambda, 1^n)$.
- Compute $h_x := \mathsf{Hash}(\mathsf{hk}, x)$
- Output $\mathsf{MPK} := (\mathsf{hk}, h_x)$ and $\mathsf{sk}_x := x$.

$\mathsf{Enc}(\mathsf{MPK}, C)$**:**

- Parse $\mathsf{MPK} = (\mathsf{hk}, h_x)$.
- Generate $(\widetilde{C}, \{\mathsf{label}_{i,\beta}\}_{i\in[n],\beta\in\{0,1\}}) \leftarrow \mathsf{GC.Grbl}(1^\lambda, C)$.
- Generate $\mathsf{hct}_{i,\beta} \leftarrow \mathsf{HEnc}(\mathsf{hk}, (h_x, i, \beta), \mathsf{label}_{i,\beta})$ for $i \in [n]$ and $\beta \in \{0,1\}$.
- Output $\mathsf{ct} := (\widetilde{C}, \{\mathsf{hct}_{i,\beta}\}_{i\in[n],\beta\in\{0,1\}})$.

$\mathsf{Dec}(\mathsf{sk}_x, \mathsf{ct})$**:**

- Parse $\mathsf{sk}_x = x$ and $\mathsf{ct} = (\widetilde{C}, \{\mathsf{hct}_{i,\beta}\}_{i\in[n],\beta\in\{0,1\}})$.
- Compute $\mathsf{label}_{i,x[i]} := \mathsf{HDec}(\mathsf{hk}, x, \mathsf{hct}_{i,x[i]})$ for $i \in [n]$.
- Output $y' := \mathsf{GC.Eval}(\widetilde{C}, \{\mathsf{label}_{i,x[i]}\}_{i\in[n]})$.

Theorem B.3. *If* HE *is a secure hash encryption and* GC *is a secure garbling, then* CPFE *is 1-bounded secure.*

Proof. We define a sequence of games to prove the theorem.

Hyb_0**:** This is $\mathsf{Expt}^{\mathsf{he}}_{\mathsf{HE},\mathcal{A}}(\lambda, \mathsf{coin})$ where $\mathsf{coin} \leftarrow \{0,1\}$.

Hyb^j_0**:** This is the same as Hyb_0 except that for $i \in [j]$ we generate $\mathsf{hct}_{i,1\oplus x[i]} \leftarrow \mathsf{HEnc}(\mathsf{hk}, (h_x, i, 1\oplus x[i]), 0^{|\mathsf{label}|})$ instead of generating $\mathsf{hct}_{i,1\oplus x[i]} \leftarrow \mathsf{HEnc}(\mathsf{hk}, (h_x, i, 1\oplus x[i]), \mathsf{label}_{i,1\oplus x[i]})$.

Hyb_1**:** This is the same as Hyb^n_0.

Hyb_2**:** This is the same as Hyb_1 except that we generate $(\widetilde{C}_{\mathsf{coin}}, \{\mathsf{label}_{i,x[i]}\}_{i\in[n]}) \leftarrow \mathsf{GC.Sim}(1^\lambda, 1^{|C|}, C_{\mathsf{coin}}(x))$ instead of generating $(\widetilde{C}_{\mathsf{coin}}, \{\mathsf{label}_{i,\beta}\}_{i\in[n],\beta\{0,1\}}) \leftarrow \mathsf{GC.Grbl}(1^\lambda, C_{\mathsf{coin}})$.

We define $\mathtt{SUC}_i$ (resp. $\mathtt{SUC}_i^j$) be the event that $\mathcal{A}$ outputs $\mathsf{coin}' = \mathsf{coin}$ in Hyb_i (resp. Hyb_i^j).

We have $\mathsf{Hyb}_0^0 = \mathsf{Hyb}_0$. It holds that $\left|\Pr\left[\mathtt{SUC}_0^{j-1}\right] - \Pr\left[\mathtt{SUC}_0^{j}\right]\right| = \mathsf{negl}(\lambda)$ for all $j \in [n]$ due to the security of HE.

Due to the security of GC, it holds that $|\Pr[\mathtt{SUC}_1] - \Pr[\mathtt{SUC}_2]| = \mathsf{negl}(\lambda)$ since $\{\mathsf{label}_{i,1\oplus x[i]}\}_{i\in[n]}$ is never used in Hyb_1.

It trivially holds $\Pr[\mathtt{SUC}_2] = \frac{1}{2}$ since $C_0(x) = C_1(x)$ and $\widetilde{C}_{\mathsf{coin}}$ is generated by $\mathsf{GC.Sim}(1^\lambda, 1^{|C|}, C_{\mathsf{coin}}(x))$ in Hyb_2. This complete the proof. □

Theorem B.4. CPFE *satisifes the succinct key property.*

Proof. It trivially holds since the setup algorithm Setup runs $\mathsf{HKG}(1^\lambda, 1^n)$ and $\mathsf{Hash}(\mathsf{hk}, x)$, and outputs $\mathsf{MPK} := (\mathsf{hk}, h)$ and $\mathsf{sk}_x := x$. □

We complete the proof of Theorem 4.4 by Theorems B.2 to B.4.

C Injective Commitment with Equivocal Mode

We present a variant of Naor's commitment where we use a commitment key instead of receiver's first message. We use injective PRG $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{3\lambda}$, which can be constructed from injective OWF (with evaluation key generation algorithm).[13] Let $\mathcal{M} := \{0,1\}^\ell$ and $\mathcal{R} := \{0,1\}^{\lambda\cdot\ell}$.

$\mathsf{Setup}(1^\lambda)$**:**

- Choose $s_i \leftarrow \{0,1\}^{3\lambda}$ for $i \in [\ell]$.
- Output $\mathsf{ck} := (s_1, \ldots, s_\ell)$.

$\mathsf{Commit}(\mathsf{ck}, m \in \{0,1\}^\ell)$**:**

- Parse $\mathsf{ck} = (s_1, \ldots, s_\ell)$.
- Choose $r_i \leftarrow \{0,1\}^\lambda$ for $i \in [\ell]$.
- Compute $x_i := \mathsf{PRG}(r_i)$ for $i \in [\ell]$.
- Set $y_i := x_i$ if $m_i = 0$, otherwise $y_i := x_i \oplus s_i$ for $i \in [\ell]$.
- Output $\mathsf{com} := (y_1, \ldots, y_\ell)$.

$\mathsf{EqSetup}(1^\lambda)$**:**

- Choose $\widetilde{r}_{b,i} \leftarrow \{0,1\}^\lambda$ for $i \in [\ell]$ and $b \in \{0,1\}$.
- Set $s_i^* := \mathsf{PRG}(\widetilde{r}_{0,i}) \oplus \mathsf{PRG}(\widetilde{r}_{1,i})$ for $i \in [\ell]$.
- Set $y_i^* := \mathsf{PRG}(\widetilde{r}_{0,i})$ for $i \in [\ell]$.
- Output $\mathsf{ck}^* := (s_1^*, \ldots, s_\ell^*)$, $\mathsf{com}^* := (y_1^*, \ldots, y_\ell^*)$, and $\mathsf{td} := \{\widetilde{r}_{b,i}\}_{i\in[\ell],b\in\{0,1\}}$.

$\mathsf{Open}(\mathsf{td}, m \in \{0,1\}^\ell, \mathsf{com}^*)$**:**

- Parse $\mathsf{td} = \{\widetilde{r}_{b,i}\}_{i\in[\ell],b\in\{0,1\}}$ and $\mathsf{com}^* := (y_1^*, \ldots, y_\ell^*)$.
- Set $r_i^* := \widetilde{r}_{0,i}$ if $m_i = 0$, otherwise $r_i^* := \widetilde{r}_{1,i}$.
- Output $r^* := (r_1^*, \ldots, r_\ell^*)$.

[13] See a remark by Kitagawa and Nishimaki [KN22, Section B.1 in the full version] for "injective OWF with evaluation key generation algorithm".

We can verify that a message m and randomness $(r_1, \ldots, r_\ell)$ is a valid opening for a commitment $(y_1, \ldots, y_\ell)$ by checking

$$y_i = \mathsf{PRG}(r_i) \oplus m_i \cdot s_i$$

for all $i \in [\ell]$.

This is the Naor's commtiement [Nao91] and it has statistical binding and computational hiding properties.

It is easy to see that the construction above satisfies injectivity since s_i is uniformly random for all $i \in [\ell]$ and PRG is injective.

It is also easy to see that the construction above satisifes trapdoor equivocality. Due to the pseudorandomness of PRG, $(\mathsf{ck}^*, \mathsf{com}^*)$ is computationally indistinguishable from $(\mathsf{ck}, \mathsf{com})$.[14] In addition, it holds that $y_i^* = \mathsf{PRG}(\widetilde{r}_{m_i,i}) \oplus s_i^* \cdot m_i$ since $s_i^* = \mathsf{PRG}(\widetilde{r}_{0,i}) \oplus \mathsf{PRG}(\widetilde{r}_{1,i})$. Thus, the trapdoor equivocality holds.

[14] We can use security of $\mathsf{PRG}(\widetilde{r}_{1,i})$ and $\mathsf{PRG}(\widetilde{r}_{0,i})$ if we open to $m_i = 0$ and $m_i = 1$, respectively.